



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Control Predictivo Basado en Escenarios para Sistemas Lineales con Saltos Markovianos

Tesis Doctoral

Autor: Manuel Alejandro Hernández Mejías

Director: Dr. Antonio Sala Piqueras

Instituto Univ. de Automática e Informática Industrial (aI2)
Departamento de Ingeniería de Sistemas y Automática (ISA)

Universitat Politècnica de València, España

Junio, 2016

Control Predictivo Basado en Escenarios para Sistemas Lineales con Saltos Markovianos

PROGRAMA DE DOCTORADO EN AUTOMÁTICA, ROBÓTICA E
INFORMÁTICA INDUSTRIAL

Director: Dr. Antonio Sala, Catedrático de Universidad Instituto
Universitario de Automática e Informática Industrial

Departamento de Ingeniería de Sistemas y Automática (ISA)
Universitat Politècnica de València, España
Junio, 2016

Este trabajo ha sido parcialmente financiado por el Gobierno de España bajo los proyectos de investigación DPI2011-27845-C02-01, DPI2011-27845-C02-02 y por la Generalitat Valenciana bajo los proyectos de investigación: PROMETEO/2008/088 y PROMETEOII/2013/004.

El autor ha sido beneficiario de una beca FPU (FPU12/02107) concedida por el Ministerio de Educación, Cultura y Deporte.

Para mi esposa Dayanna, mis padres (Manuel y Yamilesca) y mi prima Jeannareth.

Agradecimientos

Han transcurrido tres duros, pero a la vez gratificantes, años desde que comencé el camino para el desarrollo de esta tesis doctoral. Son muchas las personas que, directa o indirectamente, han ayudado a que este trabajo se haya llevado a cabo y a las que, por supuesto, tengo que agradecer.

En primer lugar, y como no, quiero agradecer infinitamente a mi director de tesis, Dr. Antonio Sala Piqueras, por la inmensa dedicación que ha tenido en estos últimos años para llevar por buen camino el desarrollo del trabajo de investigación, por el arduo esfuerzo que conlleva dirigir una tesis doctoral (incalculables horas de reuniones, revisiones exhaustivas de los trabajos a publicar, ayuda en programación, *etc*), por su dedicación en mi formación como investigador, pero sobre todo por su calidad humana: consejos personales, paciencia, humildad, sinceridad. Gracias Antonio, por ser como eres. Ha sido un verdadero placer trabajar contigo.

Quiero extender mi agradecimiento al profesor Dr. Carlos Ariño, por ayudarme siempre que lo necesité, por siempre recibirme en su despacho y aclarar todas las dudas que le exponía. Considero que su ayuda fue clave para el progreso de la tesis, gracias Carlos. Asimismo agradecer a Andrés Querol, por toda su ayuda a lo largo de estos años.

Quiero agradecer al profesor Dr. Ramón Blasco, por darme la oportunidad de trabajar con él y así poder retomar el estudio de problemas relacionados con sistemas de potencias, energías renovables, *etc*, permitiéndome combinar mis dos mundos profesionales; aplicación de técnicas de control para sistemas eléctricos.

Quiero hacer especial mención a cada uno de mis compañeros y excompañeros con los que he compartido en esta etapa de doctorado; en primer lugar (y sin un orden en especial) a Jesús Velasco, por sus invitaciones a esquiar y a Bronchales, a demás de su buena amistad, a Vanessa por siempre cambiar la dinámica de la sala y ser la organizadora de reuniones fuera de la universidad,

a José Luis por todos sus consejos en el inicio de mi doctorado, a los mexicanos Rubén y Temo, por todas las cervezas y conversaciones compartidas, a Jadira, Alberto, Gabi, Diego, Gilberto, Alejandro Laguna, Alejandro Vignoni, a los ecuatorianos (Henry, Frank, Julio, Victor). Gracias a todos por crear un entorno de trabajo espectacular.

En segundo lugar, quiero dar una mención muy especial a mi esposa Danyanna, sin su apoyo, ayuda, comprensión, paciencia, cariño, estoy seguro de que no hubiese podido realizar satisfactoriamente esta tesis. Han sido tres años muy duros para mí y siempre he contado con su ayuda para seguir adelante.

En tercer lugar, y no menos importante, quiero darle las gracias a mis padres (Manuel y Yamilesca) por siempre sentir su cariño, apoyo, a pesar de la distancia que nos separa. Es evidente que sin su apoyo no hubiese podido tomar la decisión de dejar mi país para adentrarme en esta aventura. Mil gracias. También a mi hermano y mi cuñada, por todo lo vivido en estos años lejos de nuestra familia. No quiero dejar de mencionar a mis suegro (William y Vestalia), su ayuda y apoyo es fundamental para mí, gracias por todo.

Finalmente y para terminar, agradecer a mi gran amigo Jamie Rhodes, simplemente por su amistad sincera, por siempre querer compartir conmigo las veces que viene a Valencia, por todos sus consejos, por todas sus atenciones. Siempre digo, cuando se vive fuera de su ciudad/país, los verdaderos amigos se convierten en tu familia, y a tí te considero de mi familia. Thank you Sir.

Sinceramente y de corazón gracias a todos.

Manuel Alejandro Hernández Mejías.

Abstract

In this thesis, invariant-set theory is used to study the stability and feasibility of constrained scenario-based predictive controllers for Markov-jump linear systems. In the underlying optimisation problem of the predictive controllers technique, considers all possible future realisations of certain variables (uncertainty, disturbances, operating mode) or just a subset of those.

Two different scenarios denoted as not risky and risky are studied. In the former, the trajectories of the system with initial states belonging to certain invariant sets, converge (in mean square sense) to the origin or an invariant neighbourhood of it with 100 % probability.

In such cases, the conditions that scenario trees must meet in order to guarantee stability and feasibility of the optimisation problem are analysed. Afterwards, the scenario-based predictive controller for Markov-jump linear systems under hard constraints and no disturbances is formulated.

A study is presented for risky scenarios to determine sequence-dependent controllable sets, for which there exists a control law such that the system can be driven to the origin only for a particular realisation of uncertainty, disturbances, *etc.* A control law (optimal for disturbances-free systems and suboptimal for disturbed systems) able to steer the system to the origin with a probability less than 100% (denoted as reliability bound), is proposed for states belonging to those regions.

Note that closed-loop unstable systems have zero reliability bound. Hence, an algorithm to determine the mean-time to failure is developed. In this context, failure means a violation in the constraints of the process' states and/or inputs in a future time.

Resumen

La presente tesis emplea la teoría de conjuntos invariantes para el estudio de estabilidad y factibilidad de controladores predictivos basados en escenarios para sistemas lineales con saltos markovianos sujetos a restricciones. En el problema de optimización subyacente a la técnica de controladores predictivos, se consideran bien sea todas las posibles realizaciones futuras de una variable (incertidumbres, perturbaciones, modo de funcionamiento) o solo un subconjunto de estas.

Se estudian dos escenarios diferentes, denotados como *escenarios no arriesgados* y *arriesgados*, entendiéndose como no arriesgados, aquellos en donde las trayectorias del sistema con estados iniciales pertenecientes a ciertos conjuntos invariantes, convergen –en media– al origen o a una vecindad invariante de este con un 100 % de probabilidad.

Para estos casos, se presenta un análisis de las condiciones que deben cumplir los árboles de escenarios para garantizar estabilidad –en media– y factibilidad del problema de optimización. Luego se formula el control predictivo basado en escenarios para sistema lineales con saltos markovianos sujeto a restricciones y en ausencia de perturbaciones.

En presencia de escenarios arriesgados, se propone el cálculo de conjuntos controlables dependientes de secuencias para los cuales existen una ley de control tal que el sistema puede ser conducido al origen, solo para una realización en particular de la incertidumbre, perturbaciones, *etc.* Para estados pertenecientes a estos conjuntos, se propone una ley de control (óptima para el caso de sistemas libres de perturbaciones y, subóptima para sistemas perturbados) capaz de dirigir el sistema al origen con una probabilidad menor al 100 %, dicha probabilidad es denotada como *cota de confiabilidad*.

Sistemas inestables en lazo cerrado tienen cota de confiabilidad igual a cero, por consiguiente se diseña un algoritmo que determina el tiempo medio para fallar. En este contexto, un fallo se entiende como la violación de las restricciones en los estados y/o entradas del proceso en algún instante de tiempo futuro.

Resum

La present tesi empra la teoria de conjunts invariants per a l'estudi d'estabilitat i factibilidad de controladors predictius basats en escenaris per a sistemes lineals amb salts markovians subjectes a restriccions. En el problema d'optimització subjacent a la tècnica de controladors predictius, es consideren bé siga totes les possibles realitzacions futures d'una variable (incerteses, pertorbacions, modes de funcionament) o només un subconjunt d'aquestes.

S'estudien dos escenaris diferents, denotats com a escenaris no arriscats i arriscats, entenent-se com no arriscats, aquells on les trajectòries del sistema amb estats inicials pertanyents a certs conjunts invariants, convergeixen –en mitjana– a l'origen o a un veïnatge invariant d'est amb un 100 % de probabilitat.

Per a aquests casos, es presenta una anàlisi de les condicions que han de complir els arbres d'escenaris per a garantir estabilitat –en mitjana– i factibilidad del problema d'optimització. Després es formula el control predictiu basat en escenaris per a sistema lineals amb salts markovians subjecte a restriccions i en absència de pertorbacions.

En presència d'escenaris arriscats, es proposa el càlcul de conjunts controlables dependents de seqüències per als quals existeix una llei de control tal que el sistema pot ser conduït a l'origen, solament per a una realització en particular de l'incertesa, pertorbacions, *etc.* Per a estats pertanyents a aquests conjunts, es proposa una llei de control (òptima per al cas de sistemes lliures de pertorbacions i, subòptima per a sistemes pertorbats) capaç de dirigir el sistema cap a l'origen amb una probabilitat menor del 100 %, aquesta probabilitat és denotada com a cota de confiabilitat.

Sistemes inestables en llaç tancat tenen cota de confiabilitat igual a zero, per tant es dissenya un algorisme que determina el temps mitjà per a fallar. En aquest context, una fallada s'entén com la violació de les restriccions en els estats i/o entrades del procés en algun instant de temps futur.

Índice general

Índice General	XVII
Objetivos, Estructura y Contribuciones	1
Publicaciones	7
I Estado del Arte	9
1 Control Predictivo Basado en Modelo	11
1.1 Teoría de Sistemas	11
1.1.1 Sistemas en Tiempo Discreto	11
1.1.2 Estabilidad	12
1.2 Teoría de Conjuntos Invariantes	14
1.3 Politopos	17
1.4 Problema de Optimización Estándar	22
1.5 Controladores Predictivos	23
1.5.1 Breve Historia sobre MPC	23
1.5.2 Idea Básica del MPC	24
1.5.3 Estabilidad y Factibilidad en MPC	25
1.5.4 MPC para Sistemas LTI	27
1.5.5 Control Predictivo Robusto para Sistemas LTI Inciertos	29
1.6 Diferentes Formulaciones de los MPC	36
2 Sistemas Lineales con Saltos Markovianos	41
2.1 Introducción	41
2.2 Cadena de Markov	42
2.3 Control Óptimo para MJLS sin Restricciones	44
2.4 Estabilidad de MJLS	46
2.5 Control Óptimo para MJLS con Incertidumbres sujeto a Restricciones	48
2.5.1 Formulación del problema	48
2.5.2 Discusión	53

2.6	Conclusiones	53
3	Control Predictivo para Sistemas Lineales con Saltos Markovianos	55
3.1	Introducción	56
3.2	Preliminares y Notación	56
3.2.1	Notación	57
3.2.2	Control Óptimo sin Restricciones para MJLS	59
3.2.3	Descripción del Problema	60
3.3	Desarrollo del Control Predictivo para MJLS	61
3.3.1	Modelo de Predicción	61
3.3.2	Conjuntos Terminal y Factible	63
3.3.3	Cálculo de la acción de control	65
3.3.4	Cuestiones Computacionales	66
3.4	Ejemplo Ilustrativo	68
3.4.1	Resultados para diferentes simulaciones	71
3.5	Control Predictivo Basados en Escenarios	76
3.5.1	Diseño MPC Estocástico	77
3.6	Conclusiones	83
	II Contributions	85
4	Multiple-horizon Predictive Control for Markov/switched Linear Systems	87
4.1	Introduction	88
4.2	Preliminaries and Problem statement	88
4.2.1	Predictive Control for Markov/switched Systems.	89
4.2.2	Problem Statement	90
4.3	Sequence-set Predictive Control	91
4.3.1	Prediction Model	91
4.3.2	Sequence-set-Dependent Cost Index for Predictive Control	92
4.4	Terminal and Feasible Sets	94
4.4.1	Mode-dependent Terminal Set.	94
4.4.2	Mode-dependent Feasible Sets.	94
4.5	Multiple-horizon Predictive Control	95
4.5.1	Receding Horizon Implementation (minimax case)	97
4.5.2	Choice of Sequence Sets	99
4.6	Numerical example	100
4.7	Conclusions of this Chapter	102

5	Ingredients for Stable Receding-horizon Scenario-based Predictive Control	103
5.1	Introduction	104
5.2	Preliminaries and Problem Statement	105
5.2.1	Information Model	106
5.2.2	Infinite Horizon and Mean-square Stability.	107
5.2.3	Problem Statement	109
5.3	Terminal Ingredients of the Scenario-based Approach	109
5.3.1	Quadratic Case	110
5.4	Predictive Control on Scenarios/trees	112
5.4.1	Cost Index Associated to a non-empty Tree	116
5.4.2	Constraints and Predictive Control Problem Set-up	117
5.5	Feasibility Analysis	118
5.5.1	Recursive Feasibility in Receding-horizon Implementation	122
5.6	Stability	123
5.6.1	Scenario Generation	125
5.7	Conclusions of this Chapter	126
6	Reliable MPC for Markov-jump Linear Systems	129
6.1	Introduction	130
6.2	Preliminaries	130
6.2.1	Problem Statement	131
6.3	Reliability analysis of a stochastically stable Markov-jump system	132
6.3.1	Choice of high-likelihood terminal and feasible sets for MPC technique	133
6.4	Numerical Example	137
6.5	Conclusions of this Chapter	141
7	Reliable Controllable Sets for Constrained Markov-Jump Linear Systems	143
7.1	Introduction	144
7.2	Preliminaries, Problem Statement	145
7.2.1	Robust h -step Sets	145
7.2.2	Problem Statement and Auxiliary Notation	146
7.3	Augmented and Sequence-dependent Sets	148
7.3.1	Augmented h -step Sets	148
7.3.2	Sequence-dependent h -step Sets	150
7.4	Reliable On-line Controller Design	153

7.4.1	Controller with known Current Mode in On-line Operation	153
7.4.2	Controller With no On-line Mode Information	156
7.4.3	Discussion and Computational Issues	157
7.5	Numerical Example	159
7.6	Conclusions of this Chapter	162
8	Reliability and Time-to-Failure Bounds for Constrained MJLS	165
8.1	Introduction	166
8.2	Preliminaries	167
8.2.1	Problem statement	169
8.3	Reliability Bound Computation	170
8.3.1	Iterative $RL_l(\cdot, \cdot)$ Algorithm	172
8.3.2	Terminal Set Based Algorithm	177
8.4	Mean time to failure bound computation	180
8.5	Terminal-set algorithm and implementation guidelines	181
8.6	Examples	186
8.7	Conclusion	189
	Conclusiones y Trabajo Futuro	191
	Bibliografía	196
	Apéndices	210
A	Optimización con Restricciones	211
B	Ejemplos	215
B.1	Ejemplo: MPC para Sistemas LTI	215
B.2	Ejemplo: MPC Robusto para Sistemas LTIs Inciertos	218
B.3	Ejemplo: MPC Robusto para MJLS Inciertos	224

Objetivos, Estructura y Contribuciones

Introducción y motivación

Gran parte de los sistemas dinámicos, por ejemplo: sistemas eléctricos, mecánicos, biológicos, económicos, *etc*, pueden ser descritos en términos de modelos matemáticos. Estos modelos son utilizados para representar el comportamiento del sistema y facilitan el diseño de sistemas de control capaces de modificar las entradas del proceso con el fin de afectar a conveniencia la salida del mismo.

Muchos de estos sistemas se encuentran sujetos a restricciones tanto en las entradas como en las salidas del proceso. Existen diversos métodos para el diseño de controladores para procesos sujetos a restricciones, en particular, esta tesis se centra en los *controladores predictivos basados en modelos* (MPC)¹, debido principalmente a la capacidad de incorporar las restricciones del sistema en el propio diseño del controlador, y a la posibilidad de tratar con diferentes tipos de procesos bien sea lineales o no lineales, monovariable o multivariable.

MPC es un método de control que hace uso explícito del conocimiento de la dinámica del proceso representado por un modelo (lineal o no lineal), tomando en cuenta las restricciones. La idea básica detrás de esta estrategia es la siguiente: se obtiene una secuencia de acciones de control –resolviendo *en línea*– para cada instante de tiempo, un problema de control de *horizonte finito*, utilizando el estado actual del sistema. En el siguiente instante de tiempo, solo se aplica la primera acción de control al proceso, repitiéndose el cálculo con el nuevo estado medido sobre un horizonte desplazado. Detalles sobre la

¹ Acrónimo derivado del término en inglés *Model Predictive Control*. Dado que estas siglas son con la que se conoce en la literatura esta técnica de control, se usará a lo largo del presente documento.

implementación de los MPC se presentan en el Capítulo 1.

La teoría de conjuntos invariantes desempeña un rol fundamental en el diseño de controladores predictivos para sistemas con restricciones, debido a que el cumplimiento de las mismas puede ser garantizado –para todo tiempo futuro– *si y solo si* el estado inicial del proceso está contenido dentro de un conjunto invariante bajo control. Las definiciones formales de invarianza se exponen en el Capítulo 1.

Gran parte de los sistemas dinámicos mencionado anteriormente, han sido extensamente modelados a través de sistemas lineales invariante en el tiempo (LTI, del acrónimo *linear-time invariant systems*). Un gran número de estos sistemas están sujetos a cambios abruptos en su estructura, interrupciones en las comunicaciones entre subsistemas interconectados, cambios en el punto de operación o rupturas en sus componentes, *etc.* Una forma de modelar adecuadamente esta clase de procesos es mediante *sistemas lineales con saltos markovianos* (MJLS, del acrónimo en inglés *Markov-jump linear systems*). Estos son una clase especial de sistemas estocásticos, los cuales están representados por un conjunto de sistemas lineales que se encuentran asociados a través de una cadena de Markov, que gobierna la transición entre ellos. Una característica distintiva de las cadenas de Markov es que el estado siguiente solo depende del actual y no de las transiciones históricas que condujeron a dicho estado. Ejemplo de sistemas modelados como procesos con saltos markovianos pueden encontrarse en: sistemas de manufactura, control de sistemas aeronáuticos, sistemas de control en red, sistemas económicos, sistemas roboticos, *etc.* (Costa *et al.* 2005).

En sistemas industriales modernos pueden producirse comportamientos imperfectos no deseados, debido por ejemplo a, desgastes normales en los componentes físicos del proceso, entorno ambientales externos, *etc.* Estos fenómenos se refiere a menudo como *fallos* en sensores/actuadores. Dado que los fallos pueden deteriorar el rendimiento del sistema e incluso causar accidentes catastróficos, es de gran importancia detectar dichos fallos a tiempo para la seguridad y fiabilidad de los sistemas de control. Existen estrategias de diseño de sistemas de control antes fallos, que pueden clasificarse en *detección y aislamiento del fallo* (FDI, del acrónimo *fault detection and isolation*) y *control tolerante a fallos* (FTC, del acrónimo *fault-tolerant control*). Se remite al lector al libro (Blanke *et al.* 2010) para un estudio en profundidad de estas técnicas. Los MJLS, han sido extensamente utilizado para modelar de sistemas dinámicos con fallos en sensores y/o actuadores (Li *et al.* 2014).

Existen numerosos trabajos donde se trata el diseño de leyes de control

para MJLS sin restricciones los cuales aseguran la *estabilidad media cuadrática* (MSE, del acrónimo mean-square stability) del sistema. Dicha estabilidad implica que *–en media–* las trayectorias del sistema para estados iniciales perteneciente a un cierto conjunto tienden a converger al origen o a una vecindad del mismo, a medida que se incrementa el tiempo.

El diseño de controladores óptimos para MJLS con restricciones ha sido tratado, entre otros, en los siguientes trabajos (Zhang and Boukas 2009, Zhang *et al.* 2013, Patrinos *et al.* 2014); gran parte de estas contribuciones expresan el problema de optimización en forma de desigualdades matriciales lineales (LMIs, del acrónimo *linear matrix inequality*), minimizando una cota superior del índice de coste obteniendo así una ley de control subóptima, siendo esto el principal inconveniente de este enfoque. Generalizaciones recientes del problema de control óptimo al caso \mathcal{H}_2 y \mathcal{H}_∞ aparecen en, por ejemplo, (Todorov and Fragoso 2014), donde se presentan las adecuadas definiciones de dichas normas en sistemas MJLS y condiciones LMI que garantizan cotas de ellas.

Otra técnica de diseño de controladores óptimos para esta clase de sistema esta basada en la optimización de un problema min-max (Lu *et al.* 2013), en donde se minimiza el *peor caso* de todas las realizaciones posibles de la perturbación y/o incertidumbre. El inconveniente de esta metodología es la carga computacional necesaria para la resolución del problema de optimización, en efecto, tanto el número de restricciones como el número de variables de decisión crecen exponencialmente con el horizonte de predicción, número de vértices, *etc.*

Una alternativa a los controladores basados en LMIs para manejar las restricciones es integrar los resultados antes citados de estabilidad media cuadrática en *controladores predictivos estocásticos* (SMPC, del acrónimo *Stochastic model predictive control*) (Bernardini and Bemporad 2012, Farina *et al.* 2016, Wang *et al.* 2016). Este enfoque presenta los mismos inconvenientes relacionados con el coste computacional, pues, el número de variables de decisión y de restricciones aumentan exponencialmente a medida que se incrementa el horizonte de predicción, número de vértices, *etc.* En particular, a lo largo de esta tesis se utiliza optimización estocástica para calcular una secuencia de acciones de control en un problema predictivo basado en *escenarios*. La idea principal es la siguiente: Dado que resolver el problema de optimización considerando todas las posibles realizaciones futuras de las incertidumbres y/o perturbaciones es computacionalmente muy costoso, el control predictivo basado en *escenarios* propone resolver el problema considerando solo un subconjunto de dichas secuencias. Estas deben cumplir con ciertas condiciones con el fin de asegurar la factibilidad y estabilidad del problema de optimización para instantes fu-

turo, véase los capítulos 4–5; cuando no se pueda probar estabilidad en media cuadrática, en esta tesis se presentan resultados que aseguren un tiempo medio para fallar, entendiéndose como fallo, violar restricciones en tiempo futuro Capítulo 8.

En este contexto se enmarca el desarrollo de la presente tesis, cuyos objetivos y estructura se detallan a continuación:

Objetivos y estructura

El principal objetivo de este trabajo es desarrollar estrategias de controladores predictivos para sistemas lineales con saltos markovianos basados en escenarios arriesgados y no arriesgados, entendiéndose como no arriesgado, el caso en donde las trayectorias del sistemas alcanzan cierto conjunto terminal invariante con una probabilidad del 100 % para estados iniciales pertenecientes a ciertos conjuntos.

Con base en el conocimiento de la teoría de conjuntos invariantes, combinada con la teoría de Lyapunov, se presenta el soporte teórico para comprender los conceptos de estabilidad en el control predictivo basado en escenarios.

Con esto en mente, la presente tesis se ha estructurado en dos grandes partes. En la primera se presenta el estado del arte, el cual está dividido a su vez en 3 capítulos tal como sigue:

- En el capítulo 1 se detallan las definiciones en cuanto a tipos de sistemas en tiempo discreto estudiados en este trabajo, propiedades de estabilidad para sistemas dinámicos, teorías de conjuntos convexos e invariantes, representación de poliedros. A continuación, se presenta la estrategia de diseño de controladores predictivos basado en modelo (MPC), analizando los ingredientes necesarios para probar estabilidad y factibilidad. Para cerrar el capítulo, se describe el desarrollo de MPC *robusto* para sistemas lineales con incertidumbres.
- En el capítulo segundo, se estudia los sistemas lineales con saltos markovianos. En primer lugar, se presentan algunos conceptos básicos relacionados con las cadenas de Markov. Seguidamente, se detalla el procedimiento de diseño de controladores óptimos para MJLS sin restricciones, finalizando el capítulo con una propuesta de controladores para MJLS con incertidumbre, donde el problema de optimización es expresado en términos de desigualdades matriciales lineales (LMI).

- En el capítulo 3 se expone el desarrollo de controladores predictivos para MJLS si perturbaciones sujeto a restricciones duras. Se detalla el procedimiento de obtención del modelo de predicción. Además se definen los conjuntos (terminal y factibles) necesarios para garantizar estabilidad *–en media–* y factibilidad recursiva del problema de optimización. El capítulo se cierra con un breve detalle de controladores predictivos basado en escenario, presentando un algoritmo para la construcción de árboles de escenario que serán usado posteriormente en el problema de optimización.

La segunda parte de la tesis corresponde a las contribuciones de la misma, las cuales se han dividido en:

1) Diseño de controladores predictivos basados en escenarios con políticas *no arriesgadas*.

- **Diseño de controladores predictivos con horizontes múltiples para MJLS:**

Este corresponde al primer capítulo de contribución de la tesis. Aquí, se plantea un primer enfoque de controladores predictivos basado en escenario para MJLS sujeto a restricciones, considerando secuencias de diferentes longitudes. Se proponen conjuntos terminal y factible para asegurar factibilidad y estabilidad en la implementación de la técnica de horizonte deslizante. Aquí, se asume que se conoce el modo de funcionamiento del sistema al momento de calcular la acción de control.

- **Estudio de estabilidad para MPC basado en escenarios:** En este capítulo se generaliza los resultados anteriores. En efecto, se considera que la transición entre los modos de funcionamiento del sistema, se rige por un modelo probabilístico que involucra tanto las probabilidades de observación del modo y variables adicionales *información* que permitirán predicciones sobre las futuras probabilidades de dicho modo. Para este nuevo enfoque, se proponen los ingredientes necesarios para garantizar estabilidad *–en media–* y factibilidad del problema de optimización.

2) Contribuciones para escenarios con políticas arriesgadas, es decir, no se garantiza al 100 % que el estado del sistema alcance el origen o a un cierto conjunto vecindad de este.

- **Análisis de confiabilidad de MPC para MJLS basado en simulación:** Este sexto capítulo trata con MJLS sujeto a restricciones, donde algún modo de funcionamiento genera inestabilidad en el sistema en lazo cerrado. Debido a esto, los conjuntos terminales invariantes pueden ser de

volumen vacío o solo contener el origen en su interior. Por lo tanto, se presenta un algoritmo que calcula la confiabilidad de elección de conjuntos terminales “*probables*”, para luego ser utilizado en la solución del problema de optimización.

- **Cálculo de conjuntos controlables para MJLS sujetos a restricciones:** En el séptimo capítulo se presenta una generalización del concepto de conjunto λ -contractivo para el diseño del máximo conjunto invariante bajo control. Para estados iniciales dentro de este, se puede garantizar que el sistema convergerá *–en media–* al origen el 100 % de las veces. Para aquellos estados que no pertenezcan a este, se proponen la obtención de conjuntos controlables dependientes de secuencia con su ley de control (“*sub-óptima*”) para asegurar una cota de confiabilidad, entendida como la probabilidad de alcanzar el origen sin violar restricciones en el futuro. El término *conjunto controlable dependiente de una secuencia*, debe ser entendido como el conjunto de estados para los cuales existe una acción de control tal que el estado del sistema puede ser dirigido al origen, solo para una realización en particular de todas las disponibles en el árbol de escenario.
- **Cota de confiabilidad y tiempo medio para fallar para MJLS:** En el capítulo 8 se extienden los resultados presentados en el anterior. En primer lugar, se considera la existencia de perturbaciones en el sistema. Se propone un algoritmo, con el que se obtiene una ley de control óptima (en ciertas condiciones). En segundo lugar, para el caso de sistemas no estabilizables, los controladores resultantes del capítulo 7 no resultan aplicables, por tanto, mediante programación dinámica se calcula una cota de tiempo medio de fallo, que se entiende como la probabilidad de violar restricciones en el futuro.

La tesis finaliza con una recopilación de las conclusiones extraídas del trabajo de investigación. Se proponen los trabajos futuros para la continuación de la línea de investigación.

Publicaciones

El trabajo de investigación a lo largo de esta tesis ha dado lugar a varias publicaciones que se listan a continuación:

Artículos en revistas:

- M. Hernández-Mejías, A. Sala, C. Ariño, A. Querol. *Reliable controllable sets for constrained Markov-jump linear systems*. International Journal of Robust and Nonlinear Control, 2016. Volumen 26, páginas 2075-2089. DOI: 10.1002/rnc.3394.
- M. Hernández-Mejías, A. Sala. *Reliability and time-to-failure for discrete-time constrained Markov-jump linear systems*. En segunda revisión.
- M. Hernández-Mejías, A. Sala, C. Ariño. *Ingredients for stable receding-horizon scenario-based predictive control*. En segunda revisión.

Artículos en Congresos:

- M. Hernández-Mejías, A. Sala, C. Ariño, A. Querol. *Fault-Tolerant Predictive Control for Markov Linear Systems*. Conference on Automation, Quality and Testing, Robotics, 2014. Cluj-Napoca, Romania. DOI:10.1109/AQTR.2014.6857824.
- M. Hernández-Mejías, A. Sala, C. Ariño, A. Querol. *Reliable fault-tolerant predictive control for Markov-jump linear systems*. IEEE Conference on Control Applications (CCA), 2014. Pp. 1831-1836. Antibes, Frances. DOI: 10.1109/CCA.2014.6981579.
- M. Hernández-Mejías, A. Sala, C. Ariño, A. Querol. *Análisis de confiabilidad de sistemas lineales ante fallos esporádicos*. Actas de las XXXV Jornadas de Automática, Septiembre de 2014. Valencia, Spain. ISBN-13: 978-84-697-0589-6.

- M. Hernández-Mejías, A. Sala, A. Querol, C. Ariño. *Multiple-horizon predictive control for Markov/switched linear systems*. 5th IFAC Conference on Nonlinear Model Predictive Control (NMPC), 2015. Pp. 230-235. Seville, Spain.
DOI:10.1016/j.ifacol.2015.11.288.

Durante el desarrollo de la tesis también se ha asistido a cursos especializados:

- V International School on Fault Detection and Diagnosis of Complex Systems. Madrid, Spain, July 2013.
- Numerical optimization: concepts, algorithms and applications in process modeling and control. Valencia, Spain, September 2015.

Parte I
Estado del arte

Capítulo 1

Control Predictivo Basado en Modelo

Resumen: El principal objetivo del presente capítulo es el diseño de controladores predictivos basados en modelo para sistemas dinámicos, además del estudio de la estabilidad y factibilidad de los mismos. En primer lugar, se anuncian varias definiciones necesarias para el desarrollo de la tesis, se definen los diferentes sistemas *no lineales* en tiempo discreto. Seguidamente, se hace un resumen de las propiedades de estabilidad de sistemas dinámicos basado en la teoría de Lyapunov. Se presentan conceptos fundamentales de la teoría de conjuntos invariantes utilizados para asegurar la estabilidad del sistema. A continuación, se anuncia el problema de optimización estándar. Finalmente se presenta la formulación del controlador predictivo basado en modelo.

1.1. Teoría de Sistemas

1.1.1. Sistemas en Tiempo Discreto

Definición 1.1. *La variable $k \in \mathbb{N}$ denota el tiempo discreto. Considérese los siguientes sistemas no lineales en tiempo discretos (Besselmann 2010):*

$$x_{k+1} = f(x_k) \quad (\text{Sistemas Autónomos}) \quad (1.1a)$$

$$x_{k+1} = f(x_k, u_k) \quad (\text{Sistemas no Autónomos}) \quad (1.1b)$$

$$x_{k+1} = f(x_k, w_k), \quad x_{k+1} = f(x_k, u_k, w_k) \quad (\text{Sistemas perturbados}) \quad (1.1c)$$

donde $x_k \in \mathbb{R}^{n_x}$, $u_k \in \mathbb{R}^{n_u}$, $w_k \in \mathbb{R}^{n_w}$ representan el estado del sistema, acciones de control y las incertidumbres, respectivamente. Las variables u_k, w_k representan las entradas del sistemas, donde u_k será determinada por un controlador, mientras que w_k es una entrada *externa* a la planta la cual puede ser, en general, perturbaciones, ruido de proceso, incertidumbres en el modelos, modo de operación de operación del sistema.

Definición 1.2. (*Sistemas lineales invariantes en el tiempo (LTIs)*) *Considérese los siguientes sistemas lineales invariantes en el tiempo:*

$$x_{k+1} = Ax_k, \quad (\text{Sistemas LTIs Autónomos}) \quad (1.2a)$$

$$x_{k+1} = Ax_k + Bu_k, \quad (\text{Sistemas LTIs no Autónomos}) \quad (1.2b)$$

$$x_{k+1} = Ax_k + Bu_k + Ew_k \quad (\text{Sistemas LTIs perturbados}) \quad (1.2c)$$

$$x_{k+1} = A(w_k)x_k + B(w_k)u_k \quad (\text{Sistemas LTIs inciertos}) \quad (1.2d)$$

$$(1.2e)$$

donde las matrices $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$ y $E \in \mathbb{R}^{n_x \times n_w}$ son matrices del sistema.

Definición 1.3. (*Sistemas sujeto a restricciones*)(Besselmann 2010) *Un sistema discreto se dice estar sujeto a restricciones si, para todo k :*

$$x_k \in \mathbb{X} \subset \mathbb{R}^{n_x}, \quad (1.3a)$$

$$u_k \in \mathbb{U} \subset \mathbb{R}^{n_u}, \quad (1.3b)$$

$$w_k \in \mathbb{W} \subset \mathbb{R}^{n_w} \quad (1.3c)$$

Nótese que, cada una de las anteriores restricciones son diferentes. Generalmente \mathbb{W} se supone conocida, mientras que \mathbb{U} viene forzada por las saturaciones del sistema (limitaciones físicas en las entradas). Por su parte, \mathbb{X} son las restricciones del estado del proceso y representa un objetivo a conseguir, es decir, si las trayectorias del sistema se salen del conjunto \mathbb{X} (violando restricciones), entonces, tal situación se considera a lo largo de la tesis como *fallo*.

1.1.2. Estabilidad

El análisis de estabilidad para sistemas discretos está basado en la *Teoría de Lyapunov*. En base a ésta, es posible estudiar la estabilidad de un sistema dinámico respecto a un punto de equilibrio. Las definiciones que a continuación se anuncian están basadas en los trabajos (Costa *et al.* 2005, Besselmann 2010).

Definición 1.4. (Estabilidad de Lyapunov) Un punto de equilibrio x_{eq} del sistema autónomo (1.1a) es **estable** en el sentido de Lyapunov, si para cada $\epsilon > 0$ existe un $\delta > 0$, tal que:

$$\|x_0\| < \delta \Rightarrow \|x_k\| < \epsilon, \forall k \in \mathbb{N} \quad (1.4)$$

Definición 1.5. (Asintóticamente estable) El punto de equilibrio x_{eq} de un sistema autónomo (1.1a) es dicho ser **asintóticamente estable**, si se cumple la condición de estabilidad (1.4) y además existe un $\delta > 0$, tal que:

$$\|x_0\| < \delta \Rightarrow \lim_{k \rightarrow \infty} \|x_k\| = 0 \quad (1.5)$$

Definición 1.6. (Exponencialmente estable) El punto de equilibrio x_{eq} de un sistema autónomo (1.1a) es dicho ser **exponencialmente estable**, si existe un $\delta > 0$, $\lambda \in [0, 1)$ y $c \in \mathbb{R}$, tal que:

$$\|x_0\| < \delta \Rightarrow \|x_k\| \leq c\|x_0\|\lambda^k \quad (1.6)$$

Definición 1.7. (Función de Lyapunov) Sea x_{eq} un punto de equilibrio del sistema (1.1a). Una función definida positiva $\mathcal{V} : \Gamma \rightarrow \mathbb{R}_+$, donde $\Gamma \subseteq \mathbb{R}^{n_x}$, $x_{eq} \in \Gamma$, es dicha ser **función de Lyapunov** para el sistema si las siguientes condiciones se cumplen (Costa et al. 2005, Sección 2.4):

- (a) $\mathcal{V}(\cdot)$ es una función continua,
- (b) $\mathcal{V}(0) = 0$,
- (c) $\mathcal{V}(x) > 0$, $\forall x \in \Gamma$ tal que $x \neq 0$,
- (d) $\mathcal{V}(f(x)) - \mathcal{V}(x) \leq 0$, $\forall x \in \Gamma$

Teorema 1.1. (Lyapunov)(Costa et al. 2005) Si existe una función de Lyapunov $\mathcal{V}(x)$ para el sistema autónomo (1.1a), entonces el punto de equilibrio x_{eq} es estable en el sentido de Lyapunov. Adicionalmente, si

$$\mathcal{V}(f(x)) - \mathcal{V}(x) < 0 \quad \forall x \neq 0$$

el origen es asintóticamente estable. Además, si \mathcal{V} es radialmente no acotada, i.e..

$$\mathcal{V}(x) \rightarrow \infty, \quad \text{como} \quad \|x\| \rightarrow \infty,$$

entonces, el origen es globalmente estable, lo que implica que es el único punto de equilibrio del sistema.

Definición 1.8. (Estabilidad cuadrática) El punto de equilibrio x_{eq} de un sistema autónomo (1.1a) es dicho ser **cuadráticamente estable**, si el sistema admite una función de Lyapunov cuadrática $\mathcal{V}(x) = x^T P x$, $P = P^T \succ 0$.

1.2. Teoría de Conjuntos Invariantes

La teoría de conjuntos invariantes desempeña un rol de gran importancia para el diseño de controladores para sistemas sujeto a restricciones (Gilbert and Tan 1991, Blanchini 1994, Kerrigan 2000) (y referencias citadas en estos), así como también, para el análisis de regiones de atracción (Blanchini and Miani 2000, Hu *et al.* 2002, Christophersen 2006). La principal razón de esto es debido a que el cumplimiento de las restricciones puede ser garantizado para todo tiempo futuro, si y solo si, el estado inicial del sistema x_0 se encuentra contenido dentro de un determinado conjunto invariante (Kerrigan 2000).

El problema de la invarianza puede ser generalizado a otros casos, como por ejemplo, definiendo problemas de control donde el estado de un sistema afín sólo deba salir por cierta cara de un conjunto poliédrico (Helwa *et al.* 2016); estas cuestiones no entran dentro de los objetivos de la tesis por lo que no son abordadas.

Dependiendo de la estructura del sistema en estudio, a continuación se presentan diferentes definiciones de conjuntos invariantes, dichas definiciones son extraídas de los trabajos (Blanchini 1999, Kerrigan 2000).

Definición 1.9. (Conjunto positivamente invariante) Un subconjunto del espacio de estados $\Omega \subset \mathbb{R}^{n_x}$ es dicho ser **positivamente invariante** para el sistema autónomo $x_{k+1} = f(x_k)$ (1.1a) si y solo si $\forall x_0 \in \Omega$, la evolución del sistema satisface $x_k \in \Omega$, $\forall k \in \mathbb{Z}_+$.

Observación 1.1. Otros autores, por ejemplo, (Kerrigan 2000) hacen distinción entre “positivamente” invariante e “invariante”, i.e., el conjunto Ω es **invariante** si y solo si $x_0 \in \Omega$ implica que $x_k \in \Omega$, $\forall k \in \mathbb{Z}$. Sin embargo por brevedad, a lo largo de esta tesis, cuando se mencione la característica “invariante”, en realidad, se está haciendo referencia a la definición 1.9.

Definición 1.10. (Conjunto robustamente invariante) Un conjunto $\Omega \subset \mathbb{R}^{n_x}$ es **robustamente invariante** para el sistema $x_{k+1} = f(x_k, w_k)$ si y solo si:

$$x_k \in \Omega \Rightarrow x_{k+1} \in \Omega, \forall w_k \in \mathbb{W}, \forall k \in \mathbb{Z}_+ \quad (1.7)$$

Definición 1.11. (Conjunto invariante bajo control) Un subconjunto del espacio de estados $\Omega \subset \mathbb{R}^{n_x}$ es **invariante bajo control** para el sistema $x_{k+1} = f(x_k, u_k)$

(1.1b) si y solo si existe una ley de control por realimentación $u_k = g(x_k)$ tal que, Ω es un conjunto invariante para el sistema en lazo cerrado $x_{k+1} = f(x_k, g(x_k))$ y $u_k \in \mathbb{U}, \forall x_k \in \Omega$.

Definición 1.12. (Máximo conjunto invariante bajo control) El conjunto no vacío $\mathcal{C}_\infty(\Omega)$ es el **máximo conjunto invariante bajo control** contenido en Ω para el sistema $x_{k+1} = f(x_k, u_k)$ si y solo si $\mathcal{C}_\infty(\Omega)$ es un invariante bajo control y contiene a todos los conjunto invariante bajo control contenido en Ω .

Definición 1.13. (Conjunto a un paso (Kerrigan 2000)) Considérese un subconjunto $\Omega \subset \mathbb{R}^{n_x}$, el **conjunto a un paso** $\mathcal{Q}(\Omega)$ es el conjunto de estados en \mathbb{R}^{n_x} para los cuales existe una acción de control u_k admisible que conducirá al sistema a Ω en un solo paso, i.e.:

$$\mathcal{Q}(\Omega) := \{x_k \in \mathbb{R}^{n_x} \mid \exists u_k \in \mathbb{U} \text{ tal que } f(x_k, u_k) \in \Omega\} \quad (1.8)$$

El conjunto a un paso $\mathcal{Q}(\Omega)$ tiene las siguientes propiedades: (a) para todo Ω_1, Ω_2 con $\Omega_1 \subseteq \Omega_2$, $\mathcal{Q}(\Omega_1) \subseteq \mathcal{Q}(\Omega_2)$, (b) un conjunto Ω es positivamente invariante bajo control si y solo si $\mathcal{Q}(\Omega) \cap \Omega = \Omega$ (Kerrigan and Maciejowski 2000, Proposición 2.1).

Definición 1.14. (Conjunto controlable a N pasos) Considérese un subconjunto $\Omega \subset \mathbb{R}^{n_x}$, el **conjunto controlable a N pasos** $\mathcal{K}_N(\Omega, \mathcal{C})$ para el modelo $x_{k+1} = f(x_k, u_k)$, es el conjunto de estados en Ω que pueden ser dirigidos por una secuencia de acciones de control admisible a un conjunto arbitrario $\mathcal{C} \subset \mathbb{R}^{n_x}$ en exactamente N pasos, mientras la trayectoria o evolución de los estados permanece dentro de Ω para los primeros $N - 1$ pasos, i.e.:

$$\mathcal{K}_N(\Omega, \mathcal{C}) := \{x_0 \in \mathbb{R}^{n_x} \mid \exists \{u_k \in \mathbb{U}\}_{k=0}^{N-1} : \{x_k \in \Omega\}_{k=0}^{N-1}, x_N \in \mathcal{C} \subset \mathbb{R}^{n_x}\} \quad (1.9)$$

Definición 1.15. (Conjunto robustamente invariante bajo control) Un subconjunto del espacio de estados $\Omega \subset \mathbb{R}^{n_x}$ es un **conjunto robustamente invariante bajo control** para el sistema $x_{k+1} = f(x_k, u_k, w_k)$ (1.1c) si y solo si existe una ley de control admisible $u_k = g(x_k)$, tal que, Ω es un conjunto robusto positivamente invariante para el sistema en lazo cerrado $x_{k+1} = f(x_k, g(x_k), w_k)$, i.e.:

$$x_k \in \Omega \Rightarrow \exists u_k \in \mathbb{U} \mid x_{k+1} \in \Omega, \forall w_k \in \mathbb{W} \quad (1.10)$$

Definición 1.16. (Conjunto robustamente controlable a N pasos) El **conjunto robustamente controlable a N pasos** $\tilde{\mathcal{K}}_N(\Omega, \mathcal{C})$ para el sistema $x_{k+1} = f(x_k, u_k, w_k)$, es el conjunto de estados en Ω para los cuales existe una ley de control admisible variante en el tiempo, tal que, un conjunto arbitrario $\mathcal{C} \subset \mathbb{R}^{n_x}$ es alcanzado en exac-

tamente N pasos, mientras la evolución del sistema permanece dentro de Ω en los primeros $N - 1$ pasos, para toda secuencia de perturbación permitida, i.e.:

$$\tilde{\mathcal{K}}_N(\Omega, \mathcal{C}) := \{x_0 \in \mathbb{R}^{n_x} \mid \exists \{u_k \in \mathbb{U}\}_{k=0}^{N-1} : \{x_k \in \Omega\}_{k=0}^{N-1}, x_N \in \mathcal{C}, \forall \{w_k \in \mathbb{W}\}_0^{N-1}\} \quad (1.11)$$

A continuación se presenta el algoritmo usado para la obtención del máximo conjunto invariante bajo control, el cual emplea la Definición 1.13:

Algoritmo 1 Cálculo de conjuntos controlables.

Entrada: \mathcal{C}, N

- 1: Hacer $i = 0, \mathcal{K}_0(\Omega, \mathcal{C}) = \mathcal{C}$
 - 2: **mientras** $i < N$ **hacer**
 - 3: $\mathcal{K}_{i+1}(\Omega, \mathcal{C}) = \mathcal{Q}(\mathcal{K}_i(\Omega, \mathcal{C})) \cap \Omega$
 - 4: Si $\mathcal{K}_{i+1}(\Omega, \mathcal{C}) = \mathcal{K}_i(\Omega, \mathcal{C})$, finalizar el algoritmo y $\mathcal{K}_\infty(\Omega) = \mathcal{K}_i(\Omega, \mathcal{C})$
 - 5: $i = i + 1$
 - 6: **fin mientras**
-

Observación 1.2. Aunque el Algoritmo 1 puede ser fácilmente implementado en sistemas lineales invariantes en el tiempo (LTI) sujeto a restricciones lineales (Blanchini 1994, Gilbert and Tan 1991, Dórea and Hennet 1999), resulta difícil de implementar para los sistemas no lineales en general². Sin embargo, existen diferentes trabajos en donde se estudian la teoría de conjuntos invariantes para sistemas no lineales, por ejemplo, en (Alamo et al. 2005) se presenta un método que garantiza la estimación del estado de un sistema no lineal en tiempo discreto con perturbaciones acotadas que afectan a los estados y las salidas. Se plantea un algoritmo que calcula un conjunto el cual contiene los estados que son consistentes con las salidas del sistema. Este conjunto es representado mediante un zonotopo. Por su parte, el trabajo (Bravo et al. 2005) se basan en la obtención de una aproximación interior $\mathcal{Q}_{ap}(\Omega)$ del operador a un paso $\mathcal{Q}(\Omega)$, i.e., $\mathcal{Q}_{ap}(\Omega) \subseteq \mathcal{Q}(\Omega)$, empleando la aritmética intervalar para el cálculo de dichos conjuntos. Además, existen numerosas contribuciones en cuanto a la computación de conjuntos invariantes para sistemas híbridos, en particular, para sistemas afines a trozos (Raković et al. 2004, Rubagotti et al. 2013, Åström and Wittenmark 2013). También existen formulaciones para la obtención de conjuntos invariantes para sistemas borroso en tiempo discreto (Ariño et al. 2013). En el caso de sistemas no lineales con perturbaciones, el análisis resulta más complicado y conservativo; se remite al lector a (Pitarch et al. 2015, Sala and Pitarch 2016) y las referencias ahí contenidas para detalles sobre estos aspectos, fuera de los objetivos de la presente tesis.

²Esto se debe a que la no linealidad presente en el sistema y/o en el controlador resultan en conjuntos no convexos con formas muy complejas, ocasionando a menudo una gran complejidad computacional.

Definición 1.17. (Conjunto contractivo (Blanchini 1999)) Un conjunto $\Omega \subset \mathbb{R}^{n_x}$ es **contractivo** para el sistema $x_{k+1} = f(x_k, u_k)$ (1.1b) si y solo si existe una ley de control $u(x_k)$ y un número entero positivo $\lambda \leq 1$, tal que, si $x_k \in \Omega$, entonces $x_{k+1} \in \lambda\Omega$, donde $\lambda\Omega$ representa un escalado del conjunto Ω .

1.3. Politopos

Definición 1.18. (Conjuntos convexos (Boyd and Vandenberghe 2004)) Un conjunto $C \subset \mathbb{R}^{n_x}$ es **convexo** si el segmento de línea entre dos puntos cualquiera en C pertenece en su totalidad al conjunto C , es decir, si para cualquier $s_1, s_2 \in C$ y cualquier α con $0 \leq \alpha \leq 1$, se cumple:

$$\alpha s_1 + (1 - \alpha)s_2 \in C \quad (1.12)$$

Definición 1.1. (Funciones convexas (Boyd and Vandenberghe 2004)) Una función $f : C \subseteq \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ es dicha ser **convexa** si y solo si, $\forall s_1 \in C, \forall s_2 \in C$ y $\alpha \in (0, 1]$, se tiene:

$$f(\alpha s_1 + (1 - \alpha)s_2) \leq \alpha f(s_1) + (1 - \alpha)f(s_2) \quad (1.13)$$

Definición 1.2. (Envolvente convexa)(Boyd and Vandenberghe 2004) La envolvente convexa de un conjunto C , denotada como $\text{conv}(C)$, es conjunto de todas las combinaciones convexas de puntos pertenecientes a C , i.e.:

$$\text{conv}(C) := \left\{ \sum_{i=1}^k \alpha_i s_i \mid s_i \in C, \alpha_i \geq 0, i = 1, \dots, k, \sum_{i=1}^k \alpha_i = 1, k \in \mathbb{R} \right\}$$

Definición 1.19. (Poliedro) Un poliedro $\mathcal{P} \subset \mathbb{R}^{n_x}$ es un conjunto convexo que puede ser representado como la intersección de un número finito de semi espacios cerrados, i.e.,

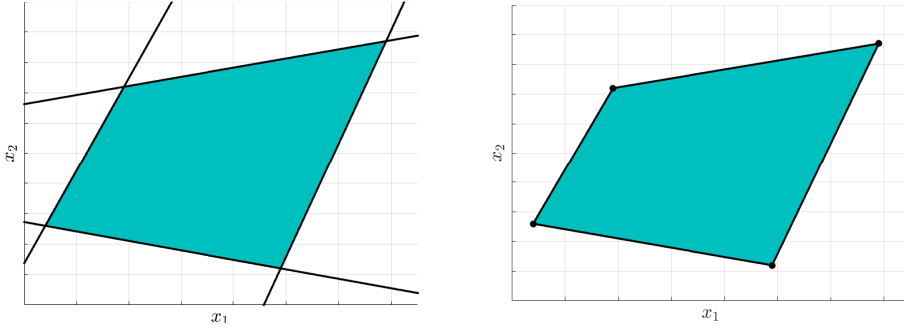
$$\mathcal{P} := \{x \in \mathbb{R}^{n_x} \mid P^x x \leq P^0\}, \quad (1.14)$$

donde $P^0 \in \mathbb{R}^q$, $P^x \in \mathbb{R}^{q \times n_x}$, con q denotando el número de semi espacios que definen al poliedro \mathcal{P} .

Definición 1.20. (Politopo) Un politopo se define como un poliedro cerrado.

Teorema 1.2. (Representación de un politopo) (Christophersen 2006) $\mathcal{P} \subset \mathbb{R}^{n_x}$ es la envolvente convexa de un conjunto de vértices $V := \{v_1, \dots, v_{N_v}\}$, $v_i \in \mathbb{R}^{n_x}$ con $i = 1, \dots, N_v$ (V -politopo)

$$\mathcal{P} := \text{conv}(V) = \left\{ x \in \mathbb{R}^{n_x} \mid x = \sum_{i=1}^{N_v} \alpha_i v_i, 0 \leq \alpha_i \leq 1, \sum_{i=1}^{N_v} \alpha_i = 1 \right\} \quad (1.15)$$



(a) H-politopo. Politopo construido a partir de 4 hiperplanos pintados como líneas.

(b) V-politopo. Representación de un politopo a través de 4 vértices pintados en puntos negros.

Figura 1-1: Ilustración de la representación de un politopo.

si y solo si se trata de una intersección acotada de hiperplanos (H-politopo)

$$\mathcal{P} := \{x \in \mathbb{R}^{n_x} \mid P^x x \leq P^0\} \quad (1.16)$$

El teorema anterior establece que un politopo cualquiera \mathcal{P} puede ser expresado, bien por sus *vértices* (V-politopo) o por un número de hiperplanos (H-politopo) (Figura 1-1).

Nótese que en la Figura 1-1a ninguna de las desigualdades (hiperplanos) que forman al politopo puede ser eliminada sin cambiar la estructura del mismo. Sin embargo existen casos en donde pueden ser eliminadas ciertas desigualdades llamadas *redundantes* sin cambiar la forma del poliedro. Este tipo de operación se encuentran fuera de los objetivos de esta tesis. Sin embargo, se remite al lector trabajo (Borrelli *et al.* 2014, Sección 5.4.1) donde se presenta un algoritmo que calcula la representación *mínima* de un politopo³, eliminando los hiperplanos redundantes, todo esto basado en programación lineal.

Observación 1.3. En particular, cuando las restricciones de estados (1.3a) y acciones de control (1.3b) sean lineales, estas pueden ser definidas en forma poliédrica, i.e.:

$$\mathbb{X} := \{x \in \mathbb{R}^{n_x} \mid P^x x \leq P^0\} \quad (1.17a)$$

$$\mathbb{U} := \{u \in \mathbb{R}^{n_u} \mid S^u u \leq S^0\} \quad (1.17b)$$

donde P^0 , S^0 y P^x , S^u son vectores y matrices de apropiadas dimensiones, respectivamente.

³La H-representación de un poliedro es *mínima* si no contiene desigualdades redundantes.

Definición 1.21. (Colección de politopos) Se define como \mathcal{P} -colección denotado como \mathcal{P}_c , a la unión (posiblemente no convexa) de un número finito de politopos \mathcal{P} .

Para un detalle exhaustivo sobre operaciones y manipulaciones entre poliedros se remite al lector a consultar los trabajos (Herceg *et al.* 2013, Borrelli *et al.* 2014).

Definición 1.22. (Proyección) Dado un politopo $\mathcal{P} = \{x \in \mathbb{R}^{n_x}, y \in \mathbb{R}^{n_y} \mid P^x x + P^y y \leq P^0\} \subset \mathbb{R}^{n_x+n_y}$, la proyección ortogonal sobre el x -espacio \mathbb{R}^{n_x} se define como:

$$\text{proj}_x(\mathcal{P}) := \{x \in \mathbb{R}^{n_x} \mid \exists y \in \mathbb{R}^{n_y} \text{ sujeto a } P^x x + P^y y \leq P^0\} \quad (1.18)$$

Definición 1.23. (Diferencia) La diferencia (resta) de dos politopos \mathcal{P} y \mathcal{D} se define como:

$$\mathcal{R} := \mathcal{P} \setminus \mathcal{D} := \{x \in \mathbb{R}^{n_x} \mid x \in \mathcal{P}, x \notin \mathcal{D}\} \quad (1.19)$$

Definición 1.24. (Diferencia de Pontryagin) La diferencia de Pontryagin de dos politopos \mathcal{P} y \mathcal{W} es un politopo:

$$\mathcal{P} \ominus \mathcal{W} := \{x \in \mathbb{R}^{n_x} \mid x + w \in \mathcal{P}, \forall w \in \mathcal{W}\} \quad (1.20)$$

Definición 1.25. (Suma de Minkowski) La suma de Minkowski de dos politopos \mathcal{P} y \mathcal{W} es un politopo:

$$\mathcal{P} \oplus \mathcal{W} := \{x + w \in \mathbb{R}^{n_x} \mid x \in \mathcal{P}, w \in \mathcal{W}\} \quad (1.21)$$

Calculo de conjuntos invariantes. A continuación se presenta un ejemplo de la aplicación del Algoritmo 1 para un sistema lineal invariante en el tiempo sujeto restricciones lineales de la forma (1.17a). Se hace uso del *Multi-parametric Toolbox (MPT)* (Herceg *et al.* 2013) para el cálculo del conjunto invariante dada por la Definición (1.9).

Ejemplo 1.1. Considérese un sistema lineal de segundo orden dado por:

$$x_{k+1} = Ax_k = \begin{bmatrix} 0,5 & 0 \\ 1 & 0,75 \end{bmatrix} \quad (1.22)$$

sujeto a las siguientes restricciones de estados:

$$x_k \in \mathbb{X} \subseteq \mathbb{R}^2 = \left\{ x \mid \begin{bmatrix} -10 \\ -10 \end{bmatrix} \leq x \leq \begin{bmatrix} 10 \\ 10 \end{bmatrix} \right\}, \quad \forall k \geq 0 \quad (1.23)$$

Las restricciones (1.23) puede representarse de la forma (1.17a), con:

$$P^x = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}, \quad P^0 = \begin{bmatrix} 10 \\ 10 \\ 10 \\ 10 \end{bmatrix}$$

El paso 3 del Algoritmo 1 usa la Definición 1.13 para obtener el conjunto a un paso, i.e.:

$$\mathcal{Q}(\mathbb{X}) = \{x \mid P^x Ax \leq P^0\} \quad (1.24)$$

Una vez obtenido dicho conjunto se procede a interceptarlo con \mathbb{X} , i.e.:

$$\mathcal{Q}(\mathbb{X}) \cap \mathbb{X}$$

para así obtener el conjunto de estado que en un paso alcanzan a \mathbb{X} . Tenga en cuenta que el conjunto resultante de la operación de intersección anterior puede contener desigualdades redundantes. De hecho usando el comando (`minHrep`) en MPT, es posible obtener el conjunto en representación mínima (sin desigualdades redundante), que está dado por:

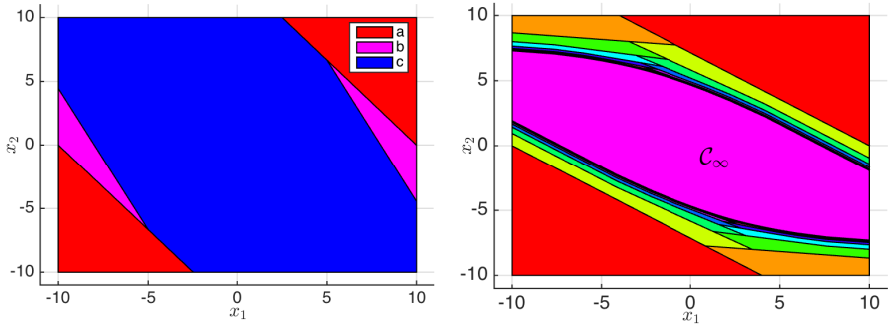
$$\mathcal{Q}(\mathbb{X}) \cap \mathbb{X} = \left\{ x \mid \begin{bmatrix} 1 & 0,75 \\ -1 & -0,75 \\ 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} x \leq \begin{bmatrix} 10 \\ 10 \\ 10 \\ 10 \\ 10 \\ 10 \end{bmatrix} \right\}$$

A continuación (paso 4 del algoritmo) se comprueba si el nuevo conjunto $\mathcal{Q}(\mathbb{X}) \cap \mathbb{X}$ es igual a \mathbb{X} . Puede observarse en la Figura 1-2a que $\mathcal{Q}(\mathbb{X}) \cap \mathbb{X} \subset \mathbb{X}$, por consiguiente, se realiza otra iteración obteniendo el conjunto en azul mostrado en la figura, i.e.:

$$\mathcal{Q}(\mathcal{Q}(\mathbb{X}) \cap \mathbb{X}) = \left\{ x \mid \begin{bmatrix} P^x A \\ P^x A^2 \end{bmatrix} x \leq \begin{bmatrix} P^0 \\ P^0 \end{bmatrix} \right\}$$

La iteraciones se repiten hasta que la condición del paso 4 sea cumplida, para este caso en concreto en $i = 3$.

Ejemplo 1.2. El siguiente ejemplo presenta la implementación del Algoritmo (1) con el fin de calcular el máximo conjunto positivamente invariante bajo control \mathcal{C}_∞ para



(a) a) conjunto de restricciones de estado X , b)(b) Mximo conjunto invariante bajo control C_∞
 $\mathcal{K}_1 = \mathcal{Q}(X) \cap X$, c) $\mathcal{K}_2 = \mathcal{Q}(\mathcal{K}_1) \cap X$ para el ejemplo 1.2

Figura 1-2: Ejemplos de cculo de conjuntos invariantes.

un sistema LTI de segundo orden dado por:

$$x_{k+1} = Ax_k + Bu_k = \begin{bmatrix} -1,2 & 0 \\ 1 & 1,4 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_k$$

sujeto a las restricciones de estado (1.23) y a la de control:

$$u_k \in \mathbb{U} = \{u \mid -5 \leq u \leq 5\}$$

Para el cculo del conjunto C_∞ usando el Algoritmo 1 es necesario reemplazar las iteraciones con:

$$C_i(X) = \mathcal{K}_i(X, X),$$

si $C_i(X) = C_{i+1}(X)$, entonces, $C_\infty(X) = C_i(X)$. Luego de un total de 38 iteraciones del algoritmo, se obtiene el conjunto $C_\infty(X)$ mostrado en la figura 1-2b

Observaci3n 1.4. La operaciones de comprobaci3n de igualdad entre dos conjunto y de proyecci3n dada por la Defini3n 1.22 requiere –en general– gran capacidad de recursos computacionales, debido al hecho de que en dichas operaciones debe obtenerse el conjunto de vrtices de los politopos. El esfuerzo computacional de esta operaci3n es exponencial en el nmero de ecuaciones de desigualdad que forman al politopo (Borrelli et al. 2014, Secci3n 5.4.4).

1.4. Problema de Optimización Estándar

Un problema de optimización puede ser representado por las siguientes ecuaciones (Boyd and Vandenberghe 2004):

$$\text{mín } f_0(x) \tag{1.25a}$$

$$\text{s.t. } f_i(x) \leq 0, \quad i = 1, \dots, m, \tag{1.25b}$$

$$h_j(x) = 0 \quad j = 1, \dots, p \tag{1.25c}$$

$$\tag{1.25d}$$

Las ecuaciones anteriores, describen el problema de encontrar un x que minimice $f_0(x)$ entre todas las x que satisfacen las condiciones $f_i(x) \leq 0$, $i = 1, \dots, m$, y $h_j(x) = 0$ $j = 1, \dots, p$. $x \in \mathbb{R}^{n_x}$ se conocen como las *variable de decisión* y $f_0(x) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ es la función objetivo (comúnmente llamada *índice de coste*). Las desigualdades $f_i(x) \leq 0$, $f_i : \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ representan las *restricciones de desigualdad*, mientras que $h_j(x) = 0$, $h_j : \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ son las *restricciones de igualdad*⁴.

El conjunto de puntos para los que se definen la función objetivo y todas las restricciones,

$$\mathcal{D} = \bigcap_{i=0}^m \text{dom } f_i \cap \bigcap_{j=1}^p \text{dom } h_j,$$

se conoce como *dominio* del problema de optimización (1.25a). Entonces, un punto $x \in \mathcal{D}$ es *factible* si satisface las restricciones (1.25b)–(1.25c).

Dependiendo de la forma de la función de coste (1.25a) y las restricciones (1.25b)–(1.25c), los problemas de optimización pueden ser clasificados en dos grandes categorías:

- Optimización convexa (Boyd and Vandenberghe 2004),
- Optimización no convexa (Bertsekas 1999, Horst and Pardalos 2013).

En optimización convexa se tiene una única solución óptima (si existe) y es *global*, mientras que en optimización no convexa puede haber múltiple punto óptimos locales, y resulta muy difícil –en general– encontrar el óptimo global. En los problemas de optimización convexa resulta “fácil” de comprobar si existe o no una solución factible del problema (1.25a), mientras que en la no convexa puede tomar mucho tiempo identificar si el problema es factible o no.

⁴Un problema de optimización es dicho ser *sin restricciones* si m y p son iguales a 0, i.e., $m = p = 0$

A lo largo del presente trabajo se trata con problema de optimización donde la función de coste (1.25a) y las restricciones (1.25b)–(1.25c) son *convexas*.

1.5. Controladores Predictivos

En esta sección se presenta el *control predictivo basado en modelo* (MPC), frecuentemente llamado *Control con horizonte móvil o deslizante* (RHC). Es bien conocido que el término *control predictivo basado en modelo* no designa una estrategia específica de control, sino más bien a un amplio conjunto de métodos de control que hace uso de un modelo para poder calcular la acción de control minimizando una determinada función de coste (Camacho and Bordons 2010). En MPC, las acciones de control a ser aplicada al sistema son obtenidas de la solución de un problema de optimización numérico que hace uso explícito de un modelo de predicción sobre un horizonte de tiempo finito.

1.5.1. Breve Historia sobre MPC

El desarrollo de los conceptos modernos de control puede hacer referencia al importante trabajo de Kalman a principios de 1960 con el regulador lineal cuadrático (LQR) (Kalman *et al.* 1960). El concepto básico de MPC fue introducido por primera vez por Propoi (Propoi 1963). Sin embargo, fue en la década de los 70 y 80 cuando ganó importancia gracias a los trabajos *control predictivo heurístico* (Richalet *et al.* 1978) y *Control por matriz dinámica* (DMC) (Cutler and Ramaker 1980). Durante este mismo periodo, diversas técnicas fueron desarrolladas para sistemas multivariable, tales como, *Control por matriz dinámica cuadrática* (QDMC), *control predictivo generalizado* (GPC), *control predictivo adaptativo* (APC), *etc.* En el trabajo (Qin and Badgwell 2003) se presenta un excelente resumen de controladores predictivo aplicado a mundo industrial. Los primeros algoritmos y aplicaciones exitosas aparecen en los trabajos (Richalet *et al.* 1976, Richalet *et al.* 1978, Cutler and Ramaker 1980). Sin embargo el gran problema detrás de las diferentes formulaciones citadas anteriormente, es que no garantizaban estabilidad del sistema bajo control. Los primeros trabajos en donde se estudió la estabilidad de sistemas controlados con MPC son, entre otros, (Kothare *et al.* 1996, Rossiter *et al.* 1998, Scokaert and Mayne 1998, Mayne *et al.* 2000, Goodwin *et al.* 2006). Más reciente, han sido publicado al menos 4 libros de gran impacto (Rawlings and Mayne 2009, Grüne and Pannek 2011, Camacho and Bordons 2013, Kouvaritakis and Cannon 2015) y miles de artículos relacionados con diferentes formulaciones y aplicación de

controladores predictivos.

El control predictivo ha alcanzado una gran popularidad e interés en aplicaciones industriales, desde sus inicios (Richalet *et al.* 1978) hasta (Berenguel *et al.* 2004, Blasco *et al.* 2007, Ramirez *et al.* 2008, Calle-Prado *et al.* 2015, Kouro *et al.* 2015) por su capacidad de manejar modelos multivariables, restricciones y retardos siendo su actuación comprensible por un operario sin grandes conocimientos en control. Estas razones lo convierten en la alternativa presente y futura más sólida al control PID en aplicaciones de control de procesos.

1.5.2. Idea Básica del MPC

La idea básica detrás de la técnica MPC está basada en la resolución *–en línea–* en cada instante de tiempo, un problema de optimización con el fin de calcular una secuencia de acciones de control, usando el estado actual de la planta y aplicando solo el primer elemento de tal secuencia al sistema. En la próxima iteración, el cálculo es repetido con el nuevo estado medido sobre un horizonte desplazado (Mayne *et al.* 2000). Los algoritmos MPC tienen los siguientes ingredientes: a) un modelo de predicción del sistema, b) función de coste la cual indica el criterio a optimizar y c) restricciones. Para un detalle en profundidad de cada elemento, se remite el lector al libro (Camacho and Bordons 2013).

Dado un sistema dinámico (posiblemente no lineal) en tiempo discreto $x_{k+1} = f(x_k, u_k)$, un MPC requiere solucionar, en cada instante de tiempo, un *problema de control óptimo de horizonte finito* definido por (Goodwin *et al.* 2006, Capítulo 4):

$$J_N^{opt}(x) := \min J_N(\{x_k\}, \{u_k\}) \quad (1.26a)$$

sujeto a:

$$x_{k+1} = f(x_k, u_k) \quad \text{para } k = 0, \dots, N-1, \quad (1.26b)$$

$$x_0 = x, \quad (1.26c)$$

$$u_k \in \mathbb{U} \quad \text{para } k = 0, \dots, N-1, \quad (1.26d)$$

$$x_k \in \mathbb{X} \quad \text{para } k = 0, \dots, N, \quad (1.26e)$$

$$x_N \in \mathbb{X}_f \subset \mathbb{X}, \quad (1.26f)$$

donde:

$$J_N(\{x_k\}, \{u_k\}) := F(x_N) + \sum_{k=0}^{N-1} L(x_k, u_k) \quad (1.27)$$

Por su parte, $\{x_k\}$, $x_k \in \mathbb{R}^{n_x}$, $\{u_k\}$, $u_k \in \mathbb{R}^{n_u}$ denotan las secuencias de estados y acciones de control $\{x_0, \dots, x_N\}$ y $\{u_0, \dots, u_{N-1}\}$, respectivamente. Las funciones F y L se conocen como: *función de penalización terminal del estado* (coste terminal) y *penalización del estado transitorio* (coste de etapa), respectivamente. Se asume que f , F y L son funciones continuas en sus argumentos. Por su parte, \mathbb{U} es un conjunto compacto (cerrado y acotado), mientras que \mathbb{X} , \mathbb{X}_f son conjuntos cerrados. \mathbb{X}_f se conoce como *región terminal*.

Las opciones típicas para las funciones de ponderación F y L son funciones cuadráticas, *i.e.*, $F(x) = x^T P x$ y $L(x, u) = x^T Q x + u^T R u$, donde $P = P^T \geq 0$, $Q = Q^T \geq 0$ y $R = R^T > 0$. Más general, se podría usar funciones de la forma: $F(x) = \|P x\|_p$ y $L(x, u) = \|Q x\|_p + \|R u\|_p$, donde $\|y\|_p$, $p = 1, 2, \dots, \infty$, es la norma- p del vector y , (Goodwin *et al.* 2006).

La solución del problema de optimización formulado anteriormente arroja una secuencia de acción de control, la cual es función del estado actual x_k de la forma:

$$\mathbf{u}^{opt}(x) := \{u_0^{opt}, u_1^{opt}, \dots, u_{N-1}^{opt}\} \quad (1.28)$$

Por consiguiente, la acción a implementar en el proceso para el instante k es el primer elemento de esta secuencia, *i.e.*,

$$u_k = u_0^{opt} \quad (1.29)$$

Utilizando la técnica de horizonte deslizante implícitamente se define una política de control invariante en el tiempo de la forma:

$$K_N(x) := u_0^{opt} \quad (1.30)$$

Definición 1.26. (*Conjunto de estados iniciales factibles*)(Goodwin *et al.* 2006) Sea \mathbb{S}_N el **conjunto de estados iniciales factibles** $x \in \mathbb{X}$ para los cuales existen secuencias de estados y acciones de control, para el problema de control óptimo con horizonte fijo (1.26).

1.5.3. Estabilidad y Factibilidad en MPC

En primer lugar, nótese que el sistema en lazo cerrado formado por el proceso $x_{k+1} = f(x_k, u_k)$ y el controlador predictivo dado por el problema (1.26) con acción de control (1.30), generalmente tiene un comportamiento no lineal, incluso cuando se trata con sistemas LTI. Una de las herramienta frecuentemente utilizada para el análisis de estabilidad en sistemas dinámicos cuya ley de control está calculada por un MPC, es la estabilidad en el sentido de

Lyapunov. Pero existe una dificultad, el problema de optimización (1.26) que se pretende resolver en cada instante de muestreo, está definido para un *horizonte finito*, sin embargo, la estabilidad es una propiedad que debe cumplirse sobre un *horizonte infinito*. Una forma muy utilizada para resolver este conflicto, es añadir una función de penalización terminal adecuada sobre el estado del sistema en el problema a horizonte finito (1.26).

Siguiendo esta línea de razonamiento, en los trabajos (Mayne *et al.* 2000, Goodwin *et al.* 2006) se proponen tres ingredientes $(\mathbb{X}_f, K_f(x), F(x))$ básicos los cuales sin incorporado en la formulación del MPC:

- (I) La región terminal $\mathbb{X}_f \subset \mathbb{X}$ debe ser un conjunto invariante bajo la ley de control terminal.
- (II) Una ley de control terminal $u_k := K(x_k), \forall x_k \in \mathbb{X}_f$.
- (III) Una ponderación terminal del estado $F(x)$, que habitualmente corresponde con el valor del índice de coste generado por el uso de la ley de control terminal sobre un horizonte infinito.

Resumiendo las propuestas más destacada en cuanto al estudio de la estabilidad del MPC, en (Mayne *et al.* 2000, Goodwin *et al.* 2006) se exponen las siguientes condiciones *suficientes* de estabilidad para el problema (1.26):

1. El coste de etapa $L(x, u)$ en (1.27) satisface $L(0, 0) = 0$ y $L(x, u) \geq \gamma(\|x\|)$ para todo $x \in \mathbb{S}_N, u \in \mathbb{U}$, donde $\gamma : [0, \infty) \rightarrow [0, \infty)$ es continua, $\gamma(t) > 0$ para todo $t > 0$, y $\lim_{t \rightarrow \infty} \gamma(t) = \infty$.
2. El coste terminal del estado $F(x)$ en (1.27) satisface $F(0) = 0, F(x) \geq 0, \forall x \in \mathbb{X}_f$ y cumple la siguiente propiedad: existe una ley de control terminal $K_f : \mathbb{X}_f \rightarrow \mathbb{U}$, tal que,

$$F(f(x, K_f(x))) - F(x) \leq -L(x, K_f(x)), \quad x \in \mathbb{X}_f$$

3. EL conjunto terminal \mathbb{X}_f es invariante para el sistema $x_{k+1} = f(x_k, u_k)$ bajo $K_f(x)$, es decir, $x_{k+1} = f(x, K_f(x)) \in \mathbb{X}_f$ para todo $x \in \mathbb{X}_f$.
4. La ley de control terminal $K(x)$ satisface las restricciones en \mathbb{X}_f , i.e., $K_f(x) \in \mathbb{U}, \forall x \in \mathbb{X}_f$.
5. Los conjuntos \mathbb{X}, \mathbb{U} contienen el origen en su interior.

Dadas las condiciones de estabilidad, es posible establecer el siguiente teorema:

Teorema 1.3. (Goodwin et al. 2006) *Considérese el sistema (posiblemente no lineal):*

$$x_{k+1} = f(x_k, u_k), \quad k \geq 0, \quad f(0, 0) = 0, \quad (1.31)$$

controlado por MPC (1.26) con ley de control (1.29). Supóngose que se satisfacen todas las condiciones enunciadas con anterioridad. Entonces:

1. El conjunto \mathbb{S}_N de estados iniciales factibles es invariante para el sistema en lazo cerrado.
2. El origen es globalmente estable en \mathbb{S}_N para el sistema en lazo cerrado.
3. Si, además de las condiciones 1–5, $0 \in \text{int } \mathbb{S}_N$ y el índice de coste J_N^{opt} es continuo en algún conjunto vecino al origen, entonces dicho origen es asintóticamente estable en \mathbb{S}_N para el sistema en lazo cerrado.

1.5.4. MPC para Sistemas LTI

Considérese un sistema LTI dado por (1.2b), *i.e.*:

$$x_{k+1} = Ax_k + Bu_k \quad (1.32)$$

Sopóngase que las restricciones $\mathbb{X}, \mathbb{U}, \mathbb{X}_f$, son poliédricas. Asumiendo que las funciones de penalización de etapa y terminal son cuadráticas, es decir:

$$L(x_k, u_k) := x_k^T Q x_k + u_k^T R u_k, \quad F(x_N) := x_N^T P x_N$$

donde $Q = Q^T \succeq 0$ es una matriz semidefinida positiva de apropiada dimensión, mientras $R = R^T \succ 0$ es una matriz definida positiva y representan las penalizaciones del estado y acciones de control, respectivamente. La función de coste (1.27) se puede escribir de la forma:

$$J_N(\{x_k\}, \{u_k\}) := x_N^T P x_N + \sum_{k=0}^{N-1} x_k^T Q x_k + u_k^T R u_k \quad (1.33)$$

donde $P = P^T \succeq 0$ es una la solución de una ecuación algebraica de Riccati.

Aplicando repetidamente la ecuación (1.32) hasta un horizonte N , se tiene:

$$\underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}}_{\mathcal{X}} = \underbrace{\begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}}_{\bar{T}} x_0 + \underbrace{\begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2} & \cdots & B \end{bmatrix}}_{\bar{S}} \cdot \underbrace{\begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}}_{\mathbf{u}} \quad (1.34)$$

Nótese en la ecuación anterior que, todos los futuros estados son funciones explícitas del estado inicial x_0 y de las futuras entradas $\{u_0, u_1, \dots, u_{N-1}\}$. Por simplicidad, la anterior ecuación es escrita de la forma compacta:

$$\mathcal{X} := \bar{T}x_0 + \bar{S}\mathbf{u} \quad (1.35)$$

Con este modelo (1.33) puede ser escrita de la siguiente forma:

$$J_N(\{x_k\}, \{u_k\}) = \mathcal{X}^T \mathbf{Q} \mathcal{X} + \mathbf{u}^T \mathbf{R} \mathbf{u} \quad (1.36)$$

donde $\mathbf{Q} = \text{blockdiag}\{Q, \dots, Q, P\}$, $\mathbf{Q} \succeq 0$ y $\mathbf{R} = \text{blockdiag}R, \dots, R$, $\mathbf{R} \succ 0$. Sustituyendo (1.35) en (1.36) y operando se tiene:

$$J_N(\{x_k\}, \{u_k\}) = \frac{1}{2} \mathbf{u}^T \mathcal{H} \mathbf{u} + x_0^T \mathcal{F} \mathbf{u} + x_0^T \mathcal{Y} x_0 \quad (1.37)$$

donde $\mathcal{H} = \mathbf{R} + \bar{S}^T \mathbf{Q} \bar{S}$, $\mathcal{F} = \bar{T}^T \mathbf{Q} \bar{S}$ y $\mathcal{Y} = \bar{T}^T \mathbf{Q} \bar{T}$. En cuanto al manejo de las restricciones, dado que las mismas son poliédricas es posible formularlas como desigualdades de la siguiente forma (Borrelli *et al.* 2014, Capítulo 12):

$$G_0 \mathbf{u} - E_0 x_0 \leq s_0$$

En el citado trabajo se muestran las estructuras de las matrices G_0 , E_0 y del vector s_0 . Finalmente, el problema de optimización cuadrática que se pretende resolver es:

$$J_N^{opt}(x) := \min_{\mathbf{u}} \frac{1}{2} \mathbf{u}^T \mathcal{H} \mathbf{u} + x_0^T \mathcal{F} \mathbf{u} + \frac{1}{2} x_0^T \mathcal{Y} x_0 \quad (1.38a)$$

sujeto a:

$$G_0 \mathbf{u} - E_0 x_0 \leq w_0 \quad (1.38b)$$

1.5.5. Control Predictivo Robusto para Sistemas LTI Inciertos

En mucho caso es complicado tener un conocimiento perfecto del modelo de la planta que se desea controlar. Esta sección está enfocada en los casos cuando el modelo del sistema es incierto, dicha incertidumbre puede ser modelada como *perturbaciones aditivas* y/o incertidumbres *paramétrica* en las matrices del espacio de estado del proceso. Para este tipo de sistemas existen dos enfoques distintos en cuanto al diseño de controladores robustos. El primero se conoce como control robusto con “predicciones en lazo abierto” (Kouvaritakis and Cannon 2015, Capítulo 3) (y los trabajos citados en este), mientras que el segundo es con “predicciones en lazo cerrado”⁵ (Kothare *et al.* 1996, Lee and Yu 1997, Scokaert and Mayne 1998, Bemporad and Morari 1999b). Para un detalle del diseño de controladores robustos formulados con optimización sobre predicciones en lazo abierto, se refiere al lector al libro (Kouvaritakis and Cannon 2015, Capítulo 3).

Control robusto basado en LMIs:

En el trabajo (Kothare *et al.* 1996) se propone el diseño de controladores predictivo robusto para sistemas con incertidumbres y sujetos a restricciones. En el citado trabajo se considera un sistema incierto y se modelan mediante una aproximación politópica o multi-modelo lineal variante en el tiempo, *i.e.*:

$$x_{k+1} = A_k x_k + B_k u_k \quad (1.39a)$$

$$y_k = C x_k, \quad (1.39b)$$

$$[A_k \ B_k] \in \Omega, \quad (1.39c)$$

donde $y_k \in \mathbb{R}^{n_y}$ es la salida del sistema. El conjunto Ω es un politopo de matrices definido por:

$$\Omega := \text{conv}\{[A_1 \ B_1], [A_2 \ B_2], \dots, [A_L \ B_L]\} \quad (1.40)$$

siendo *conv* la envolvente convexa. Cada $[A \ B]$ dentro del conjunto convexo

⁵La diferencia entre ambas formulaciones es la siguiente: El problema de control óptimo puede ser visto como un “juego” entre dos jugadores; el controlador u_k y las perturbaciones w_k , dicho controlador trata de contrarrestar cualquier realización factible de w_k solo con una secuencia simple $\{u_0, \dots, u_{N-1}\}$. El modelo de predicción en *lazo abierto* no considera que en el siguiente instante, el controlador puede medir el estado x_1 y ajustar su entrada u_1 basándose en esta medida. Por tanto al no considerar este hecho, el efecto de w_k puede crecer y conducir fácilmente a la inviabilidad del problema de optimización subyacente. Por el contrario, el modelo de predicción en *lazo cerrado* toma en cuenta que el controlador conocerá el estado, por lo tanto, podrá actualizar su valor en cada instante de tiempo.

Ω es una combinación lineal de los vértices, *i.e.*:

$$[A \ B] = \sum_{i=1}^L \alpha_i [A_i \ B_i], \quad \sum_{i=1}^L \alpha_i = 1, \quad 0 \leq \alpha_i \leq 1 \quad (1.41)$$

Nótese que para $L = 1$ se corresponde con un sistema LTI de la forma (1.1b).

Dado un sistema descrito por (1.39a) con sus incertidumbres asociadas (1.40), el problema de minimización de una función de coste robusta se define como:

$$\min_{u_{k+i|k}, i \geq 0} \quad \max_{[A_{k+j} \ B_{k+j}] \in \Omega, j \geq 0} J_{\infty}(k), \quad (1.42)$$

siendo:

$$J_{\infty}(k) := \sum_{i=0}^{\infty} \left(x_{k+i|k}^T Q x_{k+i|k} + u_{k+i|k}^T R u_{k+i|k} \right) \quad (1.43)$$

donde $x_{k+i|k}$ representa el estado en el instante $k+i$ predicho en el tiempo k , mientras que $u_{k+i|k}$ es la acción de control en el instante $k+i$ calculada en el tiempo k , con $Q > 0$ y $R > 0$.

La ecuación (1.42) es conocida como un problema de optimización “*mín-máx*”. La maximización es sobre el conjunto Ω y corresponde a la elección del modelo $[A_{k+i} \ B_{k+i}] \in \Omega, i \neq 0$, que si es utilizado como el modelo de predicción en (1.42), conduciría al valor más alto “*peor caso*” de $J_{\infty}(k)$ entre todas las plantas en Ω .

Dado que este problema mí-máx no es computacionalmente tratable, se propone calcular una cota superior del índice de coste robusto (1.42), asumiendo que en cada periodo de tiempo k , se usa una ley de control constante por realimentación $u_{k+i|k} := K x_{k+i|k}, i \neq 0$, para minimizar el coste $J_{\infty}(k)$.

Cálculo de la cota superior del coste. Considérese una función de Lyapunov cuadrática de la forma:

$$\mathcal{V}(x) := x^T P x, \quad (1.44)$$

$P > 0$. Para cualquier $[A_{k+i} \ B_{k+i}] \in \Omega, i \neq 0$, supóngase que $\mathcal{V}(x)$ satisface la siguiente desigualdad:

$$\mathcal{V}(x_{k+i+1|k}) - \mathcal{V}(x_{k+i|k}) \leq - \left[x_{k+i|k}^T Q x_{k+i|k} + u_{k+i|k}^T R u_{k+i|k} \right] \quad (1.45)$$

Para que la función objetivo robusta sea finita, debe cumplirse que $x(\infty|k) = 0$ y por lo tanto, $\mathcal{V}(x(\infty|k)) = 0$. Entonces, se construye $\mathcal{V}(x)$ de manera que

acote por la parte superior a $J_\infty(x)$, i.e.,

$$\max_{[A_{k+j} \ B_{k+j}] \in \Omega, j \geq 0} J_\infty(k) \leq \mathcal{V}(x_{k|k}), \quad (1.46)$$

entendiéndose $x_{k|k}$ como el estado medido en el instante k . El algoritmo para el diseño de un MPC robusto ha sido redefinido con el fin de calcular, en cada paso k , una ley de control por realimentación $u_{k+i|k} := Kx_{k+i|k}$ donde K se obtiene minimizando $\mathcal{V}(x_{k|k})$, resolviendo el siguiente problema:

Teorema 1.1. (Kothare et al. 1996) *Sea $x_k = x_{k|k}$ el estado medido en el instante k del sistema (1.39a), considérese que no existen restricciones en las entradas y salidas del proceso.*

Dado un conjunto de incertidumbres Ω , la matriz de realimentación del estado K de la ley de control $u_{k+i|k} := Kx_{k+i|k}$, $i \geq 0$ que minimiza la cota superior $\mathcal{V}(x_{k|k})$ viene dada por:

$$K = Y\bar{Q}^{-1} \quad (1.47)$$

donde $\bar{Q} = \gamma P(k)^{-1} > 0$, Y son obtenidas de la solución ("si existe") del siguiente problema de minimización (en términos de LMIs, Apéndice A):

$$\min_{\gamma, \bar{Q}, Y} \gamma \quad (1.48)$$

sujeto a:

$$\begin{bmatrix} 1 & x_{k|k}^T \\ x_{k|k} & \bar{Q} \end{bmatrix} \geq 0 \quad (1.49)$$

$$\begin{bmatrix} \bar{Q} & \bar{Q}A_j^T + Y^T B_j^T & \bar{Q}Q^{1/2} & Y^T R^{1/2} \\ A_j \bar{Q} + B_j Y & \bar{Q} & 0 & 0 \\ Q^{1/2} \bar{Q} & 0 & \gamma I & 0 \\ R^{1/2} Y & 0 & 0 & \gamma I \end{bmatrix} \geq 0, \quad j = 1, 2, \dots, L \quad (1.50)$$

Restricciones: Considérese que el sistema (1.39a) está sujeto a ciertas restricciones, tanto en las entradas como en las salida de la forma:

$$|u_{k+i|k}^j| \leq u_{max}^j, \quad i \geq 0, \quad j = 1, 2, \dots, n_u, \quad (1.51)$$

$$|y_{k+i|k}^j| \leq y_{max}, \quad i \geq 1, \quad j = 1, 2, \dots \quad (1.52)$$

Las restricción (1.51) es satisfecha si existe una matriz simétrica tal que (Kothare *et al.* 1996):

$$\begin{bmatrix} X & Y \\ Y^T & \bar{Q} \end{bmatrix} \geq 0, X_{jj} \leq u_{j,max}^2, j = 1, 2, \dots, n_u \quad (1.53)$$

De forma similar, las restricciones (1.52) son satisfechas si existe una matriz simétrica Z tal que, para cada vértice de Ω :

$$\begin{bmatrix} Z & C(A_j\bar{Q} + B_jY) \\ (A_j\bar{Q} + B_jY)^T C^T & \bar{Q} \end{bmatrix} \geq 0, j = 1, 2, \dots, L \quad (1.54)$$

con $Z_{rr} \leq y_{r,max}^2, r = 1, 2, \dots, n_y$.

Definición 1.27. (Wan and Kothare 2003) Dado un sistema dinámico (1.39a), un conjunto $\mathcal{E} = \{x \in \mathbb{R}^{n_x} \mid x^T \bar{Q}^{-1} x \leq 1\}$ del espacio de estado \mathbb{R}^{n_x} es un **conjunto elipsoidal invariante y asintóticamente estable**, si posee la propiedad que, cuando $x_{k_1} \in \mathcal{E}$, entonces, $x_k \in \mathcal{E}$ para todo instante de tiempo $k \geq k_1$ y $x_k \rightarrow 0$ como $k \rightarrow \infty$.

Teorema 1.2. (Kothare *et al.* 1996) Para el sistema (1.39a), en el instante de tiempo k , la matriz de realimentación del estado K en la ley de control $u_{k+i|k} = Kx_{k+i|k}, i \geq 0$, que minimiza la cota superior γ sobre el peor caso de la función de coste $J_\infty(k)$, está dada por $K = Y\bar{Q}^{-1}$, donde $\bar{Q} > 0$ e Y son obtenidas desde la solución (si existe) del siguiente problema de optimización:

$$\min_{\gamma, \bar{Q}, X, Y, Z} \gamma \text{ sujeto a (1.49), (1.50), (1.53), (1.54)} \quad (1.55)$$

La ley de control obtenida de la solución factible este teorema hace que el sistema en lazo cerrado sea robustamente asintóticamente estable.

Algoritmo de implementación (Wan and Kothare 2003): Considérese un sistema incierto dado por (1.39a) sujeto a las restricciones tanto en las entradas como en las salidas del proceso.

(a) **Fuera de línea:** Dado un estado inicial, genere una secuencia de minimizadores $\gamma_i, \bar{Q}_i, X_i, Y_i, Z_i, (i = 1, 2, \dots, N)$ de la siguiente forma, $i := 1$:

1. Calcular $\gamma_i, \bar{Q}_i, X_i, Y_i, Z_i$ para el estado x_i usando el Teorema 1.2 añadiendo una restricción adicional $\bar{Q}_{i-1} > \bar{Q}_i$ (ignorar en $i = 1$). Almacenar el resultado ($\bar{Q}_i, K^i = Y_i\bar{Q}_i^{-1}, X_i, Y_i$) en una tabla de búsqueda denotada con Γ ;

2. if $i < N$ seleccionar un nuevo estado x_{i+1} que satisfaga $x_{i+1}^T \bar{Q}_i^{-1} x_{i+1} < 1$.
 1. $i = i + 1$, ir al paso 1.
- (b) **En línea:** Dado un estado inicial x_0 que cumple $x_0^T \bar{Q}_1 x_0 \leq 1$. Sea x_k el estado en el instante k . Realizar una búsqueda sobre \bar{Q}_i^{-1} en la tabla Γ para encontrar el índice más grande i (o lo que es equivalente, encontrar la elipse más pequeña $\mathcal{E}_i := \{x \in \mathbb{R}^{n_x} \mid x^T \bar{Q}_i^{-1} x \leq 1\}$) tal que $x_k^T \bar{Q}_i^{-1} x_k \leq 1$. Aplicar la ley de control por realimentación $u_k := K^i x_k$.

Observación 1.5. *Aplicando el algoritmo anterior, se calcula –si existe solución– una secuencia de ganancias de realimentación K^i que corresponde a un conjunto elipsoidal invariante \mathcal{E}_i . Aunque cada ganancia K^i garantiza una estabilidad robusta para todo los estados pertenecientes a \mathcal{E}_i , dichas ganancias no son necesariamente óptimas. Los conjuntos invariantes elipsoidales son solo una aproximación del verdadero conjunto invariante poliédrico. Por consiguiente, ejecutando solo el paso 1 del algoritmo, se obtiene un resultado conservador, debido a que la región estabilizante de los conjuntos invariantes elipsoidales es significativamente más pequeña que los conjuntos poliédricos.*

A partir de lo anterior, se deduce le necesidad de calcular los conjuntos invariantes poliédricos utilizando las ganancias K^i obtenidas en el paso 1 del algoritmo. En el artículo (Bumroongsri and Kheawhom 2012, Algoritmo 3.1), se presenta un algoritmo para calcular una secuencia de conjuntos invariantes poliédricos, reduciendo así el conservadurismo, dado que las regiones poliédricas estabilizantes son sustancialmente más grandes. En el Apéndice B.2 se muestra un ejemplo de implementación del algoritmo fuera de línea a y aquel presente en (Bumroongsri and Kheawhom 2012).

Mín-Max MPC con Predicciones en Lazo Cerrado

Como se ha mencionado anteriormente el diseño de controladores robustos con predicciones en lazo abierto presentan el inconveniente de que no se toman en cuenta que, dicho controlador conocerá el estado del sistema a la hora de implementar la acción de control. En el trabajo (Scokaert and Mayne 1998) se presenta una metodología de controladores predictivos basado en optimización mín-máx (MMMPC), en donde se considera un modelo de predicción en lazo cerrado. Esta estrategia se caracteriza por considerar un conjunto de secuencias de acciones de control, en vez de una única secuencia de actuaciones. La estabilidad de esta propuesta es garantizada mediante un esquema de control *dual*, según el cual, el MMMPC se encarga de conducir el estado del sistema a un conjunto invariante de control, una vez que el estado

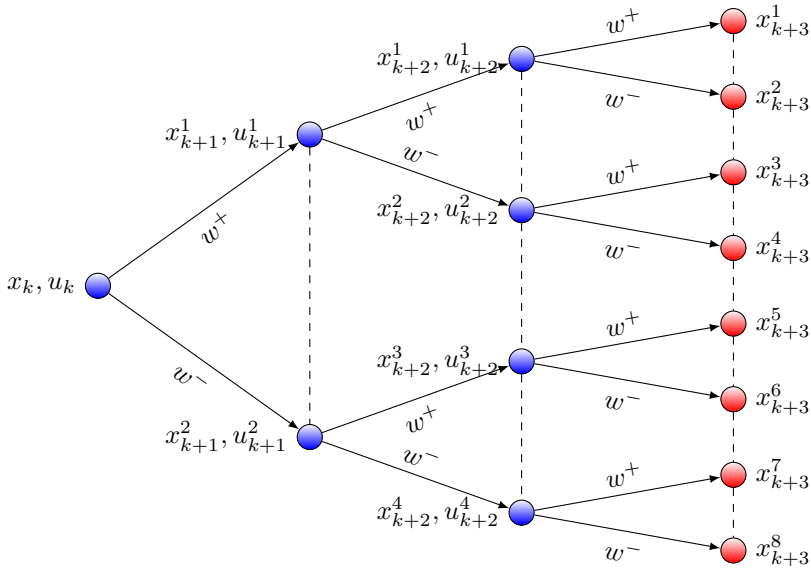


Figura 1-3: Ejemplo: trayectorias de estados para diferentes realizaciones de la perturbación. Los nodos finales (rojos) corresponden al estado terminal.

del proceso esté dentro del conjunto invariante, se usa una ley de control por realimentación para llevarlo al origen o a una vecindad de este.

En el citado trabajo se considera el siguiente sistema LTI perturbado:

$$x_{k+1} = Ax_k + Bu_k + w_k \quad (1.56)$$

donde $w_k \in \mathbb{W}$ es una perturbación aditiva acotada.

En el instante k , se denota con $\{w_{j|k}^l\}$ a las posibles realizaciones de la perturbación, donde $l \in \mathcal{L}$ son los índices que permiten identificar a dichas realizaciones. Por otro lado, $\{u_{j|k}^l\}$ representa la secuencia de acciones de control asociada a la l -ésima realización de la perturbación. Por tanto, la ecuación (1.56) puede ser escrita de la forma:

$$x_{j+1|k}^l = Ax_{j|k}^l + Bu_{j|k}^l + w_{j|k}^l, \quad l \in \mathcal{L} \quad (1.57)$$

donde $x_{j|k}^l = x_k$ para todo $l \in \mathcal{L}$. El problema de optimización puede formu-

larse como:

$$\min_{\{u_{j|k}^l\}} \max_{l \in \mathcal{L}} \sum_{j=0}^{N-1} L(x_{k+j|k}^l, u_{k+j|k}^l) \quad (1.58a)$$

sujeto a:

$$x_{j|k}^l \in \mathbb{X}, \quad j \geq t, \quad \forall l \in \mathcal{L}_v \quad (1.58b)$$

$$u_{j|k}^l \in \mathbb{U}, \quad j \geq t, \quad \forall l \in \mathcal{L}_v \quad (1.58c)$$

$$x_{k+N|k}^l \in \mathbb{X}_f, \quad l \in \mathcal{L}_v \quad (1.58d)$$

$$x_{j|k}^{l_1} = x_{j|k}^{l_2} \Rightarrow u_{j|k}^{l_1} = u_{j|k}^{l_2}, \quad \forall l_1, l_2 \in \mathcal{L} \quad (1.58e)$$

donde la función de coste L es semidefinida positiva. Nótese que la restricción (1.58d) es la que da garantía de estabilidad en esta estrategia de control. Por su parte, \mathcal{L}_v es el conjunto de l vértices de \mathbb{W} , de manera que $\{w_{j|k}^l\}$ toma valores solo en los vértices de \mathbb{W} .

Es importante resaltar la importancia de la ecuación (1.58e), la cual es conocida como *restricción de causalidad*⁶. Para ello, supóngase el siguiente ejemplo: Considere un horizonte $N = 3$ y $\mathbb{W} = \{w \in \mathbb{R}^2 \mid w^- \leq w_k \leq w^+\}$, por tanto se tienen un total $2^3 = 8$ posibles realizaciones de la perturbación, las cuales están asociadas a 8 secuencias de acciones de control $\{u_{k|k}^l, u_{k+1|k}^l, u_{k+2|k}^l, l = 1, \dots, 8\}$. Esto daría lugar a un total de 24 variables de decisión. Sin embargo con la restricción de causalidad (1.58e) este número se reduce a 7 variables, correspondiente a (véase Figura 1-3):

$$\{u_k, u_{k+1}^1, u_{k+1}^2, u_{k+2}^1, u_{k+2}^2, u_{k+2}^3, u_{k+2}^4\}$$

En cuanto a la estabilidad del sistema en lazo cerrado con ley de control resultante del problema (1.58), (Scolaert and Mayne 1998) prueban que dicha acción de control conduce el estado del sistema a un conjunto robusto invariante \mathbb{X}_f asintóticamente.

La realimentación es introducida en esta formulación por el hecho de que considera una secuencias de acciones de control para cada realización de la perturbación. Para un detalla en profundidad de controladores predictivos robustos basados en optimización mín-máx, se recomienda al lector consultar la tesis doctoral (Löfberg 2003).

⁶Esta restricción impone que para un mismo estado x , la acción de control debe ser idéntica en cada una de la realizaciones considerada de la perturbación

1.6. Diferentes Formulaciones de los MPC

A continuación se discute brevemente diferentes formulaciones sobre controladores predictivos basados en modelos, haciendo mención a MPC no lineal, MPC con soluciones explícitas, MPC económicos, MPC híbridos.

Control Predictivo no Lineal: Es bien sabido que en general, los procesos dinámicos reales presentan comportamientos no lineales. Sin embargo, en gran parte de los desarrollos de los MPC se utiliza un modelo lineal del sistema. Algunas de las razones de esto se debe a: por un lado a la simplicidad del problema de optimización subyacente (como se detalló en la Sección (1.5.4) el índice de coste puede ser formulado como programación cuadrática QP e incluso puede obtenerse una solución explícita), por otro lado, con los modelos lineales se consiguen excelentes aproximaciones para muchos de los procesos no lineales en zonas de operación acotadas.

No obstante, existen casos en donde los modelos lineales no pueden representar efectivamente el comportamiento del proceso no lineal, y por tanto, las predicciones futuras no son útiles para la formulación del problema MPC. Para estos casos, es necesaria la implementación de controladores predictivos *no lineales* (Grüne and Pannek 2011). El principal problema que surge en esta metodología, es que el problema de optimización puede ser, en un caso general, *no convexo* y se resuelve con programación no lineal (NLP) genérica. La resolución de este tipo de problemas es mucho más compleja que un problema QP. Otro problema es que la solución suele estar basada en la obtención de una solución sub-óptima. Por último, los aspectos fundamentales en el desarrollo de MPC se convierten en problemas muy complejos de resolver numéricamente, como lo son: el estudio de la estabilidad y factibilidad, la robustez, seguimiento de referencias, *etc.*

Sin embargo, existen gran cantidad de trabajo que se enfocan en este campo (Grüne and Pannek 2011) (y los trabajos referidos en éste). Una forma de resolver los problemas de optimización no convexos es mediante programación cuadrática secuencial (SQP) (Grüne and Pannek 2011, Capítulo 10). Esta técnica consiste en sustituir en cada iteración, la función de coste y restricciones no lineales del problema por una función cuadrática y restricciones lineales, respectivamente. Es un método que resuelve una secuencia de sub-problemas de optimización, en donde cada uno de ellos optimiza una función de coste cuadrática sujeto a una linealización de las restricciones.

La formulación del problema de optimización para MPC no lineales puede

ser descrita por (1.26). En el libro (Grüne and Pannek 2011) se exponen los principales resultados conseguidos en cuanto al para el diseño de MPC no lineales. En este se estudia las condiciones de estabilidad y factibilidad, así como también la robustez de las leyes de control basadas en controladores predictivos no lineales.

Para sistemas no lineales puede emplearse métodos de modelados que usan modelos borrosos Takagi-Sugeno (politópicos con funciones de interpolación conocidas, dependientes del estado) para caracterizar la dinámica del sistema (Tanaka and Wang 2004). En (Ding 2011) se diseña un control predictivo con realimentación de la salida para sistemas no lineales representados por modelos Takagi-Sugeno. Usando conjuntos invariantes y LMI, garantizan la factibilidad y estabilidad del controlador MPC. Una propuesta interesante fue desarrollado por (Zhang *et al.* 2007) en donde el diseño del controlador predictivo está formulado en dos etapas diferentes. En la primera de ella, resolviendo un problema LMI fuera de línea se obtiene una ley de control terminal basándose en las restricciones terminales y en un conjunto de desigualdades que garanticen la estabilidad del controlador a partir de una función de coste en tiempo infinito, de la forma (1.43). En la segunda etapa, se busca la resolución de un problema de optimización par así determinar una secuencia de acciones de control óptimas minimizando una función de coste de tiempo finito. Una metodología que propone QP iterativa de resolución de problemas de control predictivo no lineal con modelos Takagi-Sugeno aparece en, por ejemplo, (Arino *et al.* 2014b). En un ámbito genérico, no específicamente ligado al control predictivo, el conservadurismo de las soluciones basadas en modelos Takagi-Sugeno respecto de soluciones no lineales ideales se discute en (Sala 2009). Una aproximación al cálculo de conjuntos invariantes para sistemas TS con controladores que a su vez son TS (parallel-distributed-compensation, en la literatura en lengua inglesa), aparece en (Ariño *et al.* 2014a).

MPC Económico. En muchos de los problema de control óptimo, los parámetros presente en el coste de etapa $L(\cdot)$ son seleccionados con el fin de alcanzar objetivos de diseños tradicionales, tales como, respuesta rápida con una sobreoscilación limitada, seguimiento de referencia, *etc.* En cambio en los procesos industriales, a menudo se plantea como objetivo principal la *rentabilidad*. Los MPC económicos, en su fase actual, considera las limitaciones de rentabilidad seleccionando que la función de etapa $L(x, u)$ sea el costo económico neto de la operación de la planta en (x, u) . Como trabajos de gran interés se pueden mencionar, entre otros, (Rawlings and Amrit 2009, Diehl *et al.* 2011, Rawlings

et al. 2012). En (Diehl *et al.* 2011) muestran que el controlador resultante es estable y dirige el sistema a un estado de equilibrio admisible que es óptimo en el sentido económico, y que dicho sistema controlado además, exhibe un mejor rendimiento con respecto a la referencia que la formulación estándar del MPC. En el citado trabajo también se muestra que, en algunos casos, es posible utilizar funciones de Lyapunov para establecer las propiedades de estabilidad en los controladores predictivos económicos.

MPC Híbridos. El control predictivo basada en modelo ha sido desarrollado ampliamente para el control de sistemas en tiempo discreto o para sistemas en tiempo continuo. Sin embargo, muchos de los procesos industriales contienen componentes discretos, tales como válvulas, interruptores, selectores de velocidad *etc*, además de los componentes continuos que se describen por ecuaciones diferenciales. La consideración de sistemas *híbridos* que posean ambos tipos de componentes permite la exploración de nuevas técnicas de control para esta clase de sistema (Mayne 2014). Unos de los primeros trabajos en este ámbito es (Bemporad and Morari 1999a). En el trabajo (Camacho *et al.* 2010) se presenta una excelente revisión de la literatura existen para esta clase de sistema.

Los sistemas afines a tramos han recibido mayor atención en la comunidad académica debido probablemente a la continuidad de la función $f(\cdot)$ en el sistema $x_{k+1} = f(x, u)$ lo cual facilita el análisis de estabilidad. Para funciones $f(\cdot)$ discontinuas se remite al lector a la tesis doctoral (Lazar 2006). En (Lazar 2006) se extiende las condiciones suficientes de estabilidad proporcionadas por una restricción terminal y función de coste para sistemas discontinuos. En el citado trabajo, se proponen métodos para calcular conjuntos de restricciones terminales adecuados y funciones de costes para sistemas afines a tramos.

Por otro lado, los sistemas conmutados es uno de los principales campos de interés en sistemas híbridos. El sistema que se pretende controlar puede tener diferentes dinámicas en diferentes regiones del espacio de estados. Un claro ejemplo son los sistemas afines a tramos donde el espacio de estado está dividido en varias regiones, en cada una de las cuales, el sistema es afín. En (Mayne and Raković 2003) se presenta un algoritmo para la solución del problema de control predictivo, suponiendo que la función de coste $J(\cdot)$ no es diferenciable. Por su parte en (Müller and Allgöwer 2012) tratan con sistemas no lineales en tiempo continuo. Aquí el espacio de estado no se encuentra dividido en regiones diferentes, pero el controlador puede escoger que coste de etapa $L_i(\cdot)$ va a emplear en el problema de optimización, con el fin de obtener un candidato $K_i(x)$ para la ley de control actual.

MPC con Soluciones Explícitas. Es de conocimiento general que los MPC reemplazan el cálculo *fuera de línea* de una ley de control por la solución *en línea* de un problema de optimización, cuya solución conduce a una secuencia de acciones de control. Por tanto, la acción de control que se envía al sistema viene dada por (1.29). Como se vio en la Sección 1.5.4 el problema de optimización (1.26) puede ser transformado a un problema QP dado por (1.38), para sistemas LTIs. En (Bemporad *et al.* 2002b) se muestra como la secuencia de actuaciones $\mathbf{u}^{opt}(x)$ tiene una estructura afín a tramos de la forma:

$$\mathbf{u}^{opt}(x) = K^i x + g^i, \quad \text{para } x \in X_i = \{x \mid H_i x \leq L_i\}$$

definida en l conjuntos $X_i, i = 1, \dots, l$ que forman una partición poliédricas del espacio de trabajo. La idea es que el problema de optimización puede ser resuelto paramétricamente, obteniendo así una ley de control *explícita*. Estas leyes de control son funciones afines a tramos del estado del sistema. Un MPC explícito permite resolver el problema de optimización fuera de línea para un determinado rango de condiciones de operación de interés. Mediante las técnicas de programación multiparamétrica, un MPC explícito calcula la acción de control óptima *fuera de línea* como una función “explícita” del estado del sistema. De este modo, la implementación *en línea* se reduce a una evaluación de una función. En la mayoría de los casos, tal función es afín a tramos (Alessio and Bemporad 2009). Algoritmos para el cálculo de soluciones explícitas del problema de optimización bajo restricciones fueron desarrollados, entre otros en: (Bemporad *et al.* 2002b, Bemporad *et al.* 2002a, Raković *et al.* 2004) para sistemas lineales, en (Mayne and Raković 2003, Raković *et al.* 2004, Kvasnica *et al.* 2011) para sistemas afines a tramos en tiempo discreto y para sistemas inciertos en (Bemporad *et al.* 2003, Muñoz *et al.* 2004); el caso de restricciones no convexas se aborda en (Pérez *et al.* 2011).

Capítulo 2

Sistemas Lineales con Saltos Markovianos

2.1. Introducción

Uno de los beneficios en el control de sistemas dinámicos es su capacidad de mantener un comportamiento “aceptable” mientras que se cumple algún requerimiento de desempeño, incluso en presencia de cambios abruptos en la dinámica de la planta. Estos cambios pueden ser originados por perturbaciones ambientales abruptas, fallos o reparaciones en los componentes del sistema, cambios en subsistemas interconectados, cambios en el punto de operación, *etc.* Ejemplos de estos casos se pueden encontrar en sistemas económicos (Blair Jr and Sworder 1975), sistemas de control en aeronaves (Aberkane *et al.* 2008), sistemas de control en red (Zhang *et al.* 2005, Wu and Chen 2007), de manufactura (Boukas and Liu 2001), *etc.* En muchos casos estos sistemas pueden ser modelados a través de un conjunto de sistemas lineales en tiempo discreto cuya matriz de transición entre diferentes modos viene dada por una cadena de Markov (Costa *et al.* 2005). Esta familia de sistemas es conocida en la literatura como: *sistemas lineales con saltos markovianos*, desde ahora MJLS⁷ y, es el principal tema abordado en este capítulo.

Por ejemplo, considérese un sistema dinámico que –en cierto momento– está descrito por el modelo \mathcal{H}_1 . Se supone que este modelo esta sujeto a cambios abruptos que hacen que el sistema tenga un comportamiento distinto, el cual será descrito por el modelo \mathcal{H}_2 . Más en general, considérese que el siste-

⁷Viene del acrónimo en lengua inglesa Markov Jump Linear Systems.

ma esta sujeto a una series de posibles cambios y que el mismo conmuta entre un conjunto de modelos, *i.e.*, $\{\mathcal{H}_1, \dots, \mathcal{H}_N\}$. Es posible asociar cada uno de estos modelos con un *modo de operación o funcionamiento* del sistema, o simplemente *modo*, por tanto, para un cierto instante de tiempo k , se asume que el sistema se encuentra en *modo* i y la representación de su modelo viene dada por \mathcal{H}_i , $i = 1, 2, \dots$.

Se supone a lo largo de este trabajo que el sistema *salta* entre diferentes modos de operación y, que dichos saltos evolucionan estocásticamente de acuerdo a una cadena de Markov. A lo largo de este manuscrito, el estado de la cadena de Markov (o modo de operación) es denotado como θ_k .

2.2. Cadena de Markov

Una cadena de Markov de tiempo discreto es un proceso estocástico que es la generalización simple de una secuencia de variables aleatorias independientes., en donde la dependencia de los eventos sucesivos se remonta sólo una unidad en el tiempo. En otras palabras, la probabilidad de ir al estado siguiente solo depende del estado actual y no de las transiciones históricas que condujeron a dicho estado. Esto se conoce como propiedad de Markov (Tijms 2003).

Definición 2.1. (Tijms 2003) *El proceso estocástico $\{X_n, n \geq 0\}$ con un espacio de estado \mathbf{I} es dicho ser una **cadena de Markov** en tiempo discreto si, para cada $n = 0, 1, \dots$*

$$Pr\{X_{n+1} = i_{n+1} \mid X_0 = i_0, \dots, X_n = i_n\} = Pr\{X_{n+1} = i_{n+1} \mid X_n = i_n\} \quad (2.1)$$

para todo valor posible de $i_0, \dots, i_{n+1} \in \mathbf{I}$.

Probabilidades de Transición. A lo largo del presente trabajo se consideran cadenas de Markov *homogéneas* con una probabilidad de transición entre el estado futuro j y un estado actual i dada por:

$$\pi_{ji} := Pr\{X_{n+1} = j \mid X_n = i\}, \quad i, j \in \mathbf{I}$$

Se conoce como matriz de transición a la matriz $\mathbb{P} \in \mathbb{R}^{n \times n}$ dada por:

$$\mathbb{P} = \begin{bmatrix} \pi_{11} & \pi_{12} & \cdots & \pi_{1n} \\ \pi_{21} & \pi_{22} & \cdots & \pi_{2n} \\ \vdots & & \ddots & \\ \pi_{n1} & \pi_{n2} & \cdots & \pi_{nn} \end{bmatrix} \quad (2.2)$$

Para cada π_{ji} se satisface la condición $\pi_{ji} \geq 0$, $\sum_{j=1}^n \pi_{ji} = 1$, $i, j \in \mathbf{I}$.

Se define como probabilidades de transición a n pasos a (Tijms 2003):

$$\pi_{ji}^{(n)} := Pr\{X_n = j \mid X_0 = i\}, \quad i, j \in \mathbf{I}$$

para cualquier $n = 1, 2, \dots$. Nótese que $\pi_{ji} = \pi_{ji}^{(1)}$.

La distribución estacionaria de una cadena de Markov con una matriz \mathbb{P} es un vector ψ , cuyo elementos no son negativo y suman 1, tal que, $\psi\mathbb{P} = \psi$.

Clasificación de los Estados. Se dirá que un estado es *absorbente* cuando una vez se entra en él, no es posible salir del mismo, es decir, $\pi_{ii} = 1$, $\pi_{ji} = 0$, $i \neq j$. Por otro lado, un estado i es *transitorio* solo si $\sum_{n=1}^{\infty} \pi_{ii}^{(n)} < \infty$ y un estado j es *recurrente* solo si $\sum_{n=1}^{\infty} \pi_{ii}^{(n)} = \infty$, (Tijms 2003). Esto significa que, la probabilidad de que el sistema regrese al estado i , habiendo empezado en i , es 1.

Definición 2.2. (Tijms 2003) Un estado j es *accesible* desde el estado i (denotado como $i \rightarrow j$) si $\pi_{ji}^{(n)} > 0$ para algún $n \geq 0$. Dos estados i y j están *comunicado* si j es accesible desde i e i es accesible desde j ($i \leftrightarrow j$).

En la Figura 2-1 se muestra un ejemplo de una cadena de Markov de 3 estados. Nótese que el estado 3 es accesible desde el estado 1 (debido a: $1 \rightarrow 2$ y $2 \rightarrow 3$). También se observa que los estados 1 y 2 están comunicados, *i.e.*, $1 \leftrightarrow 2$, al igual que los estados 2 y 3 ($2 \leftrightarrow 3$), por lo tanto, el estado 1 y 3 también están comunicados. Esto se conoce como cadenas de Markov *irreducibles*, es decir, todo sus estados son accesibles entre sí y por lo tanto están comunicados, de lo contrario, la cadena se denota como *no irreducible*.

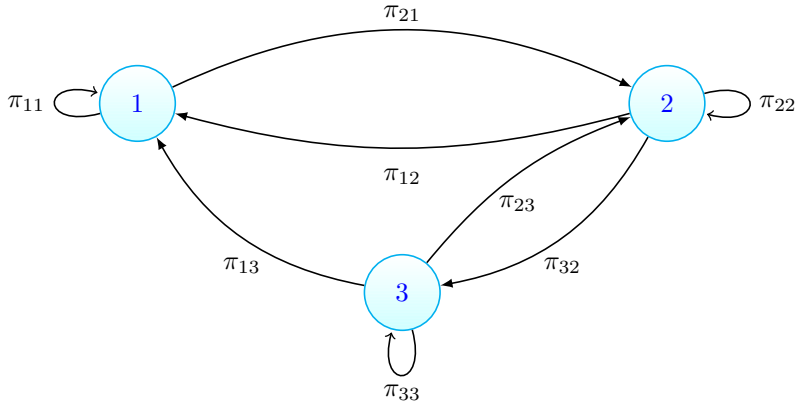


Figura 2-1: Diagrama de estado de una cadena de Markov. Cada círculo representa un posibles estados de la cadena.

2.3. Control Óptimo para MJLS sin Restricciones

Considérese un sistema lineal discreto con saltos markovianos (MJLS) dado por la siguiente ecuación:

$$x_{k+1} = A_{\theta_k} x_k + B_{\theta_k} u_k, \theta_k \in \mathcal{M} \quad (2.3)$$

donde θ_k representa el *modo de operación* del proceso. A_{θ_k} indica que la matriz de estado A depende del parámetro θ_k . Por su parte, $\mathcal{M} = \{1, 2, \dots, M\}$ son los conjuntos de modos posibles, los cuales se encuentran gobernados por una cadena de Markov discreta con una matriz de probabilidades de transiciones dada por $\mathbb{P} := [\pi_{ji}]$. Dicha matriz tiene la forma (2.2).

Se asume que el proceso (2.3) conmuta entre M modos diferentes tal que:

$$A_{\theta_k} \in \bar{\mathcal{A}} := \{A_1, \dots, A_M\}, \quad B_{\theta_k} \in \bar{\mathcal{B}} := \{B_1, \dots, B_M\} \quad (2.4)$$

Además, la matriz \mathbb{P} se asume conocida y que se mantiene constante para todo tiempo futuro.

Conocimiento de la variable de modo del sistema. Los controladores para MJLS pueden basarse en distintas suposiciones sobre el conocimiento de la variable de modo del sistema. En este capítulo, siguiendo (Costa *et al.* 2005), se asumirá que en el instante k existe un supervisor/diagnosticador que propor-

ciona⁸ la variable θ_k . La forma de implementar las opciones del diagnosticador anterior están fuera del alcance de este trabajo.

Predicciones. Se denotará con $\psi_{k+j|k}$ los valores predichos de una variable ψ en un tiempo futuro $k + j$, con la información disponible en el instante k . Obviamente $\psi_{k|k} \equiv \psi_k$, en particular $x_{k|k}$ deberá entenderse como el estado medido x_k .

Promedio condicionado. $\mathbb{E}_k(\psi_{k+j|k})$ representa la esperanza matemática (valor medio esperado) de una variable aleatoria ($\psi_{k+j|k}$) condicionada a que la información del modo disponible en el instante k .

Control LQR para MJLS sin restricciones. En primer lugar, se define el índice de coste dependiente del modo en tiempo finito como

$$J_{0,N}^i = \mathbb{E}_0 \left(\sum_{t=0}^{N-1} L(x_{t|0}, u_{t|0}, \theta_t) + P(x_{N|0}, \theta_N) \right), \quad (2.5)$$

donde $P(x_{N|0}, \theta_N)$ representa la ponderación terminal del estado y $L(x_{t|0}, u_{t|0}, \theta_k)$ el peso de coste de etapa. El superíndice i indica que la información del modo disponible en el instante k , debe ser " i ", esto es $\theta_0 = i$.

Los desarrollos relacionados con el control óptimo cuadrático para MJLS sin restricciones aparecen en (Costa *et al.* 2005) para el caso en el que el modo actual de la planta se conoce, donde se determina u_k como una función de θ_k . La ecuación de optimalidad de Bellman es:

$$J_{k,N}^{opt,i}(x_k) = \min_{u_k \in \mathbb{U}} \mathbb{E}_k \left(L(x_k, u_k, \theta_k) + J_{k+1,N}^{opt,\theta_{k+1}}(x_{k+1|k}) \right), \quad (2.6)$$

siendo i la información del modo disponible al momento de calcular la acción de control.

Debido a que el modo actual se conoce ($\theta_k = i$), el primer termino en la ecuación de Bellman (2.6) es determinista, *i.e.*:

$$\mathbb{E}_k(L(x_k, u_k, \theta_k)) = L(x_k, u_k, i)$$

Ahora, el estado sucesor se calcula con las matrices del sistema desde el

⁸En el Capítulo 3, se considerará también la opción de conocer sólo θ_{k-1} , y en el Capítulo 5 se discutirá el conocimiento de cierta variable de información que permita conocer sólo una distribución de probabilidad de θ_k .

modo i , por lo tanto, el segundo termino viene dado por:

$$\mathbb{E}_k(J_{k+1,N}^{opt,\theta(k+1)}(x_{k+1|k})) = \sum_{j \in \mathcal{M}} \pi_{ji} J_{k+1,N}^{opt,j}(A_i x_k + B_i u_k) \quad (2.7)$$

Una solución particular para un horizonte a tiempo infinito se presenta a continuación: Definiendo el coste de etapa $L(x_k, u_k, \theta_k)$ y terminal $P(x_N, \theta_N)$ cuadrático tal que:

$$L(x_k, u_k, \theta_k) := x_k^T Q^{\theta_k} x_k + u^T R^{\theta_k} u, \quad S(x_N, \theta_N) := x_N^T S^{\theta_N} x_N \quad (2.8)$$

con $Q^{\theta_k} > 0$, $R^{\theta_k} > 0$, y S^{θ_N} siendo matrices arbitrarias de dimensiones apropiadas, esto da lugar a una ecuación tipo Riccati acoplada que depende del modo actual del sistema. De hecho, el controlador óptimo se obtiene iniciando la recursión a $S_N^i = S^i$, para todo $i \in \mathcal{M}$ y retrocediendo en el tiempo $k = N - 1, \dots, 0$, usando la notación $\bar{S}_k^i = \sum_{j=1}^M \pi_{ji} S_k^j$:

$$J_{k,N}^{opt,i} = x_k^T S_k^i x_k \quad (2.9)$$

$$S_k^i = Q^i + A_i^T \bar{S}_{k+1}^i A_i - A_i^T \bar{S}_{k+1}^i B_i (B_i^T \bar{S}_{k+1}^i B_i + R^i)^{-1} B_i^T \bar{S}_{k+1}^i A_i \quad (2.10)$$

$$u_k^{opt,i} = - (B_i^T \bar{S}_{k+1}^i B_i + R^i)^{-1} B_i^T \bar{S}_{k+1}^i A_i x_k$$

$$u_k^{opt,i} = - K_k^i x_k \quad (2.11)$$

donde $u_k^{i,opt}$ es la acción de control que debe aplicarse en el instante k , asumiendo que el modo de operación fuese $\theta_k = i$.

Horizonte infinito. Iterando las ecuaciones anteriores hasta la convergencia, se obtiene el caso límite, *i.e.*, S_∞^i (desde ahora S^i), por lo tanto, la acción de control óptima y el coste esperado vienen dado por:

$$u_k^{opt,i} = -K^i x_k, \quad J^{opt,i}(x_k) = x_k^T S^i x_k \quad (2.12)$$

siendo $J^{opt,i}(x_k) = \mathbb{E}_0(\sum_{k=0}^{\infty} L(x_k, u_k, \theta_k))$. Para más detalles consulte (Costa *et al.* 2005).

2.4. Estabilidad de MJLS

Existen una gran cantidad de trabajos en donde se estudian el problema de estabilidad en media cuadrática para MJLS, entre los que se pueden mencionar por ejemplo, (Feng *et al.* 1992, Krtolica *et al.* 1994, Fang and Loparo

2002, Ling and Deng 2012). Cuando se considera que la matriz de probabilidades de transición \mathbb{P} se conoce parcialmente, la estabilidad en media cuadrática es estudiada, entre otros trabajos, en (Costa *et al.* 2005, Zhang and Boukas 2009, Zhang and Lam 2010). Por otra parte, cuando la matriz \mathbb{P} se considera incierta pero dicha incertidumbre pertenece a un conjunto convexo en donde se conoce cada uno de sus vértices, la estabilidad en media cuadrática es tratada, entre otros, en (Costa *et al.* 1999, Costa *et al.* 2005, Aberkane 2011).

Definición 2.3. (Costa *et al.* 2005) *El sistema (2.3) es dicho ser estable en media cuadrática (MSS⁹) si, para $u_k \equiv 0$ y cualquier condición inicial $x_0 \in X \subseteq \mathbb{R}^{n_x}$, $\theta_0 \in \mathcal{M}$ se cumple lo siguiente:*

$$\sum_{k=0}^{\infty} \mathbb{E}(x_k^T x_k) < \infty \quad (2.13)$$

Otros autores, por ejemplo, (Zhang and Boukas 2009, Definición 2) definen la estabilidad en el sentido estocástico. Por otra parte, en (Lu *et al.* 2013) se presenta la siguiente definición:

Definición 2.4. *El sistema autónomo (2.3), i.e., $u_k \equiv 0$ es estable en media cuadrática si, para condiciones iniciales $x_0 \in X \subseteq \mathbb{R}^{n_x}$ y modo inicial $\theta_0 \in \mathcal{M}$ se cumple lo siguiente:*

$$\lim_{k \rightarrow \infty} \mathbb{E}[x_k^T x_k] = 0 \quad (2.14)$$

Nótese que en general, para sistemas no lineales las definiciones (2.3) y (2.4) no son iguales. Sin embargo, para sistemas lineales con saltos markovianos ambas definiciones son equivalente (Feng *et al.* 1992).

En el presente trabajo, el concepto de MSS viene dado por la Definición (2.3). El principal motivo de esto se debe a que en el diseño de controladores predictivos para MJLS que garanticen MSS, basta con demostrar que la función de coste subyacente al problema de optimización es finita, lo que implicaría que la condición (2.13) sea cumplida.

Teorema 2.1. (Costa *et al.* 2005) *El sistema autónomo (2.3), i.e., $u_k \equiv 0$ es MSS si y solo si, existen un conjunto de matrices simétricas y definidas positivas S_i , $i \in \mathcal{M}$ que satisfacen la siguiente desigualdad:*

$$A_i^T \left(\sum_{j \in \mathcal{M}} \pi_{ji} S_j \right) A_i - S_i < 0 \quad (2.15)$$

⁹Acrónimo proveniente del inglés *mean square stable*.

Definición 2.5. (Costa et al. 2005) Sea $A_{\theta_k} \in \bar{\mathcal{A}}, B_{\theta_k} \in \bar{\mathcal{B}}$. El par $(\bar{\mathcal{A}}, \bar{\mathcal{B}})$ es dicho ser *estabilizable en media cuadrática* si existe $\bar{\mathcal{K}} := \{K^1, K^2, \dots, K^N\}$, tal que, cuando $u_k = K^{\theta_k} x_k$, el sistema en lazo cerrado $x_{k+1} = (A_{\theta_k} + B_{\theta_k} K^{\theta_k}) x_k$ es MSS.

Para el caso sin restricciones puede aplicarse los resultados presente en la sección anterior, con el fin de calcular las ganancias de realimentación K^{θ_k} , $\theta_k \in \mathcal{M}$. Sin embargo, existen diferentes métodos que sirven para calcular dichas ganancias, por ejemplo, en (de Souza 2006) se proponen un procedimiento para obtener una ley de control que puede o no ser dependiente del modo de funcionamiento del sistema. Este procedimiento esta dado en términos de LMI. Como se verá en capítulos posteriores, las ganancias de realimentación K^{θ_k} serán utilizadas como ley de control terminal para estados perteneciente a cierto conjunto terminal invariante. En dicho conjunto, por tanto, el sistema en lazo cerrado será MSS.

2.5. Control Óptimo para MJLS con Incertidumbres sujeto a Restricciones

La siguiente sección está basada en el trabajo (Lu et al. 2013), que a su vez se basa en (Park and Kwon 2002). El objetivo es el diseño de controladores predictivos MPC para MJLS sujeto a incertidumbres politópicas, tanto en las matrices del sistemas como en la matriz de transición \mathbb{P} .

2.5.1. Formulación del problema

Se considera un MJLS sujeto a restricciones duras y con incertidumbres, tanto en las matrices del sistema como en la matriz de probabilidades, *i.e.*:

$$x_{k+1} = A_k(\theta_k)x_k + B_k(\theta_k)u_k, \quad (2.16)$$

donde θ_k representa la variable *modo* de la cadena de Markov, $\theta_k \in \mathcal{M}$. La matriz de probabilidades de transición viene dada por $\mathbb{P}_k := [p_k(i, j)]_{i, j \in \mathcal{M}}$, donde:

$$p_k(i, j) = Pr\{\theta_{k+1} = j | \theta_k = i\}$$

Se supone que las matrices \mathbb{P}_k pertenecen al conjunto $\Omega_{\mathbb{P}}$, *i.e.*:

$$\mathbb{P}_k \in \Omega_{\mathbb{P}} := conv\{\mathbb{P}^1, \mathbb{P}^2, \dots, \mathbb{P}^T\},$$

siendo $\mathbb{P}^t := [p(i, j; t)]_{i, j \in \mathcal{M}}$, $t = 1, 2, \dots, T$. Por su parte las matrices $[A_k(i) B_k(i)]$ son desconocidas pero pertenecen al conjunto convexo:

$$[A_k(i) B_k(i)] \in \Omega_{AB}(i) := \text{conv}\{[A^1(i) B^1(i)], [A^2(i) B^2(i)], \dots, [A^L(i) B^L(i)]\} \quad (2.17)$$

Se asume que el estado x_k y el modo θ_k son medidos en el instante k .

Considere las siguientes restricciones duras en las entradas y salidas del sistema:

$$|[u_{k+n}(i)]_j| \leq [\bar{u}(i)]_j, \quad n \geq 0, \quad i \in \mathcal{M}, \quad j = 1, \dots, n_u \quad (2.18a)$$

$$|\varphi x_{k+n}| \leq \bar{x}_j, \quad n \leq 1, \quad j = 1, \dots, n_\varphi \quad (2.18b)$$

donde $\varphi \in \mathbb{R}^{n_\varphi \times n_x}$, $[\cdot]_j$ denota el j -ésimo elemento de un vector o la j -ésima fila de una matriz. Nótese que las restricciones en los estados no depende del modo de operación, debido a que si –en cierto instante k – el sistema se encuentra en modo $i \in \mathcal{M}$ con una entrada de control determinada por u_k , una vez dicha acción de control es implementada en el proceso, el estado siguiente x_{k+1} es determinado independientemente del modo en que se encontrará el sistema en el instante $k + 1$.

Se propone como objetivo del MPC, diseñar una ley de control que minimice:

$$\min_{u_{k+n|k}, n \geq 0} \max_{[A_{k+n|k}(i) B_{k+n|k}(i)] \in \Omega_{AB}(i), i \in \mathcal{M}, \mathbb{P}_{k+n|k} \in \Omega_{\mathbb{P}}, n \geq 0} J_\infty(k) \quad (2.19)$$

siendo:

$$J_\infty(k) := \mathbb{E}_k \left[\sum_{n=0}^{\infty} \left(\|x_{k+n|k}\|_{Q_{(\theta_{k+n|k})}}^2 + \|u_{k+n|k}\|_{R_{(\theta_{k+n|k})}}^2 \right) \right] \quad (2.20)$$

para $i \in \mathcal{M}$, $Q_{(\theta_{k+n|k})} > 0$, $R_{(\theta_{k+n|k})}$ con matrices de ponderaciones de dimensiones apropiadas.

Nótese que el problema 2.19 corresponde a un problema de optimización mín-máx, en donde, el objetivo es diseñar una acción de control mientras se minimiza el máximo de los valores (*peor caso*) que puede tomar la función de coste para todos los valores considerados de la incertidumbre. Una vez sea resuelto el problema (2.19) para un instante k , solo la primera acción de control u_k es implementada en el proceso, similar al control predictivo convencional.

MPC multi pasos sin restricciones. Para el diseño del MPC, en el citado trabajo se propone en primer lugar el uso de una ley de control por realimentación de múltiples pasos y dependiente del modo la cual será utilizada como ley de control terminal, de la forma:

$$u_{k+n|k} := \begin{cases} u_k & n = 0, \\ K_n(\theta_{k+n|k})x_{k+n|k} & 1 \leq n \leq N - 1, \\ K_N(\theta_{k+n|k})x_{k+n|k} & n \geq N \end{cases} \quad (2.21)$$

donde $K_1(i), K_2(i), \dots, K_N(i)$ son las ganancias de realimentación que deben de ser optimizadas.

Debido a las incertidumbres y las restricciones el problema 2.19 no es computacionalmente tratable, al igual que no lo era el problema (1.42). Por tanto, se basándose en el procedimiento de la Sección 1.5.5, aquí, se obtiene una cota superior de la función de coste esperada $J_\infty(k)$. Entonces, minimizando dicho límite superior se minimiza aproximadamente el índice de coste a horizonte infinito $J_\infty(k)$.

Para ello, en primer lugar considérese la siguiente función de Lyapunov cuadrática¹⁰:

$$\mathcal{V}(x_{k+n|k}) = \|x_{k+n|k}\|_{P_{(\theta_{k+n|k})}}, \quad n \geq 1 \quad (2.22)$$

donde para $n \geq N, i \in \mathcal{M}, P_n(i) = P_N(i)$. Entonces, para derivar un límite superior que acote a la función de coste 2.20 es necesario imponer la siguiente restricción estocástica¹¹:

$$\mathcal{V}(x_{k+n|k}) - \mathbb{E}_{k+n|k}[\mathcal{V}(x_{k+n+1|k})] \geq \|x_{k+n|k}\|_{Q_{(\theta_{k+n|k})}}^2 + \|u_{k+n|k}\|_{R_{(\theta_{k+n|k})}}^2, \quad n \geq 1 \quad (2.23)$$

Lema 2.1. (Lu et al. 2013) *Las restricciones estocásticas (2.23) son cumplidas si existes matrices simétricas $\mathbb{Q}_n(i) := P_n^{-1}, 1 \leq n \leq N, i \in \mathcal{M}$ y cualquier matriz $G_n(i), Y_n(i) = K_n(i)G_n(i), 1 \leq n \leq N$ tal que, (véase (2.24)). Si $n = N$ entonces, $\mathbb{Q}_{n+1}(i) = \mathbb{Q}_N(i), i \in \mathcal{M}$.*

¹⁰Nótese que la ecuación (2.22) es similar a la ecuación (1.44), salvo la dependencia de la variable θ_k presente en la expresión (2.22)

¹¹Note que, la restricción estocástica (2.23) es similar a la desigualdad (1.45), observándose que la diferencia es la consideración de la esperanza matemática en la ecuación (2.23), para así poder tomar en cuenta las probabilidades futuras de todas las realizaciones de la variable θ_k . A demás de la dependencia de las matrices Q y R del modo de funcionamiento del sistema. Puede concluirse que el Lema (2.1) es la generalización del Teorema 1.1 para el caso de MJLS con incertidumbres.

$$\begin{bmatrix} G_n^T(i) + G_n(i) - Q_n(i) & * & \dots & * & * & * \\ A^l(i)G_n(i) + B^l(i)Y_n(i) & p^{-1}(i, 1; t)Q_{n+1}(1) & \dots & 0 & * & * \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ A^l(i)G_n(i) + B^l(i)Y_n(i) & 0 & \dots & p^{-1}(i, M; t)Q_{n+1}(M) & * & * \\ Q^{1/2}G_n(i) & 0 & \dots & 0 & I & * \\ R^{1/2}Y_n(i) & 0 & \dots & 0 & 0 & I \end{bmatrix} > 0, \quad (2.24)$$

$$i \in \mathcal{M}, n = 1, \dots, N, l = 1, \dots, L, t = 1, \dots, T$$

Basado en las restricciones (2.23) una cota superior de la función de coste (2.20) puede ser obtenida de la siguiente forma (véase (Lu *et al.* 2013) para su demostración detallada):

$$J_\infty(k) = \mathbb{E}_k \left[\sum_{n=0}^{\infty} \left(\|x_{k+n|k}\|_{Q_{(\theta_{k+n|k})}}^2 + \|u_{k+n|k}\|_{R_{(\theta_{k+n|k})}}^2 \right) \right] \leq \|x_k\|_{Q_{(\theta_k)}}^2 + \|u_k\|_{R_{(\theta_k)}}^2 + \mathbb{E}_k[\mathcal{V}(x_{k+1|k})] \quad (2.25)$$

Ahora, se puede imponer las siguientes desigualdades:

$$\|x_k\|_{Q_{(\theta_k)}}^2 + \|u_k\|_{R_{(\theta_k)}}^2 \leq \gamma_1$$

y $\mathbb{E}_k[\mathcal{V}(x_{k+1|k})] \leq \gamma_2$ las cuales, usando complemento de Schur pueden fácilmente convertirse en término de LMI, respectivamente como:

$$\begin{bmatrix} \gamma_1 & * & * \\ x_k & Q_{(\theta_k)}^{-1} & 0 \\ u_k & 0 & R_{(\theta_k)}^{-1} \end{bmatrix} \geq 0 \quad (2.26)$$

y

$$\begin{bmatrix} \gamma_2 & * & \dots & * \\ A^l(\theta_k)x_k + B^l(\theta_k)u_k & p^{-1}(\theta_k, 1; t)Q_1(1) & \dots & * \\ \vdots & \vdots & \ddots & \vdots \\ A^l(\theta_k)x_k + B^l(\theta_k)u_k & 0 & \dots & p^{-1}(\theta_k, M; t)Q_1(M) \end{bmatrix} > 0, \quad (2.27)$$

Combinando las ecuaciones (2.25)–(2.27) es posible calcular una cota superior de la función de coste esperada de tiempo infinito, es decir, $J_\infty(k) \leq \gamma_1 + \gamma_2$, para todas las posibles realizaciones de las incertidumbres presente en las matrices del sistema y la matriz de probabilidad de transiciones. Por consiguiente, cuando se minimiza una cota superior del coste, en realidad se

está minimizando el *peor caso* posible de la función de coste esperada de tiempo infinito $J_\infty(k)$. Con esto en mente es posible plantear el siguiente problema de optimización sin restricciones, el cual debe de ser resuelto en cada instante k como:

$$\min_{\gamma_1, \gamma_2, u_k, G_n(i), Q_n(i), Y_n(i)} \gamma_1 + \gamma_2 \quad \text{sueto a (2.24), (2.26) y (2.27)} \quad (2.28)$$

Observación 2.1. Como resultado se puede establecer lo siguiente: Si existe una solución para el problema anterior en el instante k , para un estado inicial x_k y modo inicial θ_k , entonces, también existirá una solución factible para cualquier instante $t \geq k$ y el controlador MPC basado en (2.28) puede garantizar MSS para el sistema sin restricciones en lazo cerrado. La demostración puede consultarse en (Lu et al. 2013, Teorema 1).

Restricciones. En (Lu et al. 2013, Sección 4) se presenta en detalle el diseño de controladores predictivos para MJLS inciertos sujetos a restricciones duras tanto en las entradas como en los estados. A continuación un resumen del problema de optimización bajo restricciones que debe ser minimizado en cada instante de tiempo k

$$\min_{\gamma_1, \gamma_2, u_k, G_n(i), Q_n(i), Y_n(i), W_n(i), X(i), Z(i)} \gamma_1 + \gamma_2 \quad (2.29a)$$

sueto a (2.24), (2.26), (2.27) y a:

$$- [\bar{u}(\theta_k)]_j \leq [u_k(\theta_k)]_j \leq [\bar{u}(\theta_k)]_j, \quad j = 1, \dots, n_u \quad (2.29b)$$

$$- \bar{x}_j \leq \varphi[A^l(\theta_k)x_k + B^l(\theta_k)u_k] \leq \bar{x}_j, \quad j = 1, \dots, n_\varphi \quad (2.29c)$$

$$\begin{bmatrix} 1 & * \\ A^l(\theta_k)x_k + B^l(\theta_k)u_k & W_1(i) \end{bmatrix} \geq 0, \quad \begin{matrix} i \in \mathcal{M} \\ l = 1, \dots, L \end{matrix} \quad (2.29d)$$

$$\begin{bmatrix} G_n^T(i) + G_n(i) - W_n(i) & * \\ A^l(i)G_n(i) + B^l(i)Y_n(i) & W_{n+1}(j) \end{bmatrix} > 0, \quad \begin{matrix} i, j \in \mathcal{M} \\ l = 1, \dots, L \\ p(i, j; t) > 0, \end{matrix} \quad (2.29e)$$

para algún $1 \leq t \leq T$

$$\begin{bmatrix} X_n(i) & Y_n(i) \\ * & G_n^T(i) + G_n(i) - W_n(i) \end{bmatrix} > 0, \quad \begin{matrix} i \in \mathcal{M} \\ n = 1, \dots, N \\ j = 1, \dots, n_u \end{matrix} \quad [X_n(i)]_{jj} \leq [\bar{u}(i)]_j^2 \quad (2.29f)$$

$$\begin{bmatrix} \mathbb{Z}(i) & \varphi W_n(i) \\ * & W_n(i) \end{bmatrix} \geq 0, \quad [\mathbb{Z}_n(i)]_{jj} \leq \bar{x}_j^2 \quad \begin{array}{l} i \in \mathcal{M} \\ n = 2, \dots, N \\ j = 1, \dots, n_\varphi \end{array} \quad (2.29g)$$

Observación 2.1. *Considérese el sistema (2.16) sujeto a las restricciones (2.18), si existe solución factible del problema (2.29a) en el instante k para condiciones iniciales x_k, θ_k , entonces, existirá solución factible para cualquier tiempo $t \geq k$; el MPC basado en (2.29a) garantiza MSS para el sistema en lazo cerrado, véase (Lu et al. 2013, Teorema 2).*

2.5.2. Discusión

Debido al hecho de que se conocerá el estado futuro $x_{k+n|k}$ del sistema, es posible realizar una parametrización de la ley de control $u_{k+n|k}$ según la ecuación (2.21). Para ello, se propone una ley de control por realimentación y dependiente del modo $K_n(i)$ las cuales son obtenidas en la solución del problema de optimización y que sirven para garantizar MSS en lazo cerrado en el sistema. Como es típico en las estrategias MPC clásicas, solo la primera acción de control será aplicada al proceso, y se repetirá entonces, la optimización del problema (2.28) para cada instante k para el nuevo estado x_k . Un ejemplo se presenta en el Apéndice B.3.

2.6. Conclusiones

En este capítulo, en primer lugar, se ha presentado una breve descripción de un tipo de proceso estocástico denominado cadena de Markov. Luego se ha desarrollado controladores predictivos para sistemas multi-modelo cuya transición entre ellos se encuentra modelada a través de una cadena de Markov. Dichos modelos se consideran, en un primer caso, conocidos al igual que la matriz de transiciones \mathbb{P} . Para este caso el controlador predictivo resultante es óptimo y garantiza estabilidad en media cuadrática (MSS). Un segundo procedimiento de diseño fue presentado para sistemas de Markov con incertidumbres tanto en las matrices A, B del modelo, como en la matriz de transición \mathbb{P} . Debido precisamente a dichas incertidumbre, el índice de coste que tiene que ser resultado en cada instante, debe ser expresado como un problema de minimización *mín-máx*. Dicho problema para tiempo infinito no es tratable computacionalmente, por consiguiente, se plantea resolverlo minimizando una cota superior de dicho coste, expresando el mismo en términos de desigualdades matriciales lineales (LMI), extendiendo así, los resultados del trabajo (Kothare

et al. 1996) para el caso de MJLS con incertidumbres sujeto a restricciones duras.

Cabe destacar que esta última formulación del problema de optimización es *sub-óptima*, pero reduce la carga computacional del primer planteamiento, en donde se resuelve la optimización para todas las realizaciones futuras de la variable θ_k para un horizonte dado N . Es evidente que, dicha carga se incrementa exponencialmente a medida que aumenta N .

Capítulo 3

Control Predictivo para Sistemas Lineales con Saltos Markovianos

Resumen: En este capítulo se aborda el control predictivo para sistemas lineales en tiempo discreto con varios modos de funcionamiento. La transición entre dichos sistemas será modelada a través de una cadena de Markov. El sistema se supondrá sujeto a restricciones tanto en las variables de estados como en las acciones de control. El principal objetivo es diseñar una ley de control que minimice una función de coste cuadrática y a la vez cumpla con los conjuntos de restricciones. La condición inicial del sistema y el modo de funcionamiento en el instante actual o el anterior se asumirá que está disponible al controlador en cada instante de muestreo. Conceptos de la teoría de conjuntos invariantes y horizonte deslizante serán incorporados en un enfoque de controladores predictivos para sistemas lineales con saltos markovianos. Parte del contenido de este capítulo aparece en el siguiente trabajo:

- M. Hernández-Mejías, A. Sala, C. Ariño, A. Querol. *Fault-Tolerant Predictive Control for Markov Linear Systems*. Conference on Automation, Quality and Testing, Robotics, 2014. Cluj-Napoca, Romania.
DOI:10.1109/AQTR.2014.6857824.

Ese trabajo corresponde cronológicamente al primer acercamiento al problema objeto de esta tesis. Uno meses después a la referida publicación, el artículo (Patrinos *et al.* 2014) aborda un problema similar, además considerando solu-

ciones explícitas mediante programación multiparamétrica. Por esta razón, se ha considerado pertinente incluir las ideas aquí expuestas dentro de lo que se podría considerar estado del arte.

3.1. Introducción

Existe una enorme cantidad de publicaciones en las que se estudia el problema de control cuadrático sin restricciones para MJLS, véase Sección 2.3 la cual está basada en el libro (Costa *et al.* 2005). Los MJLS con restricciones duras se tratan entre otros en (Costa *et al.* 1999, Lu *et al.* 2013, Patrinos *et al.* 2014), mejorando estudios previos sobre el tema. En el control óptimo para sistemas con saltos markoviano se busca minimizar un coste medio, lo cual resulta en controladores diferentes a los reguladores robustos estudiados en, por ejemplo, el control predictivo mín-máx (Kothare *et al.* 1996), donde se busca que el peor caso posible tenga unas prestaciones mínimas garantizadas.

En este capítulo se pretende desarrollar un control predictivo para MJLS de horizonte finito cumpliendo las restricciones en los estados y acciones de control, minimizando una función de coste cuadrática (en media). Para ello, se describe la obtención del modelo de predicciones para un horizonte dado, así como también, se calculan los conjuntos necesarios (*terminal y factible*) para poder garantizar la estabilidad en el sistema, generalizando los resultados en cuanto a controladores predictivos robustos presentados en (Kothare *et al.* 1996, Kerrigan 2000).

En el presente capítulo, el modo en el que el sistema se encuentra se asumirá conocido, bien sea en el mismo instante en la cual se aplica la acción de control (común en desarrollos de control por realimentación del estado para MJLS (Costa *et al.* 2005, Lu *et al.* 2013)), o bien en el instante *previo*, según se propondrá aquí. Tal conocimiento del modo de operación de la planta (tolerancia a fallos activa) puede adquirirse a través de técnicas adecuadas de diagnóstico (Blanke and Schröder 2003, Frank 2002, Isermann 2011, Liu *et al.* 2012), que no entran en los objetivos del presente estudio.

3.2. Preliminares y Notación

Considérese un MJLS en tiempo discreto dado por:

$$x_{k+1} = A_{\theta_k} x_k + B_{\theta_k} u_k, \quad \theta_k \in \mathcal{M}, \quad (3.1)$$

donde θ_k representa la variable de modo de operación del sistema. A_{θ_k} indica que la matriz de estado A depende el parámetro θ_k . Por su parte, $\mathcal{M} = \{1, 2, \dots, M\}$ son los conjuntos de modos posibles, los cuales se encuentran gobernados por una cadena de Markov discreta con una matriz de probabilidades de transiciones dada por $\mathbb{P} := [\pi_{ji}]$, donde $Pr\{\theta_{t+1} = j | \theta_t = i\} = \pi_{ji}$, $\pi_{ji} > 0$, $\forall i, j \in M$ y $\sum_{j=1}^M \pi_{ji} = 1$. Se asumirá que el proceso (3.1) conmuta entre M modos diferentes tal que:

$$A_{\theta_k} \in \bar{\mathcal{A}} := \{A_1, \dots, A_M\}, \quad B_{\theta_k} \in \bar{\mathcal{B}} := \{B_1, \dots, B_M\} \quad (3.2)$$

La matriz de probabilidades de transición de la cadena de Markov se asume conocida y el estado medido en cada instante de tiempo se considera disponible (realimentación del estado).

Conocimiento de la variable modo del sistema. En cuanto al diagnosticador, se desarrollarán dos escenarios diferentes con respecto al conocimiento de θ_k :

1. Existe un diagnosticador tal que el conocimiento del modo previo θ_{k-1} está disponible en el instante k , o bien
2. el diagnosticador para el instante k , proporciona información del modo actual θ_k del sistema.

Si el modelo utilizado es razonablemente preciso, un diagnosticador podría desarrollarse comprobando cuál de los modos corresponde a la transición observada entre los estados x_{k-1} y x_k . Esta configuración podría ser la base de la primera opción de diagnóstico expuesta anteriormente. Sin embargo, si hay información adicional acerca del modo actual, la segunda opción sería viable, obteniéndose un rendimiento mejorado al eliminarse ese retardo. Se remite al lector a la literatura referida en la introducción para más detalles sobre las técnicas de diagnóstico, las cuales están fuera de los objetivos de la presente tesis.

3.2.1. Notación

Secuencias de modos. Existen M^{N+1} posibles realizaciones de la variable del modo θ_{k+t} desde $t = 0$ a $t = N$, $N \geq 0$. Ordenando las realizaciones (por ejemplo, lexicográficamente), la variable $l \in \mathcal{L}$, $\mathcal{L} = \{1, 2, \dots, M^{N+1}\}$ se usará para representar una realización individual (Scoaert and Mayne 1998). Tal realización se denotará por $\theta^l = \{\theta_0^l \theta_1^l \dots \theta_N^l\}$, y el elemento de la secuencia

θ^l en la posición j se representará por θ_j^l , $j = 0$ hasta $j = N$. También, las matrices que dependen del modo asociadas a tal elemento se denotarán por simplicidad como $A_j^{[l]} := A_{\theta_j^l}$, $B_j^{[l]} := B_{\theta_j^l}$.

Se define como:

$$Pr(\theta^l | \theta_0^l) := \prod_{j=1}^N \pi_{\theta_j^l | \theta_{j-1}^l} \quad (3.3)$$

a la probabilidad de una secuencia en particular θ^l , condicionada al conocimiento de su primer elemento θ_0^l . Adicionalmente, θ_{end}^l representa el ultimo elemento de la secuencia θ^l , i.e., $\theta_{end}^l = \theta_{len(\theta^l)-1}^l$, donde $len(\theta^l)$ representa la longitud de la secuencia θ^l .

Predicciones. Se denotará con $\psi_{k+j|k}^l$ los valores predichos de una variable ψ en un tiempo futuro $k + j$, con la información disponible en el instante k y la secuencia de modo l . Obviamente $\psi_{k|k}^l \equiv \psi_k$ para cualquier l , en particular $x_{k|k}^l$ deberá entenderse como el estado medido x_k .

Notación auxiliar: La siguiente notación es utilizada a lo largo de este capítulo y los siguientes. $\|\bullet\|_W^2 \triangleq \bullet^T W \bullet$.

Manipulación de listas ordenadas de matrices. A lo largo del presente escrito el operador \otimes indicará el producto de listas de matrices y se define como:

Definición 3.1. Dadas dos listas de matrices de apropiadas dimensiones y ordenadas adecuadamente, $\bar{C} = \{C_1, \dots, C_p\}$ y $\bar{D} = \{D_1, \dots, D_q\}$, su producto $\bar{C} \otimes \bar{D}$ se define como una lista de matrices \bar{R} formadas por $p \times q$ elementos, donde el elemento i , denotado como \bar{R}_i , $i = 1, \dots, p \times q$ está formado por el producto de C_{i_1} y D_{i_2} donde los enteros i_1 y i_2 son la solución de $i = p(i_2 - 1) + i_1$ para $1 \leq i_1 \leq p$, $1 \leq i_2 \leq q$.

Por ejemplo, la definición anterior implica:

$$\bar{R} := \{C_1, C_2\} \otimes \{D_1, D_2, D_3\} = \{C_1 D_1, C_2 D_1, C_1 D_2, C_2 D_2, C_1 D_3, C_2 D_3\}$$

Conversión de lista a una matriz. En algunas operaciones que se utilizarán más adelante, las listas de matrices será necesario introducirlas en un *bloque* de matrices. En este caso, abusando de la notación, se asumirá que los elementos de la lista son concatenados verticalmente, por ejemplo, $\bar{C} = \{C_1, C_2\}$ con C_i de dimensiones 2×3 y considerando matrices arbitrarias X, Y, Z de tamaño

4×7 , 5×3 y 5×7 , respectivamente, \bar{C} se debe interpretar como:

$$\begin{pmatrix} \bar{C} & X \\ Y & Z \end{pmatrix} = \begin{pmatrix} C_1 & X \\ C_2 & X \\ Y & Z \end{pmatrix} \quad (3.4)$$

3.2.2. Control Óptimo sin Restricciones para MJLS

Las ecuaciones recursivas del control óptimo en los dos escenarios de diagnóstico se discuten a continuación:

Modo Actual Conocido

Los desarrollos relacionados con el control óptimo cuadrático para MJLS sin restricciones aparecen en la Sección 2.3 para el caso en el que el modo actual del sistema se conoce.

Conocimiento del Modo Previo

Si la información conocida del modo en el instante k es $\theta_{k-1} = i$ y la función de coste es cuadrática, la ecuación de Bellman resultante para dicho caso, resulta:

$$J_{k,N}^{opt,i}(x_k) = \min_{u_k \in \mathbb{U}} \mathbb{E}_k \left(L(x_k, u_k, \theta_k) + J_{k+1,N}^{opt,\theta_k}(x_{k+1}|k) \right), \quad (3.5)$$

la cual está compuesta por un primer sumando dado por:

$$\mathbb{E}_k(L(x_k, u_k, \theta_k)) = \sum_{j \in \mathcal{M}} \pi_{ji} (x_k^T Q^j x_k + u_k^T R^j u_k), \quad (3.6)$$

y un sumando a la derecha dado por:

$$\mathbb{E}_k(J_{k+1,N}^{opt,\theta_k}(x_{k+1}|k)) = \sum_{j \in \mathcal{M}} \pi_{ji} J_{k+1,N}^{opt,j}(A_j x_k + B_j u_k) \quad (3.7)$$

Esto significa que el estado sucesor se debería calcular con el modo actual j , el cual se desconoce en el instante k , por lo que el resultado se promedia adecuadamente. La ecuación de Riccati resultante estaría dada por:

$$x_k^T S_k^i x_k = x_k^T \bar{Q}^i x_k + u_k^T \bar{R}^i u_k + \sum_{j \in \mathcal{M}} \pi_{ji} (A_j x_k + B_j u_k)^T S_{k+1}^j (A_j x_k + B_j u_k), \quad (3.8)$$

$$S_k^i = \bar{Q}^i + \sum_{j \in \mathcal{M}} \pi_{ji} A_j^T S_{k+1}^j A_j - \left(\sum_{j \in \mathcal{M}} \pi_{ji} A_j^T S_{k+1}^j B_j \right) \left(\bar{R}^i + \sum_{j \in \mathcal{M}} \pi_{ji} B_j^T S_{k+1}^j B_j \right)^{-1} \left(\sum_{j \in \mathcal{M}} \pi_{ji} B_j^T S_{k+1}^j A_j \right) \quad (3.10)$$

con $\bar{Q}^i = \sum_{j \in \mathcal{M}} \pi_{ji} Q^j$, $\bar{R}^i = \sum_{j \in \mathcal{M}} \pi_{ji} R^j$, produciendo una ley de control que depende del modo en el instante inmediatamente anterior:

$$u_k^{opt,i} := -K_k^i x_k, \quad K_k^i = - \left(\bar{R}^i + \sum_{j \in \mathcal{M}} \pi_{ji} B_j^T S_{k+1}^j B_j \right)^{-1} \sum_{j \in \mathcal{M}} \pi_{ji} B_j^T S_{k+1}^j A_j x_k \quad (3.9)$$

El cálculo recursivo del índice de coste viene dado por la expresión (3.10).

Horizonte infinito. Iterando las ecuaciones en ambos escenarios hasta su convergencia, se obtiene el caso límite S_∞^i (abreviado a S^i) y la acción de control óptima dada por:

$$u_k^{opt,i} = -K^i x_k, \quad J^{opt,i}(x_k) = x_k^T S^i x_k, \quad (3.11)$$

siendo $J^{*,i}(x_k) = \mathbb{E}_0 \left(\sum_{k=0}^{\infty} L(x_k, u_k, \theta_k) \right)$.

3.2.3. Descripción del Problema

El objetivo principal de este capítulo es desarrollar una ley de control predictiva que sea capaz de dirigir el estado del sistema al origen o a una región terminal *invariante* vecina de éste, para *cualquier* realización futura de θ^l $l \in \mathcal{L}$, adaptando el enfoque *control predictivo con horizonte deslizante* y la teoría de conjuntos invariantes al caso de sistemas lineales con saltos markovianos en consideración.

Las restricciones tanto en el estado como en la acción de control se definirán como: $x_k \in \mathbb{X}$, $u_k \in \mathbb{U}$ siendo \mathbb{X} un subconjunto convexo, cerrado de \mathbb{R}^{n_x} y \mathbb{U} un subconjunto convexo, compacto de \mathbb{R}^{n_u} , ambos conteniendo el origen en su interior.

Una vez obtenido el controlador terminal, se desea determinar el conjunto de estados iniciales, tal que la estrategia del control predictivo sea *factible* para un horizonte dado para toda realización de modo posible. En dichos conjuntos

de estados, la ley de control predictiva se encargará de conducir el sistema a una región terminal (previamente obtenida).

3.3. Desarrollo del Control Predictivo para MJLS

En esta sección se presentará la solución del problema anteriormente establecido, detallando la obtención del modelo de predicción de estados, en el cual debe incluir todas las posibles realizaciones de θ_k , i.e., θ^l $l \in \mathcal{L}$. Se calcularán los conjuntos *terminal* \mathbb{T}_f y *factible* \mathcal{K}_N^f , necesarios para asegurar estabilidad y factibilidad del problema en estudio y por último, la formulación del problema del control predictivo en termino de programación cuadrática (QP), extendiendo el planteamiento QP del control predictivo convencional.

3.3.1. Modelo de Predicción

Dada una particular realización de la cadena de Markov θ^l , $l \in \mathcal{L}$, para un estado inicial conocido x_0 y una secuencia de futuras acciones de control $\mathcal{U} := \{u_k^l, u_{k+1}^l, \dots\}$, los estados futuros predichos en el instante $k + t$ serán denotados como $x_{k+t|k}^{[l]}$ y vienen dado por¹²:

$$x_{k+t|k}^{[l]} = \prod_{j=k}^{k+t-1} A_j^{[l]} x_k + \sum_{d=k}^{k+t-1} \left(\prod_{j=d+1}^{k+t-1} A_j^{[l]} \right) B_d^{[l]} u_d^{[l]}, \quad (3.12)$$

donde puede observarse que las entradas de control $u_d^{[l]}$ son dependientes de la realización con el fin de tomar en cuenta que los valores futuros de x y θ serán conocidos en los instantes futuros, todo ello, siguiendo las ideas presentadas en (Scolaert and Mayne 1998). Adicionalmente, es necesario añadir una nueva restricción cuando los elementos anteriores al instante d de la secuencia θ^{l_1} y θ^{l_2} sean idénticos, las acciones de control también tienen que ser idéntica ("restricción de casualidad"), $u_d^{[l_1]} = u_d^{[l_2]}$.

Agrupando de forma adecuada las predicciones, es posible conformar un vector de estados futuros con: una predicción para $x_{k+1|k}$, M para $x_{k+2|k}$, M^2 para $x_{k+3|k}$ y así sucesivamente hasta completar el horizonte N . En el Capítulo 5 se expresan estas predicciones en formal de árbol de escenarios.

El modelo de predicción es diferente en cada uno de los dos escenarios de

¹²La notación de producto de matrices (no conmutativo) se supone realizada en el siguiente orden $\prod_{j=a}^b A_j = A_b A_{b-1} \dots A_a$, es decir, los valores más altos de los índices multiplican por el lado izquierdo.

diagnóstico considerado.

Modo Actual Conocido

Se denotará con la variable ψ_k al vector de predicciones de estados futuros. Este vector será función del modo θ_k y el estado x_k . Para cada modo inicial, $i \in \mathcal{M}$, el vector de predicciones puede calcularse como: $\psi_k = \Gamma_N^i v_k$, donde v_k es el vector de variable de decisión formado por x_k y todas las futuras entradas $u_{d'}^l$, para $d = k, \dots, k + N - 1$ y $l = 1, \dots, M^d$. La Matriz Γ_N^i puede obtenerse aplicando la siguiente recursión:

$$\Gamma_k^i = \begin{pmatrix} \left(\begin{array}{c|c} \Gamma_{k-1}^i & \mathbf{0} \\ \hline \Upsilon_k^i & \end{array} \right), & (3.13)$$

$$\Upsilon_k^i = \left(\bar{\mathcal{A}} \otimes \Upsilon_{k-1}^i \quad \text{blkdg}(\bar{\mathcal{B}} \otimes \bar{\mathcal{S}}_k) \right), \quad (3.14)$$

donde las matrices $\bar{\mathcal{A}}$ y $\bar{\mathcal{B}}$ vienen dadas por la Definición 3.2. Por su parte, $\text{blkdg}(\cdot)$ denotará la disposición de los elementos de una lista en forma diagonal por bloques. $\bar{\mathcal{S}}_k$ es una lista de matrices con un total de M^{k-1} elementos, siendo cada uno de los elemento igual a una matriz identidad de dimensión $m \times m$ y $\mathbf{0}$ una matriz de ceros de dimensión apropiada. La recursión es inicializada a $\Upsilon_1^i = \Gamma_1^i = (A_i \ B_i)$.

Por ejemplo, considérese un sistema con dos modos de operación ($M = 2$) y un horizonte de predicción $N = 3$, el modelo de predicción tiene la siguiente estructura:

$$\Gamma_3^i = \begin{bmatrix} A_i & B_i & 0 & 0 & 0 & 0 & 0 & 0 \\ A_1 A_i & A_1 B_i & B_1 & 0 & 0 & 0 & 0 & 0 \\ A_2 A_i & A_2 B_i & 0 & B_2 & 0 & 0 & 0 & 0 \\ A_1 A_1 A_i & A_1 A_1 B_i & A_1 B_1 & 0 & B_1 & 0 & 0 & 0 \\ A_2 A_1 A_i & A_2 A_1 B_i & A_2 B_1 & 0 & 0 & B_2 & 0 & 0 \\ A_1 A_2 A_i & A_1 A_2 B_i & 0 & A_1 B_2 & 0 & 0 & B_1 & 0 \\ A_2 A_2 A_i & A_2 A_2 B_i & 0 & A_2 B_2 & 0 & 0 & 0 & B_2 \end{bmatrix},$$

donde la primera columna representa la respuesta libre, y el resto la respuesta forzada para cada secuencia (asumiendo la adecuada expansión de las acciones de control: una en el primer instante, dos en el segundo y cuatro en el tercero).

Modo Anterior Conocido

Para este caso, la acción de control no puede depender del modo actual del sistema, por consiguiente, dadas dos realizaciones θ^{l_1} y θ^{l_2} , las variables de entradas deben verificar $u_d^{[l_1]} = u_d^{[l_2]}$ si los primeros $d - 1$ elementos de las secuencias son idénticos. Las matrices para la implementación de la ecuación de predicción pueden calcularse con la misma recursión, pero inicializada a

$$\Upsilon_1^i = \Gamma_1^i = (\bar{A} \bar{B})$$

3.3.2. Conjuntos Terminal y Factible

Es bien sabido que la técnica de control predictivo necesita tener asociado ciertos conjuntos para poder probar la estabilidad y factibilidad del problema de optimización (Mayne *et al.* 2000). A continuación se extenderá las definiciones básicas presentadas en la Sección 1.2 para obtener dichos conjuntos en cada uno de los dos escenarios de diagnóstico en consideración.

Modo Actual Conocido

Definición 3.2. (Conjunto a un paso con realimentación del modo) *Considérese un conjunto arbitrario Ω , el conjunto a un paso con realimentación de modo $\mathbb{Q}(\Omega)$, es el conjunto de estados en \mathbb{R}^{n_x} tal que, una acción de control prefijada dependiente del modo $u^i = K^i x$ conduce al sistema a Ω en un paso para todo $i \in \mathcal{M}$:*

$$\mathbb{Q}(\Omega) := \{x \in \mathbb{R}^{n_x} \mid u^i = K^i x \in \mathbb{U}, [A_i + B_i K^i]x \in \Omega, \forall i \in \mathcal{M}\} \quad (3.15)$$

Definición 3.3. *El conjunto terminal con realimentación del modo \mathbb{T}_f para la ley de control sin restricciones (3.11) se obtendrá aplicando el Algoritmo 1, estableciendo:*

$$\mathcal{K}_0 = \{x \in \mathbb{X} \mid -K^i x \in \mathbb{U}, \forall i \in \mathcal{M}\}$$

como semilla y obteniendo $\mathcal{K}_{p+1} = \mathbb{Q}(\mathcal{K}_p) \cap \mathbb{X}$ hasta su convergencia ($\mathcal{K}_{p+1} = \mathcal{K}_p$). Por lo tanto, $\mathbb{T}_f = \mathcal{K}_p$.

Nótese que, puede no existir tal convergencia, por consiguiente, no puede garantizarse la existencia del conjunto \mathbb{T}_f . Sin embargo, en este trabajo se asumirá dicha convergencia.

Definición 3.4. (Conjunto controlable a un paso con realimentación del modo) *Considérese un conjunto arbitrario Ω , el conjunto controlable a un paso con reali-*

mentación del modo $\mathbb{Q}^C(\Omega)$, se define como:

$$\mathbb{Q}^C(\Omega) := \{x \in \mathbb{R}^{n_x} \mid \exists u^i \in \mathbb{U}, [A_i x + B_i u^i] \in \Omega \forall i \in \mathcal{M}\} \quad (3.16)$$

Definición 3.5. (Conjunto controlable a N pasos con realimentación del modo) El conjunto controlable a N pasos con realimentación del modo $\mathcal{K}_N^f(\mathbb{X}, \mathbb{T}_f)$ es el conjunto de estados en \mathbb{X} para lo cuales existe una ley de control admisible dependiente del modo, tal que, el estado del sistema es conducido a una región terminal $\mathbb{T}_f \subset \mathbb{R}^n$ en como mucho N pasos, mientras se mantiene la evolución del estado dentro del conjunto admisible \mathbb{X} en todos los instantes intermedios, para cualquier secuencia de modos posible.

$$\mathcal{K}_N^f(\mathbb{X}, \mathbb{T}_f) := \left\{ x_0 \in \mathbb{X} \mid \exists \{u_k^{[l]} \in \mathbb{U}\}_{k=0}^{N-1} : \{x_{k|0}^{[l]} \in \mathbb{X}\}_{k=0}^{N-1} \text{ y } x_{N|0}^{[l]} \in \mathbb{T}_f, \forall l \in \mathcal{L} \right\}, \quad (3.17)$$

donde $x_{k|0}^{[l]}$ es la predicción en el instante k dado por (3.12), para cada realización de θ , utilizando la acción de control dependiente del modo $u_k^{[l]}$.

Este conjunto se obtiene recursivamente durante N iteraciones del siguiente procedimiento:

$$\mathcal{K}_p^f(\mathbb{X}, \mathbb{T}_f) = \mathbb{Q}^C(\mathcal{K}_{p-1}^f(\mathbb{X}, \mathbb{T}_f)) \cap \mathbb{X},$$

comenzando con el conjunto $\mathcal{K}_0^f(\mathbb{X}, \mathbb{T}_f) = \mathbb{T}_f$.

Modo previo conocido

Cuando la información del modo se conoce en el instante *previo* al calculo de la acción de control y, la ganancia de controlador (K^i) depende de tal modo *previo*, el conjunto a un paso asociado al calculo del conjunto terminal necesita redefinirse como:

$$\tilde{\mathbb{Q}}(\Omega) := \{x \in \mathbb{R}^n \mid u^i = K^i x \in \mathbb{U}, [A_j + B_j K^i]x \in \Omega, \forall (i, j) \in \mathcal{M}^2\} \quad (3.18)$$

Igualmente, el conjunto controlable a un paso debe definirse como:

$$\tilde{\mathbb{Q}}^C(\Omega) := \{x \in \mathbb{R}^n \mid \exists u \in \mathbb{U}, [A_i x + B_i u] \in \Omega, \forall i \in \mathcal{M}\}, \quad (3.19)$$

debido a que el modo del sistema en el instante a aplicar las acciones de control no es conocido, una acción de control u independiente del modo actual, debe de conducir el próximo estado de la planta a Ω . Por lo tanto, el conjunto $\tilde{\mathbb{Q}}^C(\Omega)$ es idéntico al conjunto controlable a un paso robusto que aparece en

(Kerrigan 2000). Evidentemente, en este caso, la ausencia de la información del modo actual, produciría conjuntos más pequeños que los conjuntos “con realimentación del modo” definidos anteriormente.

Cabe destacar que, pueden existir estados fuera de los conjuntos factibles obtenidos en ambas situaciones que pudieran resultar factibles para un conjunto limitado de secuencias de modos. Esto será el principal objetivo de interés del Capítulo 6. Los conjuntos obtenidos son mayores, pero no se puede garantizar al 100 % la factibilidad y, por tanto, no se puede probar estabilidad en media.

3.3.3. Cálculo de la acción de control

Modo Actual Conocido

Considere el problema de control predictivo con restricciones de horizonte finito N , el cual consiste en obtener una secuencia acciones de control $\mathcal{U} := \{u_k^{[l]}, u_{k+1}^{[l]}, \dots\}$, minimizando un coste medio utilizando la ecuación de predicción (3.12). Dicho problema de control se formula de la siguiente forma, como media de un índice cuadrático predictivo:

$$J_{k,k+N}^i = \sum_{l \in \mathcal{L}} Pr(\theta^l | \theta_k^l = i) \left(\sum_{t=0}^{N-1} \|x_{k+t|k}^{[l]}\|_Q^2 + \|u_{k+t|k}^{[l]}\|_R^2 + \|x_{k+N|k}^{[l]}\|_{S^{\theta_{k+N}^l}}^2 \right) \quad (3.20)$$

Para asegurar estabilidad, de modo análogo al control predictivo convencional, el coste terminal $S^{\theta_{k+N}^l}$ en la expresión (3.20) debe ser el coste en (3.11) el cual es obtenido con (2.10) tras convergencia. Asimismo, las siguientes restricciones transitorias y terminales deben aplicarse para las predicciones resultantes de toda realización posible $l \in \mathcal{L}$:

$$x_{k+t|k}^{[l]} \in \mathbb{X}, \quad u_{k+t|k}^{[l]} \in \mathbb{U}, \quad x_{k+N|k}^{[l]} \in \mathbb{T}_f \subset \mathbb{X} \quad (3.21)$$

El esquema anteriormente propuesto es estable *–en media cuadrática–*, en el sentido de (Zhang and Boukas 2009) si el conjunto factible contiene puntos distintos del origen.

Teorema 3.1. *La implementación mediante la técnica de horizonte deslizante del controlador óptimo anterior es MSS en lazo cerrado, para todo $x_0 \in \mathcal{K}_N^f(\mathbb{X}, \mathbb{T}_f)$, $x_0 \neq 0$ si Q es definida positiva y R semi-definida positiva.*

Demostración: Básicamente la prueba se basa en la siguiente argumentación: En primer lugar, es evidente que el conjunto controlable en N -pasos con realimentación de modo $\mathcal{K}_N^f(\mathbb{X}, \mathbb{T}_f)$, para una horizonte N dado, es un conjunto invariante bajo control. Dado que dicho conjunto es obtenido a partir de un conjunto terminal invariante admisible del sistema, los conjuntos de estados factibles cumplen la propiedad que $\mathcal{K}_{N-1}^f(\mathbb{X}, \mathbb{T}_f) \subseteq \mathcal{K}_N^f(\mathbb{X}, \mathbb{T}_f)$, por tanto es posible asegurar que $x_{k+1|k} \in \mathcal{K}_{N-1}^f(\mathbb{X}, \mathbb{T}_f) \subset \mathbb{X}$ para todo $x_0 \in \mathcal{K}_N^f(\mathbb{X}, \mathbb{T}_f)$. Esto garantiza la factibilidad del problema QP en todo instante, por lo tanto, el coste medio estimado denotado como $J^{opt}(x_k, \theta_k)$ es finito. Tal coste puede actuar como una función de Lyapunov para probar estabilidad en media si se cumple el decrecimiento medio (Costa *et al.* 2005):

$$\mathbb{E}_k(J^{opt}(x_k, \theta_k) - J^{opt}(x_{k+1}, \theta_{k+1})) \geq 0$$

La solución óptima cumple esta condición de decrecimiento en media: esto puede demostrarse introduciendo la esperanza matemática \mathbb{E}_k en las argumentaciones para probar estabilidad en control predictivo estándar que aparecen en (Goodwin *et al.* 2006). \square

Modo anterior conocido

Para este caso, en el problema a resolver se necesita reemplazar la condición en (3.20) a $\theta_{k-1}^l = i$, y $u_d^{[l_1]} = u_d^{[l_2]}$, para todas aquellas secuencias en donde coincida el primer $d - 1$ elemento, es decir, se cambia el significado de las variables de decisión de control. Análogamente a (3.5), el coste terminal debe ser también dependiente del modo anterior $S^{\theta_{t+N-1}^l}$.

En la prueba de estabilidad en media cuadrática, el conjunto factible para el problema de control QP será el conjunto controlable robusto a N -pasos obtenido en (Kerrigan 2000) y la función de Lyapunov para probar dicha estabilidad debe obtenerse con $J^{opt}(x_k, \theta_{k-1})$. En resumen, los resultados de optimalidad y estabilidad sobre modo actual conocido también pueden adaptarse a este segundo escenario de diagnóstico de forma análoga.

3.3.4. Cuestiones Computacionales

El controlador desarrollado plantea dos situaciones a considerar en cuanto al coste computacional necesario para la resolución del problema de optimización: (a) el cálculo *fuera de línea* de los conjuntos terminal y factible y, (b) la

resolución *en línea* del problema QP para el cálculo de las secuencias de control.

En cuanto a la primera de las cuestiones, el conjunto **terminal invariante con realimentación del modo** propuesto en este capítulo, tiene exactamente los mismos requisitos de computo que los conjuntos politópicos robusto presentados en el trabajo (Kerrigan 2000). Por otro lado, los conjuntos factibles (**controlables con realimentación del modo**) requieren un espacio aumentado en las variables de entrada (el número de entradas se multiplica por el número de modos), con una dimensionalidad más alta para las proyecciones que la de los conjuntos robustos de (Kerrigan 2000), lo que resulta más costoso de computar.

En cuanto a la optimización en línea, el problema más importante del enfoque propuesto, es el hecho de que el número de variables de decisión y las restricciones aumenta exponencialmente a medida que aumenta el horizonte N . Por lo tanto, grandes horizontes no son alcanzables en la práctica, excepto para los sistemas con pocos estados y número reducido de modos de funcionamiento. Podrían llevarse a cabo, por ejemplo, varios enfoques distintos para mitigar la carga computacional:

1. Una vez sean obtenidos los conjuntos controlables con realimentación del modo (o, alternativamente, robusto), una acción de control *subóptima* puede ser calculada considerando en el índice de coste y restricciones terminales, únicamente un número reducido de combinaciones de trayectorias del estados, por ejemplo, tener en cuenta las trayectorias más probables. Esto requeriría eliminar filas de las matrices Γ_N^i (Capítulo 6).
2. La complejidad computacional puede ser aliviada introduciendo en el problema de optimización algún conservadurismo basado en árbol de escenario (Capítulos 4–5).
3. Es bien conocido que el coste óptimo de un problema QP con restricciones lineales es cuadrático por regiones. No obstante cotas subóptimas cuadráticas (no por regiones) pueden obtenerse mediante LMIs (Sección 2.5) obteniéndose conjuntos invariantes elipsoidales. Podría usarse cualquier poliedro contenido en dichos elipsoides como conjunto terminal, junto con la correspondiente cota cuadrática como coste terminal.

3.4. Ejemplo Ilustrativo

Para ilustrar los resultados propuestos en este capítulo, considere el sistema térmico mostrado en la Figura 3-1, el cual consta de dos habitaciones adyacentes calentadas por radiadores independientes.

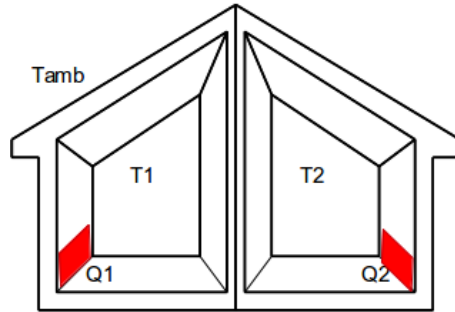


Figura 3-1: Sistema térmico.

Modelo: El modelo matemático que describe la dinámica del sistema viene dada por:

$$\begin{aligned} MC_1 \dot{T}_1 &= K_{12}(-T_1 + T_2) + K_{1a}(T_{amb} - T_1) + Q_1 \\ MC_2 \dot{T}_2 &= K_{12}(T_1 - T_2) + K_{2a}(T_{amb} - T_2) + Q_2 \end{aligned} \quad (3.22)$$

Los valores de los parámetros son: $K_{1a} = K_{2a} = 0,8333 \text{ KW}/^\circ\text{C}$, $K_{12} = 0,1750 \text{ KW}/^\circ\text{C}$ y $MC_1 = MC_2 = 200 \text{ W}/^\circ\text{C}$. El punto de operación nominal se establece a: $T_{amb}^0 = 9^\circ\text{C}$, $T_1^0 = T_2^0 = 15^\circ\text{C}$, $Q_1^0 = Q_2^0 = K_{1a}(T_1^0 - T_{amb}^0) = 5 \text{ KW}$. La temperatura exterior (T_{amb}) se asumirá constante e igual a su punto de operación.

Se considerarán dos modos de funcionamientos: modo 1, en el cual los radiadores Q_1 y Q_2 funcionan correctamente; modo 2, en donde el radiador uno Q_1 falla, es decir, ($Q_1 = 0$).

Para la temperatura exterior constante T_{amb} , tomando como origen de coordenadas el punto de operación nominal, empleando variables incrementales alrededor de este se tiene:

$$\Delta T_1 = T_1 - T_1^0, \Delta T_2 = T_2 - T_2^0, \Delta Q_1 = Q_1 - Q_1^0, \Delta Q_2 = Q_2 - Q_2^0 \quad (3.23)$$

el modelo (con $\Delta T_{amb} = 0$) en Modo 1 es el siguiente:

$$\begin{aligned}\Delta \dot{T}_1 &= -\frac{1}{M_{C1}}(K_{12} + K_{1a})\Delta T_1 + \frac{1}{M_{C1}}K_{12}\Delta T_2 + \frac{1}{M_{C1}}\Delta Q_1 \\ \Delta \dot{T}_2 &= \frac{1}{M_{C2}}K_{12}\Delta T_1 - \frac{1}{M_{C2}}(K_{12} + K_{2a})\Delta T_2 + \frac{1}{M_{C2}}\Delta Q_2\end{aligned}\quad (3.24)$$

En Modo 2, como el radiador 1 falla, i.e., $Q_1 = 0$, es fácil ver que en la ecuación (3.23), el modelo tiene una constante $\Delta Q_1 = -Q_1^0 = -5 \text{ KW}$, por lo tanto, el sistema queda afectado por una perturbación:

$$\Delta \dot{T}_1 = -\frac{1}{M_{C1}}(K_{12} + K_{1a})\Delta T_1 + \frac{1}{M_{C1}}K_{12}\Delta T_2 - \frac{5}{M_{C1}} \quad (3.25)$$

$$\Delta \dot{T}_2 = \frac{1}{M_{C2}}K_{12}(\Delta T_1 - \Delta T_2) - \frac{1}{M_{C2}}K_{2a}\Delta T_2 + \frac{1}{M_{C2}}\Delta Q_2 \quad (3.26)$$

El término constante en la ecuación (3.25) se modelará como un tercer estado $\dot{x}_3 = 0$ con valor inicial constante en todo instante $x_3 = 1$, por lo que la pérdida de potencia de 5 KW en (3.25) se reemplazará por $-5/M_{C1} \cdot x_3 = -0,4756 \cdot x_3$.

Tras discretizar las ecuaciones del sistema con un periodo de muestreo de 20 segundos, las matrices del modelo en representación interna normalizada para cada modo son:

$$\begin{aligned}A_1 &= \begin{bmatrix} 0,9042 & 0,0158 & 0 \\ 0,0158 & 0,9042 & 0 \\ 0 & 0 & 1 \end{bmatrix}, & B_1 &= \begin{bmatrix} 0,0951 & 0,0008 \\ 0,0008 & 0,0951 \\ 0 & 0 \end{bmatrix} \\ A_2 &= \begin{bmatrix} 0,9042 & 0,0158 & -0,4756 \\ 0,0158 & 0,9042 & -0,0041 \\ 0 & 0 & 1 \end{bmatrix}, & B_2 &= \begin{bmatrix} 0 & 0,0008 \\ 0 & 0,0951 \\ 0 & 0 \end{bmatrix}\end{aligned}$$

La matriz de probabilidad de transición para los cálculos se fijará a:

$$\mathbb{P} = \begin{bmatrix} 0,95 & 0,05 \\ 0,05 & 0,95 \end{bmatrix} \quad (3.27)$$

Dado que se tienen dos consignas de temperaturas para las dos habitaciones, es imposible alcanzar simultáneamente las dos referencias con error en equilibrio nulo en Modo 2 (sólo opera un único radiador) en régimen estacionario, incluso anexando acción integral. La magnitud de dicho error dependerá de la temperatura exterior.

Restricciones y especificaciones de rendimiento: Las restricciones en los radiadores serán los límites físicos establecidos en $[0,10]$ KW, i.e., ± 5 KW en coordenadas incrementales. Por su parte, las temperaturas en el régimen transitorio deberán evitar alcanzar incrementos de $\pm 16^\circ C$ sobre el punto de operación escogido.

Con el fin de seleccionar un horizonte de predicción N adecuado, también se especifica que el control predictivo debe ser factible para cualquier temperatura inicial en un rango de $\pm 8^\circ C$ alrededor del punto de operación nominal. Las matrices de pesos del índice de coste vienen dada por:

$$R = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \quad Q = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 1e^{-9} \end{bmatrix}$$

Se calculará y simulará el control en el caso en el que se conoce el modo actual de funcionamiento del sistema, en el mismo instante de aplicar las actuaciones al proceso.

Obtención de los conjuntos terminal y factible: Resolviendo recursivamente (2.9) se obtiene la *ley de control terminal* $u_t^{i,*}$. Dado que el tercer estado es no controlable, éste será el responsable de la presencia de error de posición en cada instante cuando el radiador 1 falle. Esto provocaría que los valores esperados del coste acumulado (2.9) tiendan a infinito, por lo que la recursión no convergería. Para evitar este problema y sin perder el significado práctico del modelo ampliado, el polo en $z = 1$ se ha reemplazado por un polo en $z = 0,999$ solo para el cálculo del controlador terminal¹³.

Denotando $\psi = (\Delta T_1 \ \Delta T_2 \ x_3)^T$ al vector de estado del sistema, la *ley de control terminal dependiente del modo* está dada por:

$$\begin{aligned} u_t^{1,*} &= \begin{pmatrix} 0,5559 & 0,0659 & -0,6198 \\ 0,0654 & 0,5388 & -0,1364 \end{pmatrix} \psi_t, \\ u_t^{2,*} &= \begin{pmatrix} 0 & 0 & 0 \\ 0,0830 & 0,5414 & -0,4543 \end{pmatrix} \psi_t \end{aligned} \quad (3.28)$$

Una vez resuelta la ecuación de Riccati para ambos modos, el modelo se retorna al integrador puro (polo en $z = 1$) para los pasos posteriores del di-

¹³El desplazamiento de los polos en el eje imaginario de la planta en bucle abierto (Skogestad and Postlethwaite 2007) se utiliza frecuentemente en la técnica LQR estándar con acción integral en sistemas con perturbaciones constantes, así como también, en diseño de controladores \mathcal{H}_∞ .

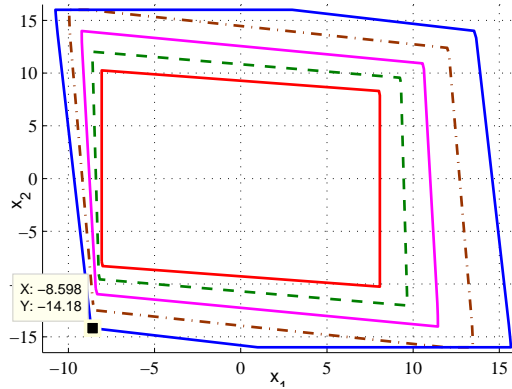


Figura 3-2: Conjunto Terminal (línea sólida roja), conjunto inicial de estados factibles para $N = 1$ (línea discontinua verde), y $N = 4$ (línea sólida azul).

seño. Los conjuntos terminal y factible se calcularon utilizando (3.15) y (3.16) aplicando el Algoritmo 1, con la ayuda de *Multi-Parametric Toolbox 3.0* (MPT) (Herceg *et al.* 2013).

Seleccionando el horizonte de predicción $N = 4$, el conjunto de estados factibles resultante \mathcal{K}_N^f incluye las especificaciones de temperaturas iniciales de $\pm 8^\circ\text{C}$ (Figura 3-2), por lo que no se necesita incrementar dicho horizonte de predicción, con el fin de cumplir los requerimientos impuestos. El conjunto terminal \mathbb{T}_f y los conjuntos factibles para $N = 1, 2, 3, 4$ se muestran en la Figura 3-2.

3.4.1. Resultados para diferentes simulaciones

En la Tabla 3.1 se presenta el tiempo de computo medio necesario para resolver una iteración del problema de control predictivo con saltos markovianos (CPM) (3.20). Dichas simulaciones fueron realizadas en una computadora con sistema operativo Linux (Ubuntu 14.04, 8 GB RAM, i7-920 CPU), usando Matlab[®] R2013a. Puede observarse, que el coste computacional crece significativamente a medida que aumenta el horizonte de predicción N .

Adicionalmente, con el fin de mostrar el crecimiento de la carga compu-

Cuadro 3.1: Tiempo de Computo (TC). Segunda fila: 2 modos de funcionamiento. Tercera fila: 3 modos de funcionamiento.

Horizonte	$N = 2$	$N = 4$	$N = 6$	$N = 8$	$N = 10$
TC-2(seg)	0.0123	0.0348	0.1796	2.9091	164.7176
TC-3(seg)	0.0258	0.1866	12.1573	295.6370	*

tacional se procede a anexar un *tercer modo* de operación al sistema, el cual consiste en la pérdida total del radiador 2, $Q_2 = 0$. En la Tabla 3.1, la última fila corresponde a los tiempo de computo tomando en cuenta 3 modos de funcionamientos en el proceso. Se observa que para un horizonte $N = 10$ no es posible ejecutar una iteración del problema, con los recursos computacionales utilizados (agotando la memoria disponible). Esto último evidencia lo comentado en la sección 3.3.4, a mayor número de modos y mayor el horizonte de predicción, mayor será la carga computacional.

También se han ejecutado varias simulaciones con el fin de mostrar las diferencias entre el control predictivo con saltos markovianos propuesto con horizonte $N = 4$, y los controladores óptimos clásicos, así como también, un controlador predictivo mín-máx.

Las diferentes simulaciones se mencionan a continuación:

- (Sim. 1) el sistema siempre se encuentra en modo 1,
- (Sim. 2) siempre en modo 2,
- (Sims. 3-6) simulación con diferentes patrones de fallos.

En las simulaciones 1-4 se asume que el estado inicial pertenece al conjunto terminal, i.e., $x_0 = [5, 5] \in \mathbb{T}_f$. El controlador predictivo resultante para este caso, evidentemente coincidirá con el controlador LQR clásico, denotado como $DLQR1$ (diseño para el sistema 1) y $DLQR2$ para cuando el radiador 1 falla (modo 2).

En la Tabla 3.2 página 73, se presentan los resultados de las diferentes simulaciones, que a continuación se detallan y discuten. La primera columna indica distintas estrategias de control comparadas, la segunda el índice de coste sumado durante 50 iteraciones (si hay error de posición, la suma infinita no se puede evaluar), la tercera columna indica el coste provocado por el error de posición estacionario, y la última columna indica las temperaturas de equilibrio en las que el sistema se mantiene.

Simulación 1 [Sim 1]: Puede observarse que el controlador $DLQR1$ es en efecto, el controlador óptimo con mejor rendimiento cuando el modo 1 siempre está activo, como intuitivamente se esperaba. Dicho controlador conduce al estado del sistema al punto de operación sin error de posición en estado estacionario (columna 4).

Por otro lado, el controlador $DLQR2$ alcanza peor rendimiento que el CPM propuesto (ver coste en columna 2) y el controlador mín-máx. Nótese que, am-

Cuadro 3.2: Comparación de rendimiento del control terminal para los dos modos de operación, Modo 1: Radiador Q_1 y Q_2 funcionan correctamente. Modo 2: Radiador Q_1 falla.

[Sim 1] Modo Normal (1)	$\sum_{t=1}^{50} L_t$	L_∞	x_∞
DLQR 1	575.8903	$1e^{-9}$	[0,0]
DLQR 2	735.5770	1.0191	[0.0783,0.4512]
CPM	607.3083	0.3852	[0.2450,0.0776]
Mín-max	679.4028	0.9223	[0.7341,0.0793]
[Sim 2] Modo de Fallo (2)			
DLQR 1	$2,8640e^3$	91.4719	[-5.0228,-0.3657]
DLQR 2	$2,1258e^3$	75.1312	[-4.9297,0.1708]
CPM	$2,4155e^3$	75.2947	[-4.9593,-0.0013]
Mín-máx	$2,5413e^3$	76.3245	[-4.8836,0.1289]

Nota: Las columnas de la tabla representa: (a) Controlador diseñado, (b) coste acumulado para 50 muestras, (c) coste final en estado estacionario y (d) estado final.

Los controladores presentan error de posición con respecto al punto de operación, esto se debe básicamente a: cuando el sistema está regulado por la ley de control proveniente del *DLQR2*, es imposible alcanzar error nulo con solo un radiador en funcionamiento, mientras que CPM toma en cuenta la posibilidad de fallo del radiador en instantes futuros, por lo tanto, sobrecalienta ambas habitaciones con el fin de anticiparse y compensar dichos fallos. Si embargo, el controlador CPM propuesto consigue un coste acumulado menor que el controlador mín-máx, el cual intenta minimizar el *peor caso* del índice de coste sobre todas las posibles secuencias de modos, es decir, minimiza el máximo valor la función de coste resultante de una secuencia de modos en particular. La implementación del controlador mín-máx se ha realizado según el problema de optimización basado en LMI que aparece en (Patrinos *et al.* 2014, Corolario 28).

Simulación 2: En este caso el modo 2 siempre está activo, y es obviamente el controlador *DLQR2* quien alcanza un rendimiento óptimo, dado que se ha optimizado para, específicamente, esta situación. Sin embargo, puede verse que existe un error de posición significativo en los *tres* controladores, esto se debe a que con un solo radiador es imposible seguir simultáneamente referencias de temperaturas en ambas habitaciones.

Cuadro 3.3: Resultados de simulación con conmutación de modos.

Sim. 3: Aleatorios	$\sum_1^{300} L$	$E(x_1)$	$E(x_2)$
DLQR 1	25,7632	-2,9878	0,2138
DLQR 2	25,2325	-3,0012	0,3427
DLQR-Conmutado	22,4275	-2,9610	0,2433
CPM	21,5679	-2,8002	0,1593

Simulación 3: En esta simulación, el modo del sistema sí conmuta de acuerdo a la matriz de transición de Markov \mathbb{P} . Para completar la comparación, se considerará un controlador adicional, denominado *DLQR-conmutado*, dicho controlador cambia entre *DLQR1* y *DLQR2* dependiendo del modo medido en cada instante. Es importante señalar que, evidentemente, no hay garantías de estabilidad para los controladores *DLQR1*, *DLQR2* o *DLQR-conmutado* bajo la condición de conmutación arbitraria del modo. Se omite la comparación con el controlador mín-máx, debido al tiempo necesitado para obtener la solución para 300 muestras.

Ejecutando la simulación para 300 muestras, el controlador que consigue el coste medio por muestra ($J/300$) más bajo corresponde al CPM propuesto (Tabla 3.3). En esta tabla también se representan los valores medios de las temperaturas controladas, observándose que el CPM alcanza valores más cercanos a la consignas (menor error).

La evolución temporal de las temperaturas y las acciones de control durante un segmento de la simulación aparecen en la Figura 3-3. Se observa cómo las temperaturas abandonan el punto de funcionamiento cuando ocurre un fallo y regresan a él (CPM ligeramente por encima) al recuperarse el modo normal.

Simulaciones adicionales: Otras simulaciones, descritas abajo, se han ejecutado con el fin de mostrar la correcta implementación del controlador CPM propuesto, para diferentes situaciones. Las simulaciones tienen una longitud de 50 muestras. En la Figura 3-4 se representan las trayectorias de los estados en el plano de fase (x_1, x_2) de tres simulaciones desde condiciones iniciales diferentes:

- La simulación etiquetada “**Sim 4**”(Verde) se realizó seleccionando un estado inicial dentro del conjunto terminal, en modo de fallo. Dicha situación se mantiene durante 9 muestras, luego el sistema conmuta a modo 1 el resto de la simulación, observándose que, tras recuperarse del fallo,

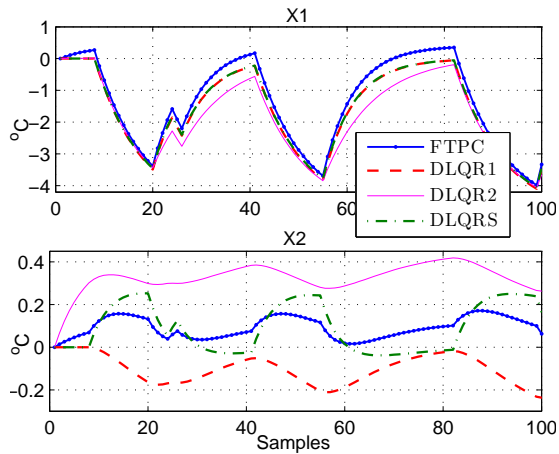


Figura 3-3: Evolución temporal (Simulación 3):

- Gráfica superior: Evolución de Temperatura habitación 1.
- Gráfica inferior: Evolución de Temperatura habitación 2.

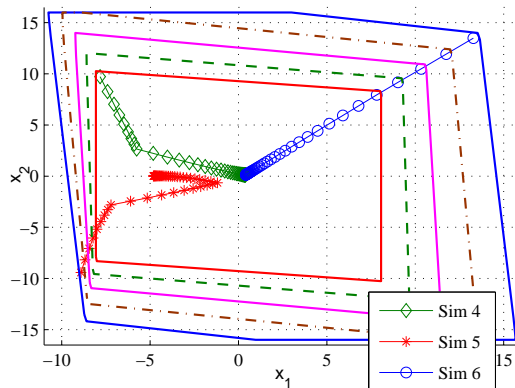


Figura 3-4: Trayectorias para diferentes estados iniciales (Simulaciones 4-6).

ambas temperaturas alcanzan el origen.

- La simulación “**Sim 5**”(roja) fue realizada escogiendo el estado inicial fuera del conjunto terminal y en modo de fallo, recuperando la condición normal de funcionamiento en $k = 8$ y volviendo al fallo en $k = 18$. Puede verse que el sistema finaliza con un error (offset) tal como se había discutido anteriormente.
- Finalmente la simulación “**Sim 6**”(azul) comienza con el estado inicial fuera del conjunto terminal y en todo instante con modo de funcionamiento normal. Nótese que la trayectoria de los estados del sistema alcanzan, sin problemas, el origen.

3.5. Control Predictivo Basados en Escenarios

La siguiente sección está basado en el trabajo (Bernardini and Bemporad 2009) en su versión congreso, extendida posteriormente a su versión revista en (Bernardini and Bemporad 2012).

Considérese un sistema lineal con incertidumbres en tiempo discreto:

$$x_{k+1} = A(w_k)x_k + B(w_k)u_k, \quad (3.29)$$

donde $w_k \in \mathbb{W}$ es una variable aleatoria y $\mathbb{W} : \{w^1, \dots, w^s\}$. Enumerando todas las s posibles realizaciones de w_k , el sistema (3.29) puede ser escrito como:

$$x_{k+1} = \begin{cases} A_1x_k + B_1u_k & \text{si } w_k = w_1 \\ A_2x_k + B_2u_k & \text{si } w_k = w_2 \\ \vdots & \vdots \\ A_sx_k + B_su_k & \text{si } w_k = w_s \end{cases} \quad (3.30)$$

Para modelar la evolución de la variable w_k , considere el vector de probabilidad $p(k) = [p_1(k), p_2(k), \dots, p_s(k)]^T$ que define la probabilidad de cada realización de w_k en el instante k , *i.e.*:

$$p_j(k) = Pr[w_k = w_j], \quad j = 1, 2, \dots, s \quad (3.31)$$

con $\sum_{j=1}^s p_j(k) = 1$, $k \in \mathbb{N}$.

Suposición 3.1. Se asume que en el instante k , el vector de probabilidades $p(k)$ es conocido. Además, $p(k) \in \mathcal{P}$, $\forall k \geq 0$, donde \mathcal{P} es un conjunto convexo, $\mathcal{P} \subseteq \mathcal{D}$,

siendo:

$$\mathcal{D} := \left\{ p \mid 0 \leq p_i \leq 1, i = 1, \dots, s, \sum_{i=1}^s p_i(k) = 1 \right\}$$

Se denota con v^1, v^2, \dots, v^m a los m vértices del politopo \mathcal{P} con $v^j = [v_1^j, v_2^j, \dots, v_s^j]$, $j = 1, 2, \dots, m$.

3.5.1. Diseño MPC Estocástico

El objetivo de esta sección es diseñar un esquema MPC estocástico que resuelve el problema de regular el estado x del sistema (3.29) usando la información disponible de la variable w_k . La idea es asegurar estabilidad media cuadrática (véase Definición 2.3).

Caso sin restricciones

El objetivo es calcular una función de Lyapunov con su asociada ley de control que pruebe estabilidad media cuadrática para el sistema (3.29) en ausencia de restricciones. Para ello, considere lo siguiente:

$$\mathbb{E}[\mathcal{V}(x_{k+1|k})] - \mathcal{V}(x_{k|k}) \leq -x_{k|k}^T L x_{k|k} \quad (3.32)$$

donde $\mathcal{V}(x)$ viene dada por la ecuación (1.44), *i.e.*, $\mathcal{V}(x) := x^T P x$. Limitándose al caso de funciones de Lyapunov cuadráticas la ecuación (3.32) corresponde a la ecuación (2.23). Dada un vector de probabilidades $p(k)$, se tiene:

$$\mathbb{E}[\mathcal{V}(x_{k+1|k})] = \sum_{j=1}^s p_j(k) x_{k+1|k,j}^T P x_{k+1|k,j}, \quad (3.33)$$

donde $x_{k+1|k,j} = A_j x_k + B_j u_k$, $j = 1, 2, \dots, s$.

Definiendo $u_k := K x_k$, $P = \bar{Q}^{-1}$, $K = Y \bar{Q}^{-1}$, $L = W^{-1}$ con $\bar{Q} = \bar{Q}^T \succ 0$, $W = W^T \succ 0$ y utilizando el complemento de Schur, el cumplimiento de la desigualdad (3.32) viene dado por el cumplimiento de la siguiente LMI (Bernardini and Bemporad 2009):

$$\left[\begin{array}{ccccccc} \bar{Q} & \bar{Q} & \sqrt{p_1}(A_1\bar{Q} + B_1Y)^T & \cdots & \sqrt{p_s}(A_s\bar{Q} + B_sY)^T & & \\ \bar{Q} & W & 0 & \cdots & 0 & & \\ \sqrt{p_1}(A_1\bar{Q} + B_1Y) & 0 & \bar{Q} & & & & \\ \vdots & \vdots & & \ddots & & & \\ \sqrt{p_s}(A_s\bar{Q} + B_sY) & 0 & & & & & \bar{Q} \end{array} \right] \succeq 0, \quad (3.34)$$

Para uso de una notación más simple, se refiere a la matriz (3.34) como $\mathcal{L}(p(k))$, haciendo énfasis en la dependencia de la solución de la LMI¹⁴ del vector de probabilidades $p(k)$. Si $\mathcal{L}(p(k)) \succeq 0$ entonces existe una solución para todo $p(k)$, entonces, la ley de control por realimentación $u_k = Y\bar{Q}^{-1}x_k$ garantiza convergencia asintótica (en media cuadrática) del sistema en lazo cerrado (Bernardini and Bemporad 2012, Lema 2).

Obtención de la Función de Lyapunov Fuera de Linea. Considere un proceso modelado por el sistema 3.29 donde $w(k) \in \mathbb{W}$ está descrita por un proceso aleatorio con una distribución de probabilidad $p(k) \in \mathcal{P}$, $\forall k \geq 0$ la cual cumple con la Suposición 3.1. Sea v^1, v^2, \dots, v^m , los m vértices del conjunto \mathcal{P} y sea $\bar{Q} \in \mathbb{R}^{n_x \times n_x}$, $W \in \mathbb{R}^{n_x \times n_x}$, $Y \in \mathbb{R}^{n_x \times n_x}$ una solución del siguiente problema de optimización semidefinida:

$$\min_{\bar{Q}, W, Y} \text{traza}(W) \quad (3.35a)$$

$$\text{s.t } \text{traza}(\bar{Q}) = 1 \quad (3.35b)$$

$$W - W_0 \succeq 0 \quad (3.35c)$$

$$\text{diag}(\mathcal{L}(v^1), \dots, \mathcal{L}(v^m)) \succeq 0 \quad (3.35d)$$

entonces, el sistema (3.29) con ley de control $u_k = Y\bar{Q}^{-1}x_k$ es exponencialmente estable en media cuadrática (Bernardini and Bemporad 2012, Lema 3).

Diseño de un Árbol de Optimización. Se asume que está disponible un modelo exacto de la evolución del vector de probabilidad $p(k)$, por ejemplo, una cadena de Markov. A continuación se propone un esquema diseño de árbol basado en el enfoque de máxima probabilidad, en el que el árbol de optimi-

¹⁴Nótese que la desigualdad (3.34) es un caso particular de aquella presente en el Lema 2.1, dado que en (3.34) no se considera que el vector de probabilidades $p(k)$ tenga incertidumbre, es decir, es conocido para todo instante k . Además, como en (Bernardini and Bemporad 2009) se busca meramente estabilidad, entonces W se considera variable de decisión. En el lema 2.1, W^{-1} es fija, cuadrática dependiente de las matrices del índice de coste a optimizar.

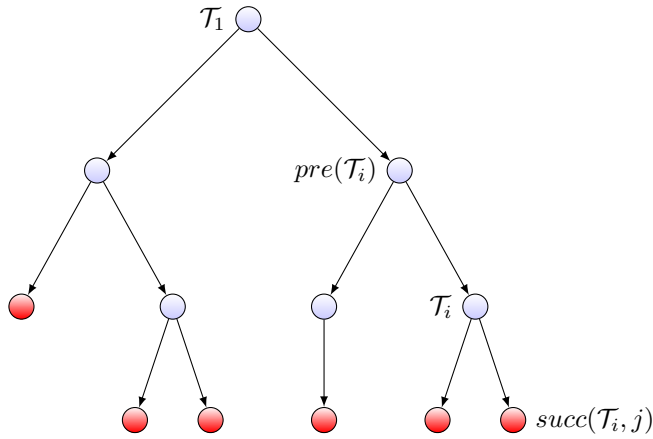


Figura 3-5: Ejemplo de un árbol de optimización. Los nodos en rojo representan el conjunto de nodos terminales S .

zación se actualiza en cada periodo de muestreo utilizando la información disponible del estado x_k y la variable w_k . Para ello, se listan diferentes definiciones necesarias para la construcción de un árbol como el mostrado en la Figura 2 de (Bernardini and Bemporad 2012).

- $\mathcal{T} := \{\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_n\}$: Conjunto de nodos del árbol. Los nodos serán indexados progresivamente al árbol.
- $x_{\mathcal{N}}$: Estado asociado al nodo \mathcal{N} .
- $u_{\mathcal{N}}$: Entrada de control asociada al nodo \mathcal{N} .
- $\pi_{\mathcal{N}}$: Probabilidad de alcanzar el nodo \mathcal{N} desde \mathcal{T}_1 .
- $pre(\mathcal{N})$: Predecesor del nodo \mathcal{N} .
- $succ(\mathcal{N})$: Sucesor del nodo \mathcal{N} generado con el modo $j \in \{1, 2, \dots, s\}$.
- $\delta(\mathcal{N})$: Modo que conduce al nodo \mathcal{N} .
- $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_c\}$: Conjuntos de nodos candidatos a ser añadidos al árbol y se definen como $\mathcal{C} := \{\mathcal{N} \notin \mathcal{T} \mid \exists(i, j) : \mathcal{N} = succ(\mathcal{T}_i, j)\}$.
- $\mathcal{S} \subset \mathcal{T}$: Conjuntos de nodos terminales definidos como $\mathcal{S} := \{\mathcal{T}_i \mid succ(\mathcal{T}_i, j) \notin \mathcal{T}, i = 1, 2, \dots, n, j = 1, 2, \dots, s\}$.

Algoritmo para el diseño de árboles. En (Bernardini and Bemporad 2012, Sección 3, Alg 1.) se presenta un algoritmo para la obtención de un árbol de optimización, el cual consiste en: cada nodo del árbol (Figura 3-5) representa un

estado predicho que se tomará en cuenta en el problema de optimización. Iniciando desde el nodo raíz \mathcal{T}_1 , que está definido con el valor actual $p(k)$, se evalúa un lista de posible nodos candidatos \mathcal{C} considerando todas las s posibles dinámicas en (3.30) con sus respectivas probabilidades $p_1(k), p_2(k), \dots, p_s(k)$. El candidato con máxima probabilidad, denotado como \mathcal{C}_{i^*} es añadido al árbol. La lista de candidato es actualizada añadiendo todos los nodos sucesores del último nodo añadido, *i.e.*, $\text{succ}(\mathcal{C}_{i^*}, j)$ para toda $j \in \{1, 2, \dots, s\}$. Este procedimiento es repetido hasta alcanzar un numero de nodos máximos deseados. Este procedimiento conduce a una estructura de árbol flexible, es decir, puede existir trayectorias que en general tienen diferentes longitudes, como se muestra en la Figura 3-5.

Nótese que la aplicación de este algoritmo favorece la construcción de un árbol de optimización dando prioridad a realizaciones de la variable w_k con mayor probabilidades, permitiendo que éstas sean de mayor longitud que aquellas realizaciones con menor probabilidades. De ahí la causa de que puedan existir secuencias de diferentes longitudes en el árbol resultante, véase la Figura 3-5.

Formulación del Problema de Control: Para facilitar la notación, $x_i, u_i, \pi_i, \delta(i), \text{pre}(i)$ serán usadas para denotar $x_{\mathcal{T}_i}, u_{\mathcal{T}_i}, \pi_{\mathcal{T}_i}, \delta(\mathcal{T}_i), \text{pre}(\mathcal{T}_i)$, respectivamente. Sea $x_{k|k}$ el estado medido en el instante k , $p(k)$ el vector de probabilidades del la variable w_k , el problema MPC estocástico sin restricciones puede ser formulado como:

$$\min_u \sum_{i \in \mathcal{T} \setminus (\{\mathcal{T}_1\} \cup \mathcal{S})} \pi_i x_i^T Q x_i + \sum_{j \in \mathcal{T} \setminus \mathcal{S}} \pi_j u_j^T R u_j + \sum_{i \in \mathcal{S}} \pi_i x_i^T Q_S x_i \quad (3.36a)$$

s.t

$$x_1 = x_{k|k} \quad (3.36b)$$

$$x_i = A_{\delta(i)} x_{\text{pre}(i)} + B_{\delta(i)} u_{\text{pre}(i)}, \quad \forall i \in \mathcal{T} \setminus \{\mathcal{T}_1\} \quad (3.36c)$$

$$\sum_{i=1}^s p_i(k) (A_i x_1 + B_i u_1)^T P (A_i x_1 + B_i u_1) \leq x_1^T (P - L) x_1 \quad (3.36d)$$

donde las matrices Q, R, Q_S son matrices de penalización de dimensiones apropiadas.

Teorema 3.2. (Bernardini and Bemporad 2009) *Se asume que se cumple el Supuesto 3.1 y que existe una solución factible del problema 3.35a. Entonces, el estado x del sistema (3.29) controlado por la ley de control óptima calculada a través del problema 3.36a, es exponencialmente estable en media cuadrática.*

Caso con Restricciones. Considérese que existen restricciones en termino de componente, tanto en el estado como en la entrada del proceso de la forma¹⁵:

$$x \in \mathbf{X} := \{x : |x_i| \leq x_i^+, i = 1, 2, \dots, n_x\} \quad (3.37a)$$

$$u \in \mathbf{U} := \{u : |u_i| \leq u_i^+, i = 1, 2, \dots, n_u\} \quad (3.37b)$$

Se desea obtener (fuera de línea) una función de Lyapunov y una ley de control que garantice el cumplimiento de las restricciones y que cumpla con la convergencia robusta del sistema al origen, *i.e.*

$$\mathcal{V}(x_{k+1|k}) - \mathcal{V}(x_{k|k}) \leq -x_{k|k}^T L x_{k|k}, \forall k \in \mathbb{N} \quad (3.38)$$

Se puede expresar la desigualdad (3.38) de la forma (3.32) en el problema de control en línea, usando la información disponible en $p(k)$. Aplicando el complemento de Schur, la condición (3.38) puede ser expresada en forma de LMI como:

$$\begin{bmatrix} \bar{Q} & (A_j \bar{Q} + B_j Y)^T & (L^{1/2} \bar{Q})^T \\ (A_j \bar{Q} + B_j Y) & \bar{Q} & 0 \\ (L^{1/2} \bar{Q}) & 0 & \gamma I \end{bmatrix} \succeq 0, j = 1, \dots, s \quad (3.39)$$

donde $P = \gamma \bar{Q}^{-1}$, $K = YQ^{-1}$ y $L \succ 0$. Con el fin de tomar en consideración las restricciones, se define el siguiente elipsoide invariante:

$$\mathcal{E} := \{x | x^T \bar{Q}^{-1} x \leq 1\} = \{x | x^T P x \leq \gamma\}$$

\mathcal{E} representa un elipsoide invariante para el sistema en lazo cerrado (3.29) controlador por la ley de control $u_k = Kx_k$, es decir, si $x_k \in \mathcal{E}$ implica que $x_{k+t} \in \mathcal{E}, \forall t \in \mathbb{N}$. Entonces, dado un estado inicial $x_0 \in \mathcal{E}$, esto puede expresarse como 1.49, *i.e.*:

$$\begin{bmatrix} 1 & x_0^T \\ x_0 & \bar{Q} \end{bmatrix} \succeq 0, \quad (3.40)$$

Una condición suficiente para el cumplimiento de 3.37 está dada por la LMI

$$\begin{bmatrix} \bar{Q} & (I^i (A_j \bar{Q} + B_j Y))^T \\ (I^i (A_j \bar{Q} + B_j Y)) & (x_i^+)^2 \end{bmatrix} \succeq 0, \quad (3.41)$$

¹⁵Otros tipos de restricciones, tales como politópicas (1.17) o elipsoidales pueden ser manejadas de forma similar, véase (Kothare *et al.* 1996).

para $i = 1, 2, \dots, n_x$, $j = 1, 2, \dots, s$, y

$$\begin{bmatrix} X & Y \\ Y^T & \bar{Q} \end{bmatrix} \succeq 0, \quad X_{ll} \leq (u_l^+)^2, \quad l = 1, 2, \dots, n_u \quad (3.42)$$

donde I^i es la i -ésima final de una matriz identidad de dimensión $n_x \times n_x$.

Por lo tanto, el problema de calcular una función de Lyapunov cuadrática y una ley de control que cumpla con las restricciones (3.37) y estabilidad robusta 3.38 puede ser establecido como:

$$\min_{\bar{Q}, X, Y, \gamma} -\log \det(\bar{Q}) \quad (3.43a)$$

$$\text{s.t.} \quad (3.39), (3.40), (3.41), (3.42) \quad (3.43b)$$

Formulación del Problema de Control: El problema MPC estocástico sujeto a restricciones puede ser formulado como:

$$\min_u \sum_{i \in \mathcal{T} \setminus (\{\mathcal{T}_1\} \cup \mathcal{S})} \pi_i x_i^T Q x_i + \sum_{i \in \mathcal{T} \setminus \mathcal{S}} \pi_i u_i^T R u_i + \sum_{i \in \mathcal{S}} \pi_i x_i^T Q_S x_i \quad (3.44a)$$

s.t

$$x_1 = x_{k|k} \quad (3.44b)$$

$$x_i = A_{\delta(i)} x_{pre(i)} + B_{\delta(i)} u_{pre(i)}, \quad \forall i \in \mathcal{T} \setminus \{\mathcal{T}_1\} \quad (3.44c)$$

$$x_i \in \mathbf{X}, \quad \forall i \in \mathcal{T} \setminus \{\mathcal{T}_1\} \quad (3.44d)$$

$$u_j \in \mathbf{U}, \quad \forall j \in \mathcal{T} \setminus \mathcal{S} \quad (3.44e)$$

$$(A_i x_1 + B_i u_1) \in \mathbf{X}, \quad \forall i \mid succ(\mathcal{T}_1, i) \notin \mathcal{T} \quad (3.44f)$$

$$(A_h x_1 + B_h u_1)^T P (A_h x_1 + B_h u_1) \leq \gamma, \quad \forall h \mid p_h(k) > 0 \quad (3.44g)$$

$$\sum_{i=1}^s p_i(k) (A_i x_1 + B_i u_1)^T P (A_i x_1 + B_i u_1) \leq x_1^T (P - L) x_1 \quad (3.44h)$$

Nótese que el sistema en lazo cerrado asegura estabilidad en media cuadrática mediante la restricción de Lyapunov decreciente (3.44h). Además, la factibilidad recursiva del problema de optimización es garantizada dado que se obliga que el estado siguiente permanezca dentro de un conjunto elipsoidal \mathcal{E} , véase la restricción (3.44g).

3.6. Conclusiones

Este capítulo se ha presentado una configuración de un control predictivo para sistemas lineales con saltos markovianos cuya transiciones entre diferentes modos son modeladas por una cadena de Markov, dicho controlador es estable en media cuadrática. Se asume que el conocimiento de bien el modo actual o bien el anterior está disponible para el controlador, en el momento de aplicar las actuaciones al proceso. Los resultados en la literatura para sistema lineales de Markov sin restricciones se han extendido a leyes de control predictivo de horizonte deslizante para sistemas con restricciones, obteniendo los conjuntos terminal y factible necesarios para asegurar estabilidad en media en esta metodología. Varios ejemplos muestran que el rendimiento simulado del enfoque propuesto en este trabajo, corresponden a las expectativas teóricas presentadas.

Finalmente se presenta el diseño de MPC estocásticos basado en escenario. Se plantean dos casos diferentes, en el primero se considera el problema de control sin restricciones, mientras que en el segundo el problema de optimización se encuentra sujeto a restricciones de estados y entradas. La ley de control resultando garantiza estabilidad y factibilidad mediante funciones de Lyapunov cuadráticas.

Part II
Contributions

Capítulo 4

Multiple-horizon Predictive Control for Markov/switched Linear Systems

Nota sobre el capítulo: El presente capítulo plantea un primer enfoque de controladores predictivos para sistemas con saltos markovianos basado en escenario, considerando la posibilidad de tener secuencias de diferentes longitudes. El contenido de este capítulo aparece en el siguiente trabajo:

- M. Hernández-Mejías, A. Sala, A. Querol, C. Ariño. *Multiple-horizon predictive control for Markov/switched linear systems.* 5th IFAC Conference on Nonlinear Model Predictive Control (NMPC), 2015. Pp. 230-235.
DOI:10.1016/j.ifacol.2015.11.288.

Abstract: This chapter proposes a model predictive control for Markov-jump or switched linear systems (switching between a finite set of modes) when simultaneously considering prediction horizons (and realisations of the mode variable) of different length, in order to balance accuracy versus computational cost. Feasibility, stability and receding horizon issues are considered.

4.1. Introduction

Most systems are subject to state and control constraints. Set invariance theory is a fundamental tool in order to obtain robust control invariant sets. The uncertain polytopic linear case is addressed in (Blanchini 1994, Kerrigan 2000). Feasible sets in MPC are also strongly related to such developments (Kerrigan 2000).

This chapter addresses MPC for Markov-jump/switched linear systems in both average or minimax cost settings, assuming that the current mode of operation is known. Unconstrained mode-dependent state-feedback terminal controllers for average cost optimisation is presented in (Costa *et al.* 2005) via coupled Riccati equations; a minimax set-up with well-known ‘guaranteed cost’ LMIs appears in (Zhang and Boukas 2009); (Lu *et al.* 2013) combines both approaches for uncertain MJLS (see Section 2.5 for details.).

The main problem of MPC set-ups for Markov/switched linear systems is the exponential increase of the number of possible outcomes needed to craft the complete prediction models. The number of decision variables also grows exponentially with the prediction horizon.

The objective of this chapter is to show how predictive control can be formulated when *simultaneously* considering prediction vectors of different lengths. Conditions for guaranteeing a finite cost bound for all cases and receding-horizon stability are proposed. Preliminary ideas on choosing multiple-length sequences are given, in order to achieve a trade-off between conservatism and computational cost.

4.2. Preliminaries and Problem statement

Consider is the following linear time-varying system:

$$x_{k+1} = A_{\theta_k} x_k + B_{\theta_k} u_k, \quad k \geq 0 \quad (4.1)$$

where $x_k \in \mathbb{R}^n$ represents the state vector, and $u_k \in \mathbb{R}^m$ the control actions. $\{\theta_k, k = 0, 1, \dots\} \in \mathcal{M}$, $\mathcal{M} = \{1, 2, \dots, M\}$ is the so-called mode variable, which follows a discrete-time Markov chain with a transition probability matrix given by $\mathbb{P} = [\pi_{ji}]$, i.e., $Pr\{\theta_{k+1} = j | \theta_k = i\} = \pi_{ij}$, where $\pi_{ji} \geq 0$, $\forall i, j \in \mathcal{M}$ and $\sum_{j=1}^M \pi_{ji} = 1$. System (4.1) switches between M different operation modes, i.e.,

$$A_{\theta_k} \in \bar{\mathcal{A}} = \{A_1, \dots, A_M\}, \quad B_{\theta_k} \in \bar{\mathcal{B}} = \{B_1, \dots, B_M\} \quad (4.2)$$

System (4.1) is subject to state and control constraints, i.e., $u_k \in \mathbb{U}$ and $x_k \in \mathbb{X}$, where $\mathbb{U} \subset \mathbb{R}^m$ is compact, $\mathbb{X} \subset \mathbb{R}^n$ is closed, and each set contains the origin in its interior.

It is assumed that there exists a diagnoser such that at time k , mode θ_k is known, as well as the state x_k and the Markov chain probability matrix \mathbb{P} . Apart from parameter jumps, the mode matrices are assumed time-invariant ($\bar{\mathcal{A}}$ and $\bar{\mathcal{B}}$ are constant) and π_{ji} will not change with time.

Chapter 3 assumes that the transition probabilities matrix of a Markov chain is complete, i.e., $\pi_{ji} \neq 0, \forall j, i \in \mathcal{M}$. However, in many cases there is no transition between some operation modes. In such cases, we have:

Admissible Sequences.

\mathcal{S}_i is defined as the set of all modes $j \in \mathcal{M}$ accessible from a mode $i \in \mathcal{M}$ in one time step, i.e., $\mathcal{S}_i := \{j \in \mathcal{M} | \pi_{ji} > 0\}$. An admissible switching sequence of length N , $\theta^l = \{\theta_0^l, \theta_1^l, \dots, \theta_{N-1}^l\}$ for $(x_{k+1} = A_{\theta_k} x_k + B_{\theta_k} u_k, \theta_k \in \mathcal{M},)$ is a switching path for which $\theta_{k+1}^l \in \mathcal{S}_{\theta_k}$ for any $0 \leq k \leq N-1$. The set of all admissible mode sequences of length N is denoted as \mathcal{G}^N , and \mathcal{G} is the set of all admissible finite-length sequences. Notation $len(\theta^l)$ represents the length of the particular sequence θ^l .

Let us denote $\Theta \subset \mathcal{G}$ as a set of sequences $\theta^l, l \in \mathcal{L}$, i.e., $\Theta := \{\theta^1, \theta^2, \dots\}$. Furthermore, when two arbitrary sequences are given, i.e., η and ξ , the notation $\eta\xi$ will represent the concatenation of these sequences, so by definition $len(\eta\xi) := len(\eta) + len(\xi)$.

4.2.1. Predictive Control for Markov/switched Systems.

In order to set-up predictive control computations, terminal controllers (mode-dependent terminal state feedback K^i and terminal weights P^i) and finite-horizon cost indices must be crafted. Let us separately consider the ave-

rage cost set-up (MJLS) and the minimax case (arbitrary switching).

MJLS.

Terminal controllers using unconstrained average-cost minimisation for MJLS are well developed, see Section 2.3, using coupled Riccati equations. The average-cost solutions are exact in the terminal set (maximal invariant set in which the terminal control does not saturate).

Switched case.

Widely used minimax state-feedback controllers (unconstrained case) appear in (Scokaert and Mayne 1998) for the arbitrary switching case; (Lazar *et al.* 2006, Lemma IV) poses them in LMI form (used here in later examples). The solutions in the cited literature are only upper bounds on the true infinite-time quadratic cost.

Once terminal controllers are obtained, predictive set-ups are crafted by considering: (a) a finite prediction horizon N , (b) a prediction model, this model consist of each possible realisations of the mode variable (Section 3.3.1); (c) an average/minimax finite-time quadratic cost index.

If the terminal cost over-bounds the infinite-time cost, such schemes can be guaranteed stable or stochastically stable (Lu and Arkun 2000, Patrinos *et al.* 2014) in receding-horizon implementation.

Robust invariant sets (terminal sets) and controllable sets (guaranteed feasible for the predictive control set-up) can be obtained using the ideas and software in (Blanchini 1994, Kerrigan 2000, Patrinos *et al.* 2014). Modifications when some switching sequences are not possible will be later proposed.

4.2.2. Problem Statement

The main problem of minimax/average cost optimisation is the exponential increase of the number of considered sequences and decision variables as the horizon N increases. For accurate cost estimations, such horizon must be large: if we solve the minimax predictive control for horizon $N_1 > N$, then the optimal cost for the longer horizon is lower or equal, because of the terminal over-bounding; this is also the case in average cost optimisation if the initial state is outside of the terminal set.

The objective of this chapter is to discuss preliminary ideas on how using sets of sequences Θ with fewer elements than \mathcal{G}^{N_1} (being N_1 a large integer

which computational resources cannot handle) but with more elements than those in \mathcal{G}^N (for small-enough N) can be used to get a reasonable trade-off between computational complexity and proven cost bound, while maintaining, in any case, stability guarantees and a clear characterisation of the sets on which the proposed problem is feasible.

As all the sequences in the considered set will not have the same length, the approach will be coined as *multiple-horizon* predictive control.

4.3. Sequence-set Predictive Control

4.3.1. Prediction Model

Consider a set $\Theta \subset \mathcal{G}$ of variable-length sequences. Consider the model (2.3), given a particular realization $\theta^l \in \Theta$, $l \in \mathcal{L}$, for known initial state x_k and a sequence of future actions $\mathcal{U}^l = \{u_k^{[l]}, u_{k+1|k}^{[l]}, u_{k+2|k}^{[l]}, \dots\}$, the future states predicted for it will be denoted as $x_{k+i|k}^{[l]}$, for any $0 \leq i \leq \text{len}(\theta^l)$. It can be expressed as:

$$x_{k+i|k}^{[l]} = \prod_{j=k}^{k+i-1} A_j^{[l]} x_k + \sum_{d=k}^{k+i-1} \left(\prod_{j=d+1}^{k+i-1} A_j^{[l]} \right) B_d^{[l]} u_{d|k}^{[l]} \quad (4.3)$$

Given two sequences θ^{l_1} and θ^{l_2} with different lengths, say $N_1 < N_2$ being $N_1 = \text{len}(\theta^{l_1})$ and $N_2 = \text{len}(\theta^{l_2})$, if the first one is a subsequence of the second one, i.e., there exists a sequence ξ such that $\theta^{l_1} \xi = \theta^{l_2}$, some refinements to the above causality issues must be made: as receding-horizon stability assumes that the terminal controller is used forever after the terminal set is reached, that means that the longer sequence should assume such terminal control law (no new decision variables should be added, i.e., forcedly $u_{k+h|k}^{[l_2]} = K^i x_{k+h|k}$, with $i = \theta_h^{l_2}$ for $h \geq N_1$).

Remark 4.1. In standard MPC set-ups, considering all admissible sequences of length N , the number of sequences $\text{card}(\mathcal{G}^N)$ can be determined via the so-called incidence matrix \mathcal{I} , defined as the one whose element (i, j) is equal to 1 if $\pi_{ij} \neq 0$, and equal to zero otherwise. Indeed, it can be proved that the number of sequences is the sum of the elements of the N -Th power of the incidence matrix \mathcal{I}^N : the largest eigenvalue of \mathcal{I} governs the exponential increase of the number of admissible sequences for large N .

4.3.2. Sequence-set-Dependent Cost Index for Predictive Control

Given a particular sequence $\theta^l \in \Theta$, the **sequence-dependent** quadratic cost index will be defined as:

$$J^l(x_0, \mathcal{U}^l, \theta^l) := \|x_{k+\text{len}(\theta^l)-1|k}^{[l]}\|_{P_{\theta^l}^l} + \sum_{n=0}^{\text{len}(\theta^l)-2} \|x_{k+n|k}^{[l]}\|_{Q_{\theta_n^l}}^2 + \|u_{k+n|k}^{[l]}\|_{R_{\theta_n^l}}^2 \quad (4.4)$$

being P_i , Q_i and R_i suitable symmetric positive definite weighting matrices of appropriate dimensions, $i \in \mathcal{M}$. In this case, in order to be coincident with standard predictive control concepts, the prediction horizon N is set to $N = \text{len}(\theta^l) - 1$, being the last element of θ^l the mode at the time instant when terminal set is reached.

Minimax cost (arbitrary switching systems).

the Θ -minimax problem may be defined as obtaining the decision variables $\mathcal{U}^l := \{u_d^{[l]}, l \in \mathcal{L}, d = 0, \dots, \text{len}(\theta^l) - 1\}$ minimising:

$$J(\theta_0, \Theta) = \min_{\mathcal{U}^l} \max_{l, \theta_0^l = \theta_0} J^l(x_0, \mathcal{U}^l, \theta^l) \quad (4.5)$$

subject to the standard constraints: intermediate states in validity region \mathbb{X} , control action in \mathbb{U} and terminal state inside a terminal set to be later computed. This problem can be expressed as an LMI problem as follow:

Given the prediction model (4.3), for a particular sequence θ^l , we denote with $\mathcal{X}_{N|k}^l$, the set of future predicted states with information at time k , from time instants k to $k+(N-1)$. Such predictions, as well as the control actions \mathcal{U}^l will be assumed juxtaposed in column vector form, abusing the notation, as in standard MPC set-ups. Then, it is well known that a matrix representation:

$$\mathcal{X}_{N|k}^l = \Xi^l x_k + \Upsilon^l \mathcal{U}^l \quad (4.6)$$

can be set up. Also, a similar expression of the state $x_{k+N|k}^{[l]}$ as a function of x_k and \mathcal{U}^l can be obtained.

The problem (4.5) is equivalent to:

$$\text{minimise } \gamma \quad (4.7)$$

$$\text{subject to } J^l(x_0, \mathcal{U}^l, \boldsymbol{\theta}^l) \leq \gamma, \forall l \in \mathcal{L} \quad (4.8)$$

$$\text{and } x_{k+i|k}^{[l]} \in \mathbb{X}, u_{i|k}^{[l]} \in \mathbb{U}, x_{k+N|k}^{[l]} \in \mathbb{T}^{\theta^l_N} \quad (4.9)$$

being $\mathbb{T}^{\theta^l_N}$ the mode-dependent terminal set, to be defined later. Expressing $J^l(x_0, \mathcal{U}^l, \boldsymbol{\theta}^l)$ as a quadratic form (4.4). This problem can be expressed as an LMI problem, applying Schur complement transforms, setting $N = \text{len}(\boldsymbol{\theta}^l) - 1$:

$$\begin{pmatrix} \gamma & * & * & * \\ \mathcal{X}_{N|k}^l & (\hat{Q}^l)^{-1} & 0 & 0 \\ \mathcal{U}^l & 0 & (\hat{R}^l)^{-1} & 0 \\ x_{k+N|k}^{[l]} & 0 & 0 & (P_{\theta^l_N})^{-1} \end{pmatrix} > 0, \quad \forall l \in \mathcal{L}, \quad (4.10)$$

where:

$$\hat{Q}^l = \text{diag}(Q_{\theta^l_1}, \dots, Q_{\theta^l_{N-1}})$$

and

$$\hat{R}^l = \text{diag}(R_{\theta^l_1}, \dots, R_{\theta^l_{N-1}})$$

plus the well-known expressions of the constraints forcing a point being inside a polytope (Lu and Arkun 2000). Causality constraints (Scokaert and Mayne 1998) should also be enforced by suitably coding of \mathcal{U}^l .

Average cost (MJLS).

As an alternative option when probabilities of transitions are available, the index:

$$J(\theta_0, \boldsymbol{\Theta}) = \min_{\mathcal{U}^l} \mathbb{E}_{\theta_0^l = \theta_0} (J^l(x_0, \mathcal{U}^l, \boldsymbol{\theta}^l)) \quad (4.11)$$

denotes minimising the conditional expectation of the cost (conditioned to knowledge of the initial mode). This cost can be minimised via QP or LMIs:

With suitable computation of the probability of a sequence $\boldsymbol{\theta}^l$ given its first element (current mode is assumed available, transition probabilities are known), using (3.3), the MPC minimising the average cost can be obtained by solving on-line the LMI optimisation problem:

$$\min \sum_{l \in \mathcal{L}} Pr(\boldsymbol{\theta}^l) \gamma^l$$

Subject to:

$$\begin{pmatrix} \gamma^l & * & * & * \\ \mathcal{X}_{N|k}^l & (\hat{Q}^l)^{-1} & 0 & 0 \\ \mathcal{U}^l & 0 & (\hat{R}^l)^{-1} & 0 \\ x_{k+N|k}^{[l]} & 0 & 0 & (P_{\theta_N^l}^{-1}) \end{pmatrix} > 0 \quad \forall l \in \mathcal{L}, \quad (4.12)$$

plus the same polytopic constraints on $\mathcal{X}_{N|k}^l$ and \mathcal{U}^l .

Note: in order for the expectation to be well defined, it will be assumed that the sequences in Θ are such that conditional probabilities add one (see later).

4.4. Terminal and Feasible Sets

The proposed developments require determining some sets in which feasibility of the different problems and stability can be guaranteed. As the mode at the moment of application of the control action is assumed known, the relevant terminal and feasible sets of standard predictive control can be made mode-dependent, as follows.

4.4.1. Mode-dependent Terminal Set.

Consider an arbitrary set \mathcal{C} , let us define the **mode-dependent One-Step set** $\mathcal{Q}_i(\mathcal{C})$ as:

$$\mathcal{Q}_i(\mathcal{C}) = \{x \in \mathbb{R}^n \mid u^i = K^i x \in \mathbb{U}, [A_i - B_i K^i]x \in \mathcal{C}\} \quad (4.13)$$

The **Mode-dependent Terminal Set** \mathbb{T}^i , $i \in \mathcal{M}$, will be obtained using the above definition, the state-feedback gain and applying the following 2:

4.4.2. Mode-dependent Feasible Sets.

Now, let us define the **Mode-dependent One-Step Controllable Set** $\mathbb{Q}_i^{\mathcal{C}}(\mathcal{C})$ as:

$$\mathbb{Q}_i^{\mathcal{C}}(\mathcal{C}) := \{x \in \mathbb{R}^n \mid \exists u^i \in \mathbb{U}, [A_i x + B_i u^i] \in \mathcal{C}\} \quad (4.14)$$

Inspired on N -step sets in conventional MPC (Definition 1.14), let us generalise the concept by defining the **Mode-dependent N-step Controllable Set** \mathbb{K}_N^i as the output of the following recursive algorithm:

Algorithm 2 Mode-dependent terminal sets

```

For all  $i$ , Set  $\mathbb{K}_0^i = \{x : K^i x \in \mathbb{U}\}$ ;
Let  $v = 1$ ;
For all  $i \in \mathcal{M}$ , compute  $\mathbb{K}_v^i = \mathcal{Q}_i(\cap_{j \in \mathcal{S}_i} \mathbb{K}_{v-1}^j) \cap \mathbb{X}$ ;
si  $\mathbb{K}_v^i = \mathbb{K}_{v-1}^i, \forall i$  entonces
    let success=true, let  $\mathbb{T}^i = \mathbb{K}_v^i$ , STOP.
fin si
Let  $v = v + 1$ ;
si  $v < v_{MAX}$  entonces
    go to step 3
si no
    let success=false, STOP.
fin si

```

Algorithm 3 Mode-dependent N -step controllable sets

-
1. For all i , Set $\mathbb{K}_0^i = \mathbb{T}^i$. Let $v = 1$,
 2. For all i , compute $\mathbb{K}_v^i = \mathbb{Q}_i^C(\cap_{j \in \mathcal{S}_i} \mathbb{K}_{v-1}^j) \cap \mathbb{X}$
 3. Let $v = v + 1$; if $v = N$ stop else, go to step 2.
-

Then, we have (proof is omitted because is similar to (Kerrigan 2000)):

Lemma 4.1. *If $x \notin \mathbb{K}_v^i$, the horizon- v predictive control is infeasible if all admissible sequences of length $v + 1$ starting in mode i are considered in Θ for the optimisations (4.5) or (4.11). Controllable sets are nested: $\mathbb{K}_v^i \subset \mathbb{K}_w^i$ if $v \leq w$.*

4.5. Multiple-horizon Predictive Control

The proposed predictive control has a cost index depending on all sequences in a particular set Θ . The standard choice (Sokaert and Mayne 1998) would be selecting *all* sequences, up to a given prediction horizon. However, that is computationally very expensive in most cases (discussed in Remark 4.1), hence, long horizons are out of reach for most practical applications. So the problem is that short horizons may be *suboptimal*, but long horizons are *intractable*.

Definition 4.1. *A set of finite-length sequences $\Theta \subset \mathcal{G}$ is **irredundant** if there do not exist sequences η, ξ such that $\eta \in \Theta$ and $\eta\xi \in \Theta$. If a set of sequences is not irredundant, it will be denoted as **redundant**.*

Lemma 4.2. *Consider a redundant sequences set Θ . There exists an irredundant subset of it $\hat{\Theta} \subset \Theta$ so that the optimal value of the minimax predictive control is the same in Θ and in $\hat{\Theta}$.*

Proof: Indeed, in minimax case the shorter sequence will always have a greater or equal value for the cost bound (decision variables are identical to the longer one) so presence of the longer sequence is irrelevant. \square

Definition 4.2. *A set of sequences $\Theta_1 \subset \mathcal{G}$ is a **cover** of another set of sequences $\Theta_2 \subset \mathcal{G}$ if, for each $\theta^l \in \Theta_2$ there exists a pair of sequences γ, ξ such that $\gamma \in \Theta_1$ and $\theta^l = \gamma\xi$. We will denote such situation as $\Theta_1 \triangleleft \Theta_2$.*

For instance Θ_1 is a cover of Θ_2 with:

$$\Theta_1 = \{\{11\}, \{121\}, \{122\}\} \quad (4.15)$$

$$\Theta_2 = \{\{111\}, \{112\}, \{121\}, \{122\}\} \quad (4.16)$$

and both sets are irredundant. Also, $\mathcal{G}^N \triangleleft \mathcal{G}^M$ if $M \geq N$.

Theorem 4.1. *If the predictive control problem (4.5) or (4.11) is feasible with sequences set $\Theta = \Theta_1$, so it is with any set of sequences Θ_2 for which $\Theta_1 \triangleleft \Theta_2$. Also, the cost over Θ_2 will be lower than or equal to the cost over Θ_1 .*

Proof: If the problem (4.5) or (4.11) is feasible with $\Theta = \Theta_1$ with the suitable terminal constraints, delaying the terminal constraints to later times will also be feasible (that is the case with Θ_2) and cost will be lower or equal. \square

Theorem 4.2. *The solution of a predictive control problem with set of sequences Θ , if feasible, guarantees a finite cost bound for all admissible sequences if there exists an integer N such that $\Theta \triangleleft \mathcal{G}^N$.*

Proof: The main idea is that all cases are considered then, as Θ over-bounds to several sequence of the longer size, this fact force an earlier reaching of the terminal sets. \square

Joining Theorem 4.1 and Theorem 4.2, the following main theorem ensues in a straightforward way:

Theorem 4.3. *If $x_0 \in \mathbb{K}_N^i$, for any set of sequences Θ such that there exists M , $M \geq N$ such that $\mathcal{G}^N \triangleleft \Theta \triangleleft \mathcal{G}^M$, then the predictive controller is feasible and guarantees a finite cost bound for all admissible sequences.*

The above result is the most relevant of the section, indicating conditions for feasibility and finite cost (i.e., stability if future actions were applied in open-loop). Receding-horizon stability needs further refinements, discussed next.

4.5.1. Receding Horizon Implementation (minimax case)

Theorem 4.4. *Consider that a control action u_0 at $t = 0$ had been applied with mode θ_0 , considering a set of sequences Θ_0 , such that all sequences in Θ_0 have at least length 2. Consider now that mode θ_1 is available at $t = 1$. Consider the set of sequences:*

$$\Theta_{01} = \{\gamma : \theta_0\gamma \in \Theta_0, \text{ and } \exists \psi \text{ s.t. } \gamma = \theta_1\psi\}$$

The predictive controller for any set of sequences Θ_1 such that $\Theta_{01} \triangleleft \Theta_1 \triangleleft \Theta_M$ (for some M) is feasible and proves a lower than or equal to minimax cost bound that found at $t = 0$, i.e.:

$$J(\theta_1, \Theta_1) + x_0^T Q_{\theta_0} x_0 + u_0^T R_{\theta_0} u_0 \leq J(\theta_0, \Theta_0)$$

Proof: Indeed, the problem at $t = 0$ considered all sequences $\theta_0\gamma$ with $\gamma \in \Theta_{01}$ (evidently, because it did so for all $\theta_0\gamma \in \Theta_0$, and $\Theta_{01} \subset \Theta_0$), so $J(\theta_1, \Theta_{01}) + x_0^T Q_{\theta_0} x_0 + u_0^T R_{\theta_0} u_0 \leq J(\theta_0, \Theta_0)$ and $J(\theta_1, \Theta_{01})$ is feasible. As $\Theta_{01} \triangleleft \Theta_1$, Theorem 4.1 ensures feasibility at $t = 1$ and, too, the estimated minimax cost will be lower than that considered in the predictions at $t = 0$ starting with $\theta_0\theta_1$:

$$J(\theta_1, \Theta_1) + x_0^T Q_{\theta_0} x_0 + u_0^T R_{\theta_0} u_0 \leq J(\theta_1, \Theta_{01}) + x_0^T Q_{\theta_0} x_0 + u_0^T R_{\theta_0} u_0 \leq J(\theta_0, \Theta_0)$$

□

Theorem 4.5. *A set of sequences Θ is stable in receding horizon policy if it verifies the conditions in Theorem 4.3 and the set:*

$$\text{trim}(\Theta, \theta_0) := \{\gamma : \theta_0\gamma \in \Theta\} \quad (4.17)$$

obtained by trimming the first element of all sequences in Θ , hence, $\text{trim}(\Theta, \theta_0)$ is a cover of Θ , i.e., $\text{trim}(\Theta, \theta_0) \triangleleft \Theta$.

Proof: Note that, given a particular initial mode θ_0 , $\Theta_{01} \subset \text{trim}(\Theta, \theta_0)$ for any admissible θ_1 . So, if $\text{trim}(\Theta, \theta_0) \triangleleft \Theta$, the minimax cost bound can be shown to be a Lyapunov function, because it decreases, as the set of sequences

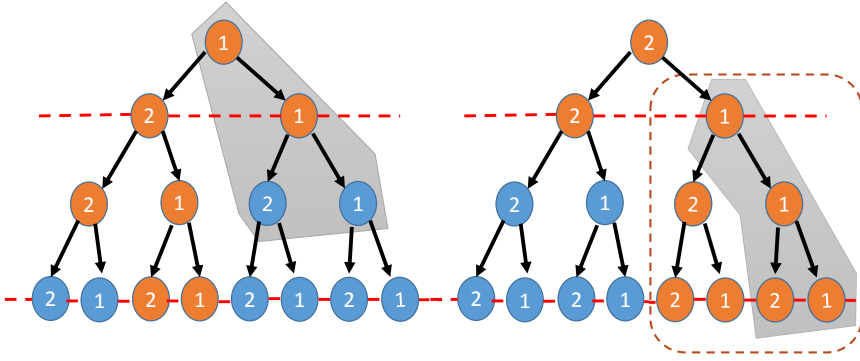


Figura 4-1: Not guaranteed receding horizon stability. Brown dot-line box are the sequence belong to Θ_{01} . Left tree represents the policy Γ_1 , whereas the right one Γ_2 .

at each step verifies the cover conditions in Theorem 4.4. \square

Example 4.1. Consider a system with two operation modes. If the following policy (using $\Theta = \Gamma_1$ when initial mode is 1, and $\Theta = \Gamma_2$ otherwise) is chosen:

$$\theta_0 = 1 : \Gamma_1 = \{11, 122, 1211, 1212\}$$

$$\theta_0 = 2 : \Gamma_2 = \{22, 2111, 2112, 2121, 2122\}.$$

It cannot be guaranteed receding horizon stable, because if $\theta_0 = 2$ and at next sample $\theta_1 = 1$, we have Θ_{01} is no a cover of Γ_1 , i.e., $\Theta_{01} \not\triangleleft \Gamma_1$ as:

$$\Theta_{01} = \{111, 112, 121, 122\}$$

so, the sequence $\{11\}$ does not belong to the above set Θ_{01} . This fact can be seen in Figure 4-1. Note that, the sequences within gray area of the right tree does not belong to the gray area of left tree in Figure 4-1. Therefore, with this selection it can be guaranteed the feasibility of the optimization problem, but not in receding horizon implementation.

Example 4.2 (Alternative choice). With $\Theta = \Gamma_1 \cup \Gamma_2$:

$$\theta_0 = 1 : \Gamma_1 = \{111, 112, 122, 1211, 1212\}$$

$$\theta_0 = 2 : \Gamma_2 = \{22, 2111, 2112, 2121, 2122\}.$$

- if $\theta_0 = 1, \theta_1 = 1$ we would have

$$\Theta_{01} = \{11, 12\} \triangleleft \Gamma_1$$

- if $\theta_0 = 1, \theta_1 = 2$ we would have

$$\Theta_{01} = \{22, 211, 212\} \triangleleft \Gamma 2$$

- if $\theta_0 = 2, \theta_1 = 1$ we would have

$$\Theta_{01} = \{111, 112, 121, 122\} \triangleleft \Gamma 1$$

- if $\theta_0 = 2, \theta_1 = 2$ we would have

$$\Theta_{01} = \{2\} \triangleleft \Gamma 2$$

It can be easily checked that this choice verifies the conditions in Theorem 4.5.

4.5.2. Choice of Sequence Sets

Once the above results characterising feasibility, cost improvement, guarantees of finite cost for all sequences, etc. are established, some criteria for choosing the sequence sets should be brought forward. Several possible options might arise:

- *Average-cost optimisation:* In this case, avoiding conservatism in high-likelihood cases should be addressed. Generating a tree of sequences by branching the most likely sequence (increasing the length by one, via adding the successors of the last element) might be a reasonable option.
- *Minimax-cost optimisation:* in this case, avoiding conservatism in most-likely cases (i.e., applying the same idea above) might also be a reasonable option. However, as the minimax cost bound may be hit with unlikely sequences, a plausible way of operating could be branching the sequences whose cost equals the obtained bound.

Note that the above ideas should be refined in order to verify the conditions in Theorem 4.5 for receding-horizon stability. Also, out of the N -step sets, success (asymptotically reaching the origin) can only be ensured with a certain probability. An in-depth analysis of these options, as well as intermediate cases (guaranteeing a cost bound with a certain probability) will be the deal in Chapter 7.

4.6. Numerical example

Consider the second-order MJLS (1), with $x_k \in \mathbb{R}^2$. Let $\theta_k \in \mathcal{M} = \{1, 2, 3\}$ represent modes of the system under three situations. The mode-dependent matrices of the system realization are:

$$A_1 = \begin{bmatrix} 0 & 1 \\ -2,5 & 3,2 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 & 1 \\ -4,3 & 4,5 \end{bmatrix}, \quad A_3 = \begin{bmatrix} 0 & 1 \\ 5,3 & -5,2 \end{bmatrix},$$

$B_1 = B_2 = B_3 = [0 \ 1]^T$. The weighting matrices are:

$$Q_1 = \begin{bmatrix} 3,6 & -3,8 \\ -3,8 & 4,87 \end{bmatrix}, \quad Q_2 = \begin{bmatrix} 10 & -3 \\ -3 & 8 \end{bmatrix}, \quad Q_3 = \begin{bmatrix} 5 & -4,5 \\ -4,5 & 5 \end{bmatrix},$$

and $R_1 = 2,6$, $R_2 = 1,165$, $R_3 = 1,111$. The state and input constraints will be:

$$-10 \leq x_k(1) \leq 10, \quad -10 \leq x_k(2) \leq 10, \quad -9 \leq u \leq 9$$

respectively. The transition probability matrix will be:

$$\mathbb{P} = \begin{bmatrix} 0,67 & 0,30 & 0,26 \\ 0,17 & 0,47 & 0,10 \\ 0,16 & 0,23 & 0,64 \end{bmatrix}$$

In order to compute the mode-dependent terminal sets, Algorithm 2 has been executed giving the results depicted in Figure 4-2. Once the terminal sets are obtained, Algorithm 3 is used to get the mode-dependent feasible sets for $N = 3$, these sets appear in Figure 4-3.

With these sets, a simulation starting in the 3-step set associated to mode 1 has been executed. The set of sequences to be considered has, hence, a minimum length of 4 (considering the different terminal modes). These sequences are a total of $3^4 = 81$ ones. Assuming that there are computing resources enough for 150 sequences, the 34 most likely sequence of length 4 have been branched to length 5, and the minimum average cost controller over the resulting set of 149 sequences (47 of length 4, 102 of length 5) has been simulated in receding horizon (the choice of sequences verifies Theorem 4.5). The time response appears on figure 4-4.

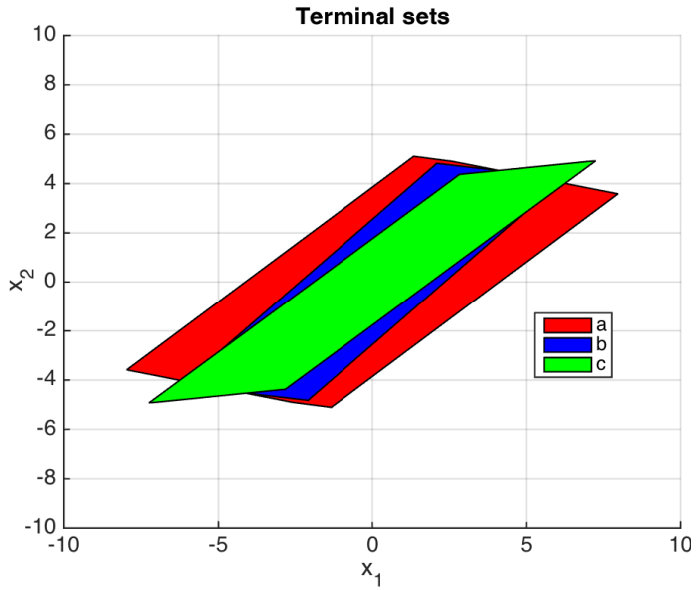


Figura 4-2: Mode-dependent terminal sets: a) (mode 1) \mathbb{T}^1 , b) (mode 2) \mathbb{T}^2 , c) (mode 3) \mathbb{T}^3 .

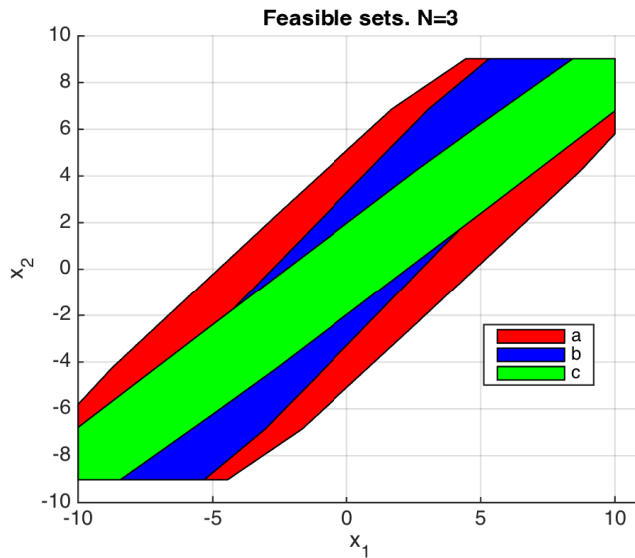


Figura 4-3: Mode dependent 3-step feasible sets: a) (mode 1) \mathbb{K}_3^1 , b) (mode 2) \mathbb{K}_3^2 , c) (mode 3) \mathbb{K}_3^3 .

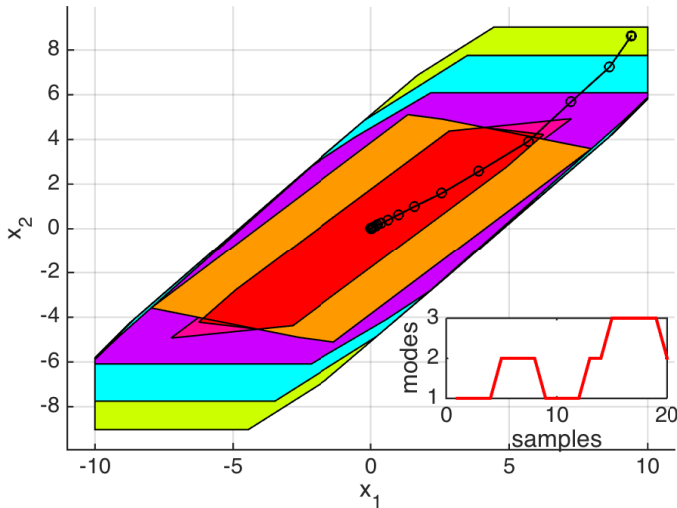


Figura 4-4: Simulation of a mixed length-4 and length-5 MPC set-up.

4.7. Conclusions of this Chapter

The possibility of simultaneously considering different prediction horizons in MPC problems has been presented, based on sequence-related definitions. The results apply to linear systems which switch between different operation modes with a given transition probability matrix. Conditions for feasibility and stability have been outlined. Some guidelines on how to select these sets of sequences based on their probability and the available computational resources have also been proposed. The results in this chapter open up possibilities for future research on a suitable joint management of (a) risk (probability of the cost above some thresholds, Chapter 7 and 8, (b) stability guarantees, see Chapter 5.

Capítulo 5

Ingredients for Stable Receding-horizon Scenario-based Predictive Control

Nota sobre el capítulo: En el capítulo anterior se propone un primer enfoque de MPC para MJLS basado en escenarios asumiendo que se dispone información del modo de operación del sistema al momento de calcular la acción de control. Aquí, se generaliza este concepto disponiendo de información con la cual se puede obtener una cierta probabilidad el modo de funcionamiento del sistema al momento de calcular la ley de control. El contenido de este capítulo se encuentra actualmente en revisión.

Abstract: This chapter discusses predictive control laws for discrete-time linear systems which jump between different modes according to a Markov probabilistic transition/observation model, minimising an average cost. Scenario predictive control deals only a subset of the possible transitions, due to the exponential explosion of the number of possible realisations as horizon grows. Prior literature enforces stability and feasibility via auxiliary Lyapunov functions. This work proposes terminal and scenario ingredients such that a receding horizon implementation of a stochastic predictive control law in a given scenario proves mean-square stability and recursive feasibility under hard input and state constraints.

5.1. Introduction

The main issue in the model predictive control for dynamic MJLS presented in prior chapter, is the exponential explosion on the number of possible mode realisations to be averaged as optimisation horizon grows, as well as, the number of input decision variables also grows exponentially. A preliminary idea presented in Chapter 6 propose a solution to solve such predictive control problem eliminating some sequences with low probability. *scenario-based* optimisation approaches have been proposed in prior literature (Calafiore *et al.* 2006, Bernardini and Bemporad 2012, Patrinos *et al.* 2014, Schildbach *et al.* 2014). The idea of the scenario approach is not considering ‘all’ possible future mode realisations, but only a reduced subset of them with ‘high enough’ probability. Chapter 4 shows how to formulate the predictive control when *simultaneously* considering prediction vectors of different lengths (understood as ‘*scenario*’). However, leaving out some possible mode realisations in the chosen scenario opens up the possibility of unfeasibility/instability or future constraint violation.

Regarding constraints, this chapter will consider only the case of ‘hard’ constraints, to be guaranteed at all future times for all admissible mode realisations¹⁶. For instance, in (Bernardini and Bemporad 2012), a hard Lyapunov-decrease constraint (with the terminal Lyapunov function) is added in order to ensure stability and recursive feasibility.

Note that, standard MPC for linear systems can prove such feasibility and stability issues ‘for free’ with mild assumptions on the cost index (see Section 1.5.3) without explicitly crafting a Lyapunov function to decrease. There

¹⁶Other authors propose accepting some likelihood of failure in chance-constrained approaches, in which a bound on the probability of violating constraints is set (Schwarm and Nikolaou 1999, Schildbach *et al.* 2015).

is, hence, a ‘gap’ between the extra requirements needed in the scenario approach (Bernardini and Bemporad 2012) and those in classic MPC (Goodwin *et al.* 2006, Chap. 4) or the MJLS version (Patrinos *et al.* 2014). One of the reasons is that the probabilistic model in (Bernardini and Bemporad 2012) is quite general, so handling possibly time-varying probabilities in stability issues needs to resort to ‘robust’ Lyapunov functions.

This work will consider transitions between modes to be governed by a probabilistic model involving both ‘mode’ observation probabilities and additional ‘information’ variables which will allow predictions on future mode probabilities. Mode variables will be assumed unmeasurable, but information ones will be assumed known at time of computing control actions.

The goal of this chapter is, in the above context, to set-up suitable (a) terminal ingredients and (b) scenario properties, such that recursive feasibility and mean-square stability are guaranteed in *receding-horizon* implementations, for any initial condition in a suitable feasible set (which will also be determined generalising the feasible set ideas in robust predictive control).

5.2. Preliminaries and Problem Statement

Consider the discrete-time LTV model:

$$x_{k+1} = A_{\theta_k} x_k + B_{\theta_k} u_k, \quad k \geq 0 \quad (5.1)$$

being θ_k a mode parameter taking values in a finite set $\theta_k \in \mathcal{M} = \{1, 2, \dots, M\}$, whereas A_j and B_j are constant matrices of appropriate dimensions.

By assumption, the sequence of mode variables $\theta^l = \{\theta_0^l, \theta_1^l, \theta_2^l, \dots\}$, $l \in \mathcal{L}$, is generated via a particular known probabilistic transition model, which can be used to compute the probability of present θ_k and future θ_{k+j} modes, $j > 0$, conditioned to some ‘information’, to be denoted as z_k , assumed available at each sample k (details later).

Cost Index Optimisation. Given a particular mode sequence realisation θ^l , the cost index for such sequence is defined as (4.4). Given the above-discussed probabilistic model, the probability of each sequence conditioned to current information, $p(\theta^l | z_0)$, can be computed by assumption. Then, the predictive control objective is to obtain the sequence of actions minimising (4.5) (min-max case) or (4.11) (Average cost case).

By assumption, the above problem must be solved under *hard* constraints

in the form:

$$x_k^{[l]} \in \mathbb{X}, \quad u_k^{[l]} \in \mathbb{U} \quad \forall k \geq 0 \quad \forall \theta^l \text{ s.t. } p(\theta^l | z_0) \neq 0 \quad (5.2)$$

for known sets \mathbb{X}, \mathbb{U} , usually polyhedral. These constraints will be later generalised to be dependent on the measurable information variables.

The solution to the predictive control problem appears in Chapter 3, as well as, in the work (Patrinos *et al.* 2014, Section 4). However, in order to get an exact solution for this problem, all possible future realizations of θ_k should, ideally, be considered. Note that the number of terms in the summation in (4.11) grows exponentially (at a rate up to M^N), as well as the number of mode-dependent control decision variables in \mathcal{U}^l , therefore, long horizons N require an insurmountable computational effort.

Scenarios. In order to alleviate the computational load, the scenario-based predictive control approach (Bernardini and Bemporad 2012, Schildbach *et al.* 2014) proposes minimising the cost index (4.11) only considering a subset of the possible realisations with ‘high-enough’ probability; the length of each of the realisations may differ so it is a *multiple-horizon* setting. The key idea to solve such problem is recasting it as an optimisation over a tree. The major drawback of the cited approaches is that omitting some sequences spoils stability/feasibility guarantees, so additional constraints (Lyapunov-based) must be enforced. In the approach to be followed here, the need of such additional Lyapunov-based constraints will be relieved.

5.2.1. Information Model

Throughout this chapter, we will consider there exist M mode variables θ_k and information variables $z_k \in \mathcal{Z} = \{1, \dots, Z\}$ which evolve in time according to known transition (π_{ijk}) and mode observation (e_{ij}) probability matrices defined below:

$$p(z_{k+1} = i | \theta_k = j, z_k = l) := \pi_{ijl}, \quad (i, j, l) \in \mathcal{Z} \times \mathcal{M} \times \mathcal{Z} \quad (5.3)$$

$$p(\theta_k = i | z_k = j) := e_{ij}, \quad (i, j) \in \mathcal{M} \times \mathcal{Z} \quad (5.4)$$

where z denote the *information variables*. Particular examples of this probabilistic set-up may be:

- $z_k = \theta_k$ (current mode assumed measurable, evolving from a Markov chain with transition probabilities $p(\theta_{k+1} = i | \theta_k = l) = p_{il}$, as used in

(Costa *et al.* 2005, Patrinos *et al.* 2014)). This amounts to $e_{ij} = 1$ if $i = j$ and $e_{ij} = 0$ otherwise, and $\pi_{ijl} = p_{jl}$. These systems are the MJLS ones utilized in the above chapters.

- Prior mode evolving from a Markov chain (same transition probabilities p_{il}) assumed measurable, i.e., $z_k = \theta_{k-1}$. In this case, we would set $e_{il} = p_{il}$, and $\pi_{ijl} = 1$ if $i = j$ and zero otherwise.

Assumption 5.1. *The information variable z_k will be assumed measurable at time k .*

Definition 5.1. *The **admissible modes** for a given information z will be the set $\text{adm}_e(z) := \{i \in \mathcal{M} : e_{iz} > 0\}$. The **admissible successor** information values for some $\delta := (\theta, z) \in \mathcal{M} \times \mathcal{Z}$ will be denoted by $\text{adm}_\delta(\theta, z) := \{i \in \mathcal{Z} : \pi_{i\theta z} > 0\}$.*

Constraints: Information-dependent state and input constraints will be considered. In particular, two ordered lists of sets

$$\mathbf{X}^* := \{\mathbb{X}^{[1]}, \dots, \mathbb{X}^{[Z]}\}, \mathbf{U}^* := \{\mathbb{U}^{[1]}, \dots, \mathbb{U}^{[Z]}\}$$

will be assumed known. The lists encode the constraints for each information z , understood as $x_{k+t|k}^{[l]}$ must belong to $\mathbb{X}^{[z_k]}$ and $u_{k+t|k}^{[l]}$ must belong to $\mathbb{U}^{[z_k]}$. The origin will be assumed to belong to every $\mathbb{X}^{[z]}$ and $\mathbb{U}^{[z]}$, for all $z \in \mathcal{Z}$.

5.2.2. Infinite Horizon and Mean-square Stability.

Under the above-presented probabilistic model, the cost (4.4) can now be recursively expressed as:

$$J_N(x_0, \mathcal{U}_0, z_0) = \sum_{i \in \mathcal{M}} e_{iz_0} \hat{l}(x_0, i, u_0) + \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{Z}} \pi_{jiz_0} e_{iz_0} J_{N-1}(A_i x_0 + B_i u_0, \mathcal{U}_1, j) \quad (5.5)$$

where $\hat{l}(x_0, i, u_0)$ is a mode-dependent *stage cost*, \mathcal{U}_0 is the collection of all sequence-dependent \mathcal{U}^l and \mathcal{U}_1 the decision variables from instant 1 onwards. Note that, the stage cost can be changed to an equivalent information-dependent expression $l(x, z, u)$, for $z \in \mathcal{Z}$:

$$l(x, z, u) := \mathbb{E}(\hat{l}(x, \theta, u) | z) = \sum_{i \in \mathcal{M}} e_{iz} \hat{l}(x, i, u) \quad (5.6)$$

where $\mathbb{E}(\cdot)$ denotes the mathematical expectation over all mode sequences, given their probability.

If a particular controller $u = K(x, z)$ is in operation, control decision variables can be computed from x and z at each instant, so, abusing the notation, we

can omit argument \mathcal{U}^l in (5.5) above, computing the cost $J_N^K(x, z)$ for a given control policy. Also, letting $N \rightarrow \infty$, denoting $J^K(x, z) := \lim_{N \rightarrow \infty} J_N^K(x, z)$, such cost will fulfil the Bellman equation:

$$J^K(x_0, z_0) = l(x_0, z_0, K(x_0, z_0)) + E_+(J^K, x_0, K(x_0, z_0), z_0) \quad (5.7a)$$

where E_+ is defined, for any $\mathcal{V} : \mathbb{R}^n \times \mathcal{Z} \mapsto \mathbb{R}$, as:

$$E_+(\mathcal{V}, x, u, z) := \sum_{i \in \text{adm}_e(z)} \sum_{j \in \text{adm}_\delta(i, z)} \pi_{jiz} e_{iz} \mathcal{V}(A_i x + B_i u, j) \quad (5.7b)$$

$E_+(\cdot)$ should be understood as the expected value of \mathcal{V} at next instant knowing current state, information and control action. Note that the definition of E_+ removes zero-probability combinations in the rightmost summation in (5.5).

Of course, if there exists a function \mathcal{V} which fulfils:

$$\mathcal{V}(x, z) \geq l(x, z, K(x, z)) + E_+(\mathcal{V}, x, K(x, z), z) \quad (5.8)$$

then,

$$\mathcal{V}(x, z) \geq J^K(x, z)$$

The above equation has close ties with stability notions for Markov-jump linear systems. In particular, our model (5.1), in closed-loop with some controller $K(x, z)$, will be *mean-square stable* if $\lim_{k \rightarrow \infty} \mathbb{E}(x_k^2) = 0$. The following Lyapunov theorem for mean-square stability, from minor modifications of those in (Costa *et al.* 2005, Chapter 3), (Zhang and Boukas 2009), (Patrinos *et al.* 2014, Thm. 20), will be used in this paper¹⁷:

Theorem 5.1. *If there exists a function $\mathcal{V}(x, z)$ and a controller $u(x, z)$, such that $\mathcal{V}(0, z) = 0$, $\mathcal{V}(x, z) > 0$ for all $x \neq 0$, for all z , such that it fulfils the stochastic Lyapunov decrease condition, for some $\gamma > 0$:*

$$\mathcal{V}(x, z) - E_+(\mathcal{V}, x, u(x, z), z) > \gamma x^T x \quad \forall x \neq 0 \quad (5.9)$$

then the system is mean-square stable.

¹⁷Indeed, the above theorem is stated in the cited works for MJLS, but suitably averaging on θ , as $p(z_{k+1} = j | z_k = l) = \sum_i \sum_j \pi_{ijl} e_{il}$, we can consider the evolution of the system as a MJLS on z , and making the Lyapunov function depend on just z ; in this setting, θ would be analogous to a random disturbance.

5.2.3. Problem Statement

As previously discussed, scenario tree approaches in prior literature prove stability by enforcing an additional Lyapunov-decrease constraint, see (Bernardini and Bemporad 2012, Eq. (25h)); then, recursive feasibility is also enforced by constraining next state to lie inside a guaranteed feasible set (associated to a control-invariant Lyapunov level set (Bernardini and Bemporad 2012, Eq.(25g))).

The aim of this chapter is to set-up suitable *terminal ingredients* and *geometric* conditions on the scenario tree, such that *recursive feasibility* and *mean-square stability* are guaranteed in *receding-horizon* implementations for any initial condition in a suitable *feasible set*. The transition model to which such guarantees will apply will be the one discussed in Section 5.2.1.

5.3. Terminal Ingredients of the Scenario-based Approach

In order to solve the above-stated problem for a given stage cost $l(x, z, u)$, let us first propose some conditions on terminal sets and cost function, as follows.

Assumption 5.2. *The stage cost, $l(x, z, u)$, fulfils $l(0, z, 0) = 0$ and $l(x, z, u) \geq \gamma x^T x$ with $\gamma > 0$ for all $x \in \mathbb{X}^{[z]}$, for all $z \in \mathcal{Z}$.*

Theorem 5.2. *If Assumption 5.2 holds and there exists a controller $K(x, z)$ and a function $\mathcal{V}(x, z)$ such that:*

1. *there exists an indexed collection of **information-dependent terminal sets**, $\mathbf{T}^* := \{\mathbb{T}^{[1]}, \dots, \mathbb{T}^{[Z]}\}$, and an **information-dependent terminal control law** $u = K(x, z)$ such that, for all $z \in \mathcal{Z}$ and for all $x \in \mathbb{T}^{[z]}$, admissibility condition $K(x, z) \in \mathbb{U}^{[z]}$ holds, and:*

$$A_i x + B_i K(x, z) \in \bigcap_{j \in \text{adm}_\delta(i, z)} \mathbb{T}^{[j]} \quad \forall i \in \text{adm}_e(z) \quad (5.10)$$

2. *inequality (5.8) holds for all $z \in \mathcal{Z}$, for all $x \in \mathbb{T}^{[z]}$,*

then, the terminal control law will be admissible in future times (recursive feasibility, $u_k \in \mathbb{U}^{[z_k]}$), as well as mean-square stable, if initial conditions (x_0, z_0) are such that $x_0 \in \mathbb{T}^{[z_0]}$. Furthermore, $J^K(x, z) \leq \mathcal{V}(x, z)$.

Proof: Indeed, Equation (5.8) jointly with Assumption 5.2 ensures that (5.9) holds; on the other hand, conditions (5.10) recursively ensure that $x_k \in \mathbb{T}^{[z_k]}$ for all $k \geq 1$ if $x_0 \in \mathbb{T}^{[z_0]}$, so the terminal controller K will never violate constraints in the future. \square

5.3.1. Quadratic Case

For computational reasons, a quadratic stage cost $l(x, z, u) := x^T Q_z x + u^T R_z u$ will be assumed throughout this chapter, with $Q_z - \gamma I \geq 0$, $\gamma > 0$, and $R_z \geq 0$; notation $Q \geq 0$ on matrices will be understood as Q being positive semidefinite.

In order to obtain \mathbf{T} , \mathcal{V} and K when the stage cost is quadratic, a feasible solution for the Theorem 5.3 below can be pursued. Then, if such feasible decision variables are found, the controller $K(x, z) = F_z P_z^{-1} x$ will be the one ensuring that $\mathcal{V}(x, z) = x^T P_z^{-1} x$ fulfils (5.8).

Theorem 5.3. *If there exists $P_z \geq \epsilon I$, for some $\epsilon > 0$, for all $z \in \mathcal{Z}$ and Γ_{ijz} , $i \in \text{adm}_e(z)$, $j \in \text{adm}_\delta(i, z)$ such that LMI's (5.11a) and (5.11b) below hold for all $z \in \mathcal{Z}$, the cost function $\mathcal{V}(x, z) = x^T P_z^{-1} x$ and controller $K(x, z) = F_z P_z^{-1} x$, fulfil (5.8) for $l(x, z, u) = x^T Q_z x + u^T R_z u$:*

$$\begin{pmatrix} P_z - \sum_{i \in \text{adm}_e(z)} \sum_{j \in \text{adm}_\delta(i, z)} \pi_{jiz} e_{iz} \Gamma_{ijz} & P_z & F_z^T \\ P_z & Q_z^{-1} & \mathbf{0} \\ F_z & \mathbf{0} & R_z^{-1} \end{pmatrix} \geq 0 \quad (5.11a)$$

$$\begin{pmatrix} \Gamma_{ijz} & P_z A_i^T + F_z^T B_i^T \\ A_i P_z + B_i F_z & P_j \end{pmatrix} \geq 0 \quad \begin{matrix} \forall i \in \text{adm}_e(z) \\ \forall j \in \text{adm}_\delta(i, z) \\ \forall z \in \mathcal{Z} \end{matrix} \quad (5.11b)$$

Proof: Well-known Schur's complements manipulations easily show that the LMIs above imply (5.8) \square

Once the terminal controller and cost are found, the following steps should be carried out, in order to obtain the collection of terminal sets \mathbf{T}^* .

Information Dependent One-step Set. Let us denote as $\mathbf{C}^* = \{\mathcal{C}^{[1]}, \dots, \mathcal{C}^{[Z]}\}$ an indexed collection of arbitrary sets. Inspired in (Patrinos *et al.* 2014, Def. 7), let us define the **information-dependent 1-step** set as follows:

$$\mathcal{Q}(z, \mathbf{C}^*) := \{x \in \mathbb{X}^{[z]} : u = K(x, z) \in \mathbb{U}^{[z]}, A_i x + B_i u \in \bigcap_{j \in \text{adm}_\delta(i, z)} \mathcal{C}^{[j]} \forall i \in \text{adm}_e(z)\} \quad (5.12)$$

Terminal Information-dependent Sets can be computed by initialising $\mathbf{C}_{[0]}^* := \{\mathbb{X}^{[1]}, \dots, \mathbb{X}^{[Z]}\}$, and carrying out iteration:

$$\mathbf{C}_{[k+1]}^* := \{\mathcal{Q}(1, \mathbf{C}_{[k]}^*), \mathcal{Q}(2, \mathbf{C}_{[k]}^*), \dots, \mathcal{Q}(Z, \mathbf{C}_{[k]}^*)\} \quad (5.13)$$

until convergence; then, $\mathbf{T}^* := \mathbf{C}_{[\infty]}^*$.

Indeed, the above iterations adapt to the setting in consideration well-known ideas dating from (Blanchini 1994, Kerrigan 2000), and recent related approaches in (Patrinos *et al.* 2014, Def. 13, Lemma 16). The converged set fulfils:

$$\mathbf{C}_{[\infty]}^* := \{\mathcal{Q}(1, \mathbf{C}_{[\infty]}^*), \mathcal{Q}(2, \mathbf{C}_{[\infty]}^*), \dots, \mathcal{Q}(Z, \mathbf{C}_{[\infty]}^*)\} \quad (5.14)$$

so it fulfils conditions in (5.10) and admissibility of the control action, $K(x, z) \in \mathbb{U}$.

Assumption 5.3. *It will be assumed that iteration (5.13) converges in a finite number of iterations¹⁸.*

Remark 5.1. *The terminal sets in \mathbf{C}^* are not ‘invariant’ in the state-only space, as defined in, for instance (Kerrigan 2000). However, they are invariant in an augmented (x, z) space, analogous to the ones in (Patrinos *et al.* 2014). Also, under Assumption 5.3, if all $\mathbb{X}^{[z]}$ are polytopic, then $\mathbb{T}^{[z]}$ will be polytopic, too, computable with widely-known software (MPT, (Herceg *et al.* 2013)).*

It can be proved that the terminal controller and Lyapunov function obtained with Theorem 5.3, jointly with the \mathbf{T}^* resulting from (5.13), fulfil the requirements of Theorem 5.2, so terminal controllers and associated sets generalising those of standard MPC or MJLS (Patrinos *et al.* 2014, Costa *et al.* 2005) have been found.

Example 5.1. *Consider the second-order discrete-time linear system of the form (5.1), with:*

$$A_i = \begin{pmatrix} -0,8 & 1 \\ 0,1 & w_i \end{pmatrix}, \quad B_i = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

where $i \in \mathcal{M}$, with $\mathcal{M} = \mathcal{Z} = \{1, 2\}$ and $w_i = \{0,85, -0,9\}$. Transition probability matrices are:

$$\pi_{ij1} = \begin{pmatrix} 0,6 & 1 \\ 0,4 & 0 \end{pmatrix}, \quad \pi_{ij2} = \begin{pmatrix} 0,8 & 1 \\ 0,2 & 0 \end{pmatrix}, \quad e_{ij} = \begin{pmatrix} 0,65 & 0 \\ 0,35 & 1 \end{pmatrix} \quad (5.15)$$

¹⁸ Conditions for such convergence are discussed in, for instance (Patrinos *et al.* 2014, Lemma 21). Alternatively, we can assume that a finitely-determined subset of the terminal sets can be obtained with λ -contractive argumentations analogous to (Blanchini 1994, Theorem 3.1), adapting (5.12) by considering $\lambda \mathcal{C}^{[j]}$ inside the intersection. replacing $\mathcal{C}^{[j]}$ by $\lambda \mathcal{C}^{[j]}$, $\lambda \leq 1$.

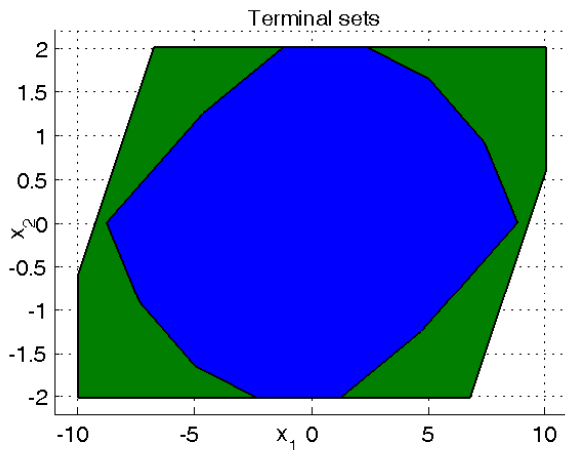


Figura 5-1: Information-dependent terminal sets. The smallest (blue) polytope represents $\mathbb{T}^{[1]}$; the largest one is $\mathbb{T}^{[2]}$.

The weight matrices used in the stage cost $l(x, z, u)$ are:

$$Q_z = \begin{pmatrix} 1 & 0 \\ 0 & 5 \end{pmatrix}, \quad R_z = 1$$

Using Theorem 5.3 and minimising the trace of $(p_1 * P_1 + p_2 * P_2)$, being p_1 and p_2 stationary probabilities of z —see footnote 17—, it is possible to obtain the following **information-dependent terminal controllers**:

$$K(x, 1) = (-0,184 \quad -0,037) x, \quad K(x, 2) = (-0,191 \quad 1,012) x$$

Finally, the two information-dependent terminal sets are shown in Figure 5-1, after four iterations needed for convergence. The constraint sets are:

$$\mathbb{X}^{[1]} = \mathbb{X}^{[2]} = \{(x_1, x_2) : |x_1| \leq 10, |x_2| \leq 2\}, \quad \mathbb{U}^{[1]} = \mathbb{U}^{[2]} = \{u : |u| \leq 1\}$$

5.4. Predictive Control on Scenarios/trees

Definition 5.2. A *tree* \mathcal{T} (also denoted as *scenario*) will be defined as a finite collection of ‘node’ objects such that each node \mathcal{N} has a single predecessor, except a single node to be denoted as root node, $\text{root}(\mathcal{T})$.

- For each node, its **predecessor** will be denoted as $\text{pre}(\mathcal{N})$ and the set of **suc-**

cessors, as $\text{succ}(\mathcal{N})$. If $\text{succ}(\mathcal{N}) = \emptyset$ then the node will be said to be a **leaf node**; the set of leaf nodes in \mathcal{T} will be denoted as $\text{leaf}(\mathcal{T})$.

- The **depth** of a node \mathcal{N} will be defined as 1 plus the number of recursive $\text{pre}(\cdot)$ operations until the root node is reached from \mathcal{N} . The depth of a tree, $\text{depth}(\mathcal{T})$, will be the maximum depth of its nodes. The **cardinal** of a tree $\text{card}(\mathcal{T})$ will denote the number of nodes of the tree.
- Each node \mathcal{N} will be associated with a **mode-information pair** $\delta(\mathcal{N}) := (\theta(\mathcal{N}), z(\mathcal{N}))$, $\theta(\mathcal{N}) \in \mathcal{M} \cup \emptyset$, $z(\mathcal{N}) \in \mathcal{Z}$. $\theta(\mathcal{N})$ will denote the mode leading to the node \mathcal{N} at previous sample, and $z(\mathcal{N})$ will denote the information allowing to compute the probabilities of the current mode and those of successor information. Values assigned to a tree will be defined to be that of its root node: $\theta(\mathcal{T}) := \theta(\text{root}(\mathcal{T}))$, $z(\mathcal{T}) := z(\text{root}(\mathcal{T}))$, $\delta(\mathcal{T}) := \delta(\text{root}(\mathcal{T}))$.

Assumption 5.4. In the sequel, for all considered trees there will be no two nodes such that there exists \mathcal{N} for which $\mathcal{N}_1, \mathcal{N}_2 \in \text{succ}(\mathcal{N})$ and $\delta(\mathcal{N}_1) = \delta(\mathcal{N}_2)$, i.e., δ is unique between successors of the same parent node.

Definition 5.3. The **conditional probability** of a node \mathcal{N}_1 given its parent \mathcal{N}_p , will be defined as:

$$p(\mathcal{N}_1 | \mathcal{N}_p) := \pi_{z(\mathcal{N}_1) \theta(\mathcal{N}_1) z(\mathcal{N}_p)} \cdot e_{\theta(\mathcal{N}_1) z(\mathcal{N}_p)} \quad (5.16)$$

A tree is said to be **probabilistic-complete** if for every non-leaf node \mathcal{N} ,

$$\sum_{\mathcal{N}' \in \text{succ}(\mathcal{N})} p(\mathcal{N}' | \mathcal{N}) = 1 \quad (5.17)$$

A tree will be **non-overbranched** if $p(\mathcal{N}' | \mathcal{N}) > 0$ for all $\mathcal{N}' \in \text{succ}(\mathcal{N})$. A tree which is both probabilistic-complete and non-overbranched will be denoted as **strictly complete**. Finally, a tree \mathcal{T} will be said to be **non-trivial** if it is strictly complete and $\text{card}(\mathcal{T}) \geq 2$.

Overbranched trees should be avoided, as they induce extra computing cost and conservatism by forcing to consider non-admissible transitions.

The probability of node \mathcal{N}_1 conditioned to the root node in an scenario \mathcal{T} , denoted by $p(\mathcal{N}_1)$, can be recursively defined from its parent's probability:

$$p(\mathcal{N}_1) := p(\mathcal{N}_p) \cdot p(\mathcal{N}_1 | \mathcal{N}_p), \quad p(\text{root}(\mathcal{T})) := 1 \quad (5.18)$$

If a tree is probabilistic-complete, it can be proved that

$$\sum_{\mathcal{N} \in \text{leaf}(\mathcal{T})} p(\mathcal{N}) = 1 \quad (5.19)$$

Also, if a tree is strictly complete, successors of non-leaf nodes *exactly* span all admissible mode-information transitions. The proof of such assertions is straightforward, omitted for brevity.

Remark 5.2. Note that $\theta(\mathcal{N}_p)$, is not used in (5.16): the value of $\theta(\mathcal{T})$, i.e., θ in the root node, is irrelevant for probability computations.

As the above remark points out that θ associated to the root node is irrelevant, we will define the following (sorted) array associated to each node:

$$\zeta(\mathcal{N}) := [z(\mathcal{T}), \dots, \delta(\text{pre}(\text{pre}(\mathcal{N}))), \delta(\text{pre}(\mathcal{N})), \delta(\mathcal{N})]$$

In this way, each node in the tree can be indexed by the above unique sequence of $\delta(\mathcal{N})$ variables leading to it, starting from the root node.

A tree \mathcal{T} will be identified (uniquely up to possible different $\theta(\mathcal{T})$) with the set of $\zeta(\mathcal{N})$ for all its nodes:

$$\Xi(\mathcal{T}) := \{\zeta(\mathcal{N}) : \mathcal{N} \in \mathcal{T}\} \quad (5.20)$$

With the set of sequences Ξ , equality, inclusion, and union of non-empty trees can be defined as:

Definition 5.4. *Tree operations:*

- A tree \mathcal{T}_1 is equal to \mathcal{T}_2 , i.e., $\mathcal{T}_1 = \mathcal{T}_2$, if $\Xi(\mathcal{T}_1) = \Xi(\mathcal{T}_2)$.
- $\mathcal{T}_1 \subset \mathcal{T}_2$ will be defined equivalent to $\Xi(\mathcal{T}_1) \subset \Xi(\mathcal{T}_2)$.
- The union of two trees $\mathcal{T}_1 \cup \mathcal{T}_2$ such that $z(\mathcal{T}_1) = z(\mathcal{T}_2)$ will be defined as the tree such that $\Xi(\mathcal{T}_1 \cup \mathcal{T}_2) := \Xi(\mathcal{T}_1) \cup \Xi(\mathcal{T}_2)$.

Note that union of trees without coincident information at the root node is undefined.

Definition 5.5. The *sub-tree* emanating from a node \mathcal{N} on tree \mathcal{T} will be denoted by $\mathcal{E}_{\mathcal{T}}(\mathcal{N})$, so $\text{root}(\mathcal{E}_{\mathcal{T}}(\mathcal{N})) = \mathcal{N}$. The nodes of $\mathcal{E}_{\mathcal{T}}(\mathcal{N})$ will be assumed to keep the mode-information pair δ unchanged from that of the nodes they originate from. The collection of trees arising from suppressing a root node will be defined by a “trim” operator:

$$\text{trim}(\mathcal{T}) := \{\mathcal{E}_{\mathcal{T}}(\mathcal{N}) : \mathcal{N} \in \text{succ}(\text{root}(\mathcal{T}))\} \quad (5.21)$$

If \mathcal{T} is depth 1, i.e., it has a single node, $\text{trim}(\mathcal{T}) := \emptyset$.

Example 5.2. Figure 5-2 shows a scenario example in which leaf nodes are represented in dark red colour, whereas non-leaf are in light blue colour. As $\theta(\mathcal{T})$ is not relevant for probability computation, any arbitrary value could have been used; the empty-set symbol at the root node has been used to denote such situation.

Consider the node \mathcal{N} , labelled with \blacklozenge . Then,

$$\zeta(\mathcal{N}) = \{1, \binom{1}{1}, \binom{2}{1}, \binom{1}{1}, \binom{1}{2}\}$$

The conditional probability of such node is computed using the transition matrices (5.15) as:

$$p(\mathcal{N}) = 1 * \pi_{111}e_{11} * \pi_{121}e_{21} * \pi_{111}e_{11} * \pi_{211}e_{11} = 0,02$$

Let us show a couple of examples for Definition 5.4. First, the definition can be used to formally assert that the tree emanating from the node labelled with \blacktriangle is 'equal' to that emanating from \blacktriangledown , as the initial θ is disregarded when the node becomes a root node. Also, the tree emanating from node \blacktriangledown , i.e.: $\zeta(\mathcal{N}) = \{\binom{\emptyset}{1}, \binom{2}{1}\}$ is a sub-tree of the whole tree. Last, $\text{trim}(\mathcal{T})$ will originate a set of four trees, whose root node lies at the horizontal dashed line. This fact will be key for ensuring recursive feasibility of predictive control problems, to be later discussed.

5.4.1. Cost Index Associated to a non-empty Tree

Definition 5.6. Each node \mathcal{N} of a non-empty tree \mathcal{T} will be assigned a **node-dependent control decision variable** $u_{\mathcal{T}}(\mathcal{N})$. The first control variable, key for receding-horizon implementation, will be singled out by notation $u(\mathcal{T}) := u_{\mathcal{T}}(\text{root}(\mathcal{T}))$. Given a measured plant state x , a **node state** $x(\mathcal{N})$ will be recursively defined as:

$$x(\mathcal{N}) := A_{\theta(\mathcal{N})}x(\text{pre}(\mathcal{N})) + B_{\theta(\mathcal{N})}u_{\mathcal{T}}(\text{pre}(\mathcal{N})) \quad (5.22)$$

starting with $x(\text{root}(\mathcal{T})) := x$. Decision variables for leaf nodes will be left empty, i.e.: $u_{\mathcal{T}}(\mathcal{N}) := \emptyset \ \forall \mathcal{N} \in \text{leaf}(\mathcal{T})$, as they are not used in (5.22). The set of node-dependent control variables of the whole tree \mathcal{T} will be denoted by $\mathcal{U}(\mathcal{T}) := \{u_{\mathcal{T}}(\mathcal{N}) : \mathcal{N} \in \mathcal{T}\}$.

Let us assume that terminal ingredients $\{\mathcal{V}(x, z), K(x, z), \mathbf{T}^*\}$ fulfilling conditions in Theorem 5.2 are available.

Definition 5.7 (Cost). Each non-empty tree will have an associated cost function $J(\mathcal{T}, x, \mathcal{U}(\mathcal{T}))$ defined as follows:

1. If the tree \mathcal{T} contains a single (leaf) node, the cost will be defined as

$$J(\mathcal{T}, x, \mathcal{U}(\mathcal{T})) := \mathcal{V}(x, z(\mathcal{T})) \quad (5.23)$$

being $\mathcal{V}(x, z)$ any terminal cost bound fulfilling (5.8) (for instance, the one arising from Theorem 5.3), recalling that $\mathcal{U}(\mathcal{T}) = \emptyset$.

2. If the tree \mathcal{T} has more than one node, the cost function $J(\mathcal{T}, x, \mathcal{U}(\mathcal{T}))$ will be recursively defined as:

$$\begin{aligned} J(\mathcal{T}, x, \mathcal{U}(\mathcal{T})) &:= l(x, z(\mathcal{T}), u(\mathcal{T})) + \\ &\sum_{\mathcal{T}' \in \text{trim}(\mathcal{T})} \pi_{z(\mathcal{T}')\theta(\mathcal{T}')z(\mathcal{T})} e_{\theta(\mathcal{T}')z(\mathcal{T})} J(\mathcal{T}', A_{\theta(\mathcal{T}')}x + B_{\theta(\mathcal{T}')}u(\mathcal{T}), \mathcal{U}(\mathcal{T}')) \end{aligned} \quad (5.24)$$

Note that, from (5.16), expression (5.24) defines a conditional average cost over all considered future sub-trees.

5.4.2. Constraints and Predictive Control Problem Set-up

Considering the terminal ingredients and information-dependent constraints $\mathbb{X}^{[z]}$, and $\mathbb{U}^{[z]}$, $z \in \mathcal{Z}$, we can set up the following optimisation problem:

Definition 5.8 (Constrained scenario MPC). *Given a scenario \mathcal{T} and a measured state x , the **constrained scenario predictive control problem** is defined to be the one consisting on minimising $J(\mathcal{T}, x, \mathcal{U}(\mathcal{T}))$ in Definition 5.7, over decision variables $\mathcal{U}(\mathcal{T})$, subject to constraints:*

$$u(\mathcal{N}) \in \mathbb{U}^{[z(\mathcal{N})]} \quad \forall \mathcal{N} \in \mathcal{T} \sim \text{leaf}(\mathcal{T}) \quad (5.25a)$$

$$x(\mathcal{N}) \in \mathbb{X}^{[z(\mathcal{N})]} \quad \forall \mathcal{N} \in \mathcal{T} \sim \text{leaf}(\mathcal{T}) \quad (5.25b)$$

$$x(\mathcal{N}) \in \mathbb{T}^{[z(\mathcal{N})]} \quad \forall \mathcal{N} \in \text{leaf}(\mathcal{T}) \quad (5.25c)$$

where (5.25c) are terminal constraints. If the problem is feasible, the optimal cost will be denoted as $J^*(\mathcal{T}, x)$, and the optimal decision variables as $\mathcal{U}^*(\mathcal{T}, x)$, being the optimal ones for each node labelled as $u_{\mathcal{T}}^*(\mathcal{N}, x)$.

Remark 5.3. *The set-up closely resembles that in (Bernardini and Bemporad 2012, Sec B). The differences are the inclusion of **terminal constraints** and **information-dependent** state and control constraints, as well as elimination of the **Lyapunov-decrease constraints**, present in the cited work. The terminal ingredients, jointly*

with some tree properties will allow to prove recursive feasibility and stability without the need of such Lyapunov-decrease enforcing.

Note: The reason of $\delta(\mathcal{N})$ including the mode leading to it, $\theta(\mathcal{N})$, is to allow to take into account that state-feedback will be carried out at future samples, so future actions will be different for each value of the ‘current’ mode; indeed, the idea is well established since (Scokaert and Mayne 1998). So, $\delta(\mathcal{N})$ contains a ‘state-feedback’ component θ plus measurable information, z , for later predictions.

5.5. Feasibility Analysis

Definition 5.9. Each non-empty tree \mathcal{T} will have an associate *feasible set* $\mathcal{K}(\mathcal{T})$, defined as follows:

- If the tree has a single node, define $\mathcal{K}(\mathcal{T}) := \mathbb{T}^{[z(\mathcal{T})]}$ (i.e., as the terminal set).
- Otherwise, define $\mathcal{K}(\mathcal{T})$ recursively as:

$$\mathcal{K}(\mathcal{T}) := \{x \in \mathbb{X}^{[z(\mathcal{T})]} : \exists u \in \mathbb{U}^{[z(\mathcal{T})]} \text{ s.t.} \\ \forall \mathcal{T}' \in \text{trim}(\mathcal{T}) \ A_{\theta(\mathcal{T}')}x + B_{\theta(\mathcal{T}')}u \in \mathcal{K}(\mathcal{T}')\} \quad (5.26)$$

Proposition 5.1. For any *non-overbranched* tree \mathcal{T} , $\mathcal{K}(\mathcal{T}) \supset \mathbb{T}^{[z(\mathcal{T})]}$.

Proof: Let us prove the proposition by induction. First, proposition is trivial for depth-1 trees, by definition. Now, assume that it is true for trees \mathcal{T}' such that $1 \leq \text{depth}(\mathcal{T}') \leq N$, for an arbitrary N .

Consider a depth- $(N + 1)$ tree \mathcal{T} . Then, all trees in $\text{trim}(\mathcal{T})$ are of depth at most N , so $\mathcal{K}(\mathcal{T}')$ at the end of (5.26) is larger or equal than $\mathbb{T}^{[z(\mathcal{T}')]}$ by the induction assumption. We will show that the terminal controller $u = K(x, z(\mathcal{T}))$ fulfils the conditions in (5.26) if $x \in \mathbb{T}^{[z(\mathcal{T})]}$.

Indeed, for such x , (5.10) ensures $A_i x + B_i K(x, z(\mathcal{T})) \in \cap_{j \in \text{adm}_\delta(i, z(\mathcal{T}))} \mathbb{T}^{[j]}$ for any $i \in \text{adm}_e(z(\mathcal{T}))$ from (5.10). As the tree is non-overbranched, $\theta(\mathcal{T}') \in \text{adm}_e(z(\mathcal{T}))$ and $z(\mathcal{T}') \in \text{adm}_\delta(\theta(\mathcal{T}'), z(\mathcal{T}))$. Hence,

$$\cap_{j \in \text{adm}_\delta(i, z(\mathcal{T}))} \mathbb{T}^{[j]} \subset \mathbb{T}^{[z(\mathcal{T}')] } \subset \mathbb{K}(\mathcal{T}')$$

Thus, $\mathcal{K}(\mathcal{T}) \supset \mathbb{T}^{[z(\mathcal{T})]}$ and, hence, the proposition is true for trees of depth $N + 1$. \square

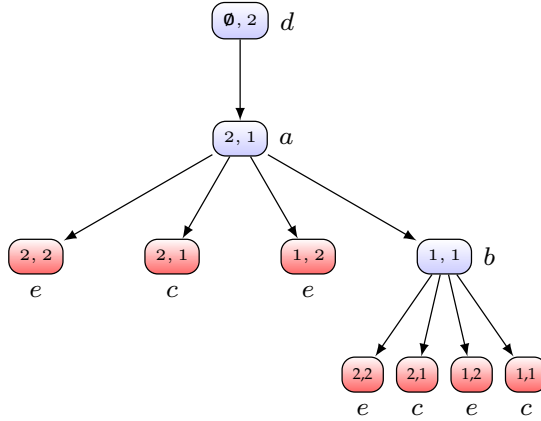


Figura 5-3: Scenario-tree \mathcal{T} with initial information $z = 2$.

Example 5.3. Consider the tree in Figure 5-3. Feasible sets from Definition 5.9 are depicted in Figure 5-4:

- The top figure presents the terminal set $\mathbb{T}^{[1]}$ (in blue colour, corresponding to nodes labelled as 'c') and the feasible sets of the node labelled as 'a' (red), and that of the node labelled as 'b' (pink).
- The bottom figure represents the terminal set $\mathbb{T}^{[2]}$ (green, nodes with label 'e'), and the feasible set of the root node (red, label 'd').

As expected from Proposition 5.1, feasible sets of parent nodes with the same information are larger than those of corresponding child nodes (and, of course, larger than the corresponding terminal set).

Theorem 5.4. Given a scenario \mathcal{T} , the MPC problem stated in Definition 5.8 is feasible for initial state x_0 and information $z_0 = z(\mathcal{T})$ if $x_0 \in \mathcal{K}(\mathcal{T})$.

Proof: Indeed, if the theorem were correct for sub-trees $\mathcal{T}' \in \text{trim}(\mathcal{T})$, then the problem would be feasible if $x_0 \in \mathcal{K}(\mathcal{T})$ by (5.26). For trees with a single leaf node, the only constraint is (5.25c), and the theorem is trivially true as the feasible set is defined to be the terminal set. So, the assertion is true for the whole tree, travelling from leaf nodes to root. \square

If conditions in Theorem 5.4 hold, we will say that \mathcal{T} is a *feasible scenario* for state x_0 and information z_0 .

Proposition 5.2. If \mathcal{T} is a feasible scenario for given (x_0, z_0) and any u_0 in the set

$$\bar{U}(x_0, \mathcal{T}) := \left\{ u \in \mathbb{U}^{[z(\mathcal{T})]} : A_{\theta(\mathcal{T}')}x_0 + B_{\theta(\mathcal{T}')}u \in \mathcal{K}(\mathcal{T}'), \forall \mathcal{T}' \in \text{trim}(\mathcal{T}) \right\} \quad (5.27)$$

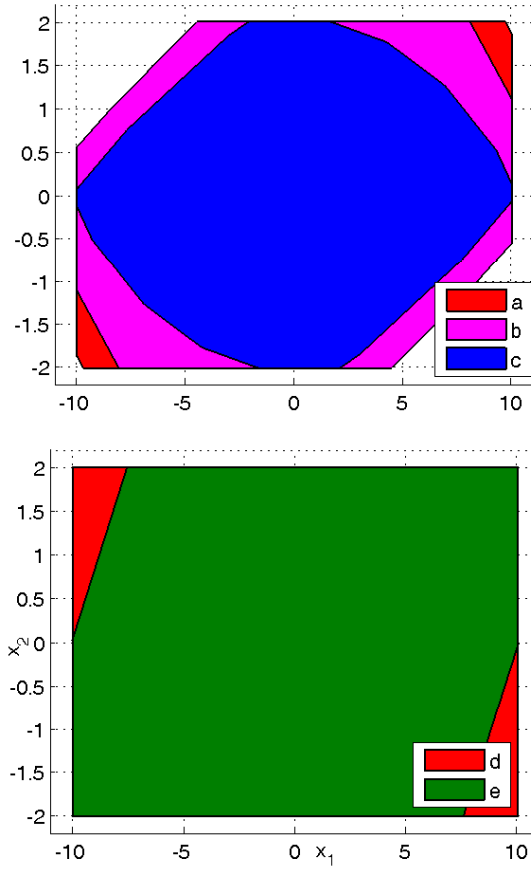


Figura 5-4: Feasible sets for Example 5.3: upper figure, nodes with $z = 1$; bottom figure, nodes with $z = 2$.

is applied, then the tree in $\mathcal{T}' \in \text{trim}(\mathcal{T})$ such that $\delta(\mathcal{T}') = \{\theta, z_+\}$ will be feasible for state $A_\theta x_0 + B_\theta u_0$ and information z_+ . Additionally, if \mathcal{T} is probabilistic-complete, all $\mathcal{T}' \in \text{trim}(\mathcal{T})$ such that $z(\mathcal{T}') = z_1$ will be feasible for $(x_1 = A_{\theta_0} x_0 + B_{\theta_0} u_0, z_1)$, for any admissible θ_0, z_1 according to the transition model in Section 5.2.1.

Proof: The definition of feasible sets (5.26) ensures that $\overline{U}(x_0, \mathcal{T})$ is not empty; thus, if mode and information are equal to $\delta(\mathcal{T}')$ such tree will be feasible if $u_0 \in \overline{U}(x_0, \mathcal{T})$ is applied. If \mathcal{T} is probabilistic-complete, for all possible current modes $\theta \in \text{adm}_e(z_0)$, the successor state lies in the intersection of all feasible sets for all possible successor information $z \in \text{adm}_\delta(\theta, z_0)$; of course the actual θ_0 (yielding successor x_1) and z_1 will be among them. \square

Enlarging the tree in order to explore longer horizons will improve the feasible set (or, at least, keep it unchanged):

Proposition 5.3. *For any strictly-complete tree \mathcal{T} , if $x \in \mathcal{K}(\mathcal{T})$, then $x \in \mathcal{K}(\mathcal{T}')$ for any non-overbranched tree \mathcal{T}' such that $\mathcal{T} \subset \mathcal{T}'$.*

Proof: Consider three sets of ‘new’, ‘formerly leaf’ and ‘unchanged’ nodes (Figure 5-5), respectively defined as:

$$S_1 := \{\mathcal{N} : \mathcal{N} \in \mathcal{T}', \mathcal{N} \notin \mathcal{T}\} \quad (5.28a)$$

$$S_2 := \{\mathcal{N} : \mathcal{N} \notin \text{leaf}(\mathcal{T}'), \mathcal{N} \in \text{leaf}(\mathcal{T})\} \quad (5.28b)$$

$$S_3 := \{\mathcal{N} : \mathcal{N} \notin S_2, \mathcal{N} \in \mathcal{T}\} \quad (5.28c)$$

Note that nodes in the original (smaller) \mathcal{T} are $S_3 \cup S_2$. Note also that, if $\mathcal{N} \in S_3$ then $\text{succ}(\mathcal{N})$ will be the same on both \mathcal{T} and \mathcal{T}' , because all admissible successors modes were already branched out in \mathcal{T} (due to the probabilistic-complete assumption).

Consider $\mathcal{N} \in S_2$. As \mathcal{N} is a leaf node on \mathcal{T} , $\mathcal{E}_{\mathcal{T}}(\mathcal{N})$ has a single node, so $\mathcal{K}(\mathcal{E}_{\mathcal{T}}(\mathcal{N})) = \mathbb{T}^{[z(\mathcal{N})]}$ by definition. As \mathcal{N} is a non-leaf node on \mathcal{T}' , the sub-tree $\mathcal{E}_{\mathcal{T}'}(\mathcal{N})$ will have a feasible set larger than $\mathbb{T}^{[z(\mathcal{N})]}$ by Proposition 5.1, as \mathcal{T}' is non-overbranched. Hence, for leaf nodes of \mathcal{T} such that $\mathcal{N} \in S_2$, $\mathcal{K}(\mathcal{E}_{\mathcal{T}'}(\mathcal{N})) \supset \mathcal{K}(\mathcal{E}_{\mathcal{T}}(\mathcal{N}))$.

Now, for the rest of leaf nodes of \mathcal{T} , i.e., $\mathcal{N} \in \text{leaf}(\mathcal{T}) \sim S_2$ we have $\mathcal{K}(\mathcal{E}_{\mathcal{T}'}(\mathcal{N})) \equiv \mathcal{K}(\mathcal{E}_{\mathcal{T}}(\mathcal{N})) \equiv \mathbb{T}^{[z(\mathcal{N})]}$.

So, for any $\mathcal{N} \in \text{leaf}(\mathcal{T})$, we have $\mathcal{K}(\mathcal{E}_{\mathcal{T}'}(\mathcal{N})) \supset \mathcal{K}(\mathcal{E}_{\mathcal{T}}(\mathcal{N}))$. Therefore, completing upwards the recursive evaluation of (5.26) until the root node is reached will yield a larger feasible set $\mathbb{K}(\mathcal{T}')$. \square

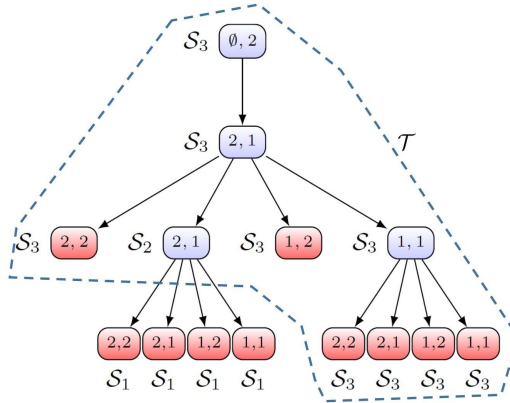


Figura 5-5: Example of the sets of nodes in proof of Proposition 5.3: let the larger \mathcal{T}' be the whole tree; let the smaller tree \mathcal{T} be the tree in Figure 5-3, formed only with nodes labelled \mathcal{S}_2 and \mathcal{S}_3 (nodes inside dashed-line).

5.5.1. Recursive Feasibility in Receding-horizon Implementation

A controller computes $u(x_k, z_k)$ upon knowledge of information z_k and state x_k at each instant. The control decision will be made assuming a different tree (i.e., scenario) for each of the possible values of z_k :

Definition 5.10. A *policy* \mathcal{P} is a indexed collection of trees $\mathcal{P} = \{\mathcal{T}^{[1]}, \dots, \mathcal{T}^{[Z]}\}$ so that optimisation over tree $\mathcal{T}^{[z]}$ would be carried out if measured information were z . A tree \mathcal{T}' will be a *sub-scenario* of a policy \mathcal{P} , denoted as $\mathcal{T}' \subset \mathcal{P}$, if $\mathcal{T}' \subset \mathcal{T}^{[z(\mathcal{T}')]}$. A policy \mathcal{P} is *probabilistic-complete* if $\mathcal{T}^{[j]} \neq \emptyset$ for all j , and all $\mathcal{T}^{[j]}$ are themselves probabilistic-complete. A non-probabilistic-complete policy will be denoted as 'risky'. A policy \mathcal{P} will be denoted as *non-trivial* if all its component trees are non-trivial.

Receding-horizon implementation of scenario-MPC will be understood as applying the same policy at all times.

As the goal of this chapter is to guarantee stability and constraint satisfaction, risky policies will not be considered. A non-predictive approach to handle risky policies with some reliability bounds will be presented in next chapter. Additionally, scenarios with a single-node tree will be associated to applying the terminal controller: the control problem in Definition 5.8 will be assumed to be posed in situations excluding that trivial one.

Definition 5.11. A policy \mathcal{P} is *trim-contained* if it is non-trivial and $\mathcal{T}' \subset \mathcal{P}$ (understood as in Definition 5.10) for all $\mathcal{T}' \in \text{trim}(\mathcal{T}^{[z]})$, for all $z \in \mathcal{Z}$.

Theorem 5.5. *A trim-contained policy \mathcal{P} is recursively feasible in receding horizon implementation if initial (x_0, z_0) are such that $x_0 \in \mathcal{K}(\mathcal{T}^{[z_0]})$ and $u_0 \in \bar{U}(x_0, \mathcal{T}^{[z_0]})$.*

Proof: Consider $\mathcal{T}' \in \text{trim}(\mathcal{T}^{[z_0]})$. By Proposition 5.2, if any $u_0 \in \bar{U}(x_0, \mathcal{T}^{[z_0]})$ is applied, then all \mathcal{T}' such that $z(\mathcal{T}') = z_1$ will be feasible for next (x_1, z_1) because $\mathcal{T}^{[z_0]}$ is probabilistic-complete. As \mathcal{T}' is non-overbranched and $\mathcal{T}^{[z_1]}$ is strictly-complete and, by assumption $\mathcal{T}' \subset \mathcal{T}^{[z_1]}$, then Proposition 5.3 can be applied: the feasible set of $\mathcal{T}^{[z(\mathcal{T}')]}$ will be larger than that of \mathcal{T}' , so it will be, too, feasible for (x_1, z_1) . This will, hence, recursively occur for all future time. \square

5.6. Stability

In all results in this section, we will assume terminal ingredients fulfilling Theorem 5.2 are suitably incorporated.

Proposition 5.4. *For any strictly-complete tree, if $x \in \mathbb{T}^{[z(\mathcal{T})]}$ then $J^*(\mathcal{T}, x) \leq \mathcal{V}(x, z(\mathcal{T}))$.*

Proof: Let us show that, for $x \in \mathbb{T}^{[z(\mathcal{T})]}$, setting all the node control decision variables as $u(\mathcal{N}) = K(x(\mathcal{N}), z(\mathcal{N}))$ results in a feasible predictive control problem, and $J(\mathcal{T}, x, \mathcal{U}^K(\mathcal{T})) \leq \mathcal{V}(x, z(\mathcal{T}))$, where $\mathcal{U}^K(\mathcal{T})$ denotes the referred choice of decision variables.

The above assertion can be proved from (a) the fact that the terminal controller steers successor states to suitable terminal sets in \mathbf{T} and, (b) all tree nodes have a feasible set larger than or equal to such terminal sets because \mathcal{T} is non-overbranched (Proposition 5.1). Indeed, these two ideas prove that the terminal control law fulfils $K(x(\mathcal{N}), z(\mathcal{N})) \in \bar{U}(x(\mathcal{N}), \mathcal{E}_{\mathcal{T}}(\mathcal{N}))$, with a reasoning parallel to the proof of the mentioned proposition.

The fact that $J(\mathcal{T}, x, \mathcal{U}^K(\mathcal{T})) \leq \mathcal{V}(x, z(\mathcal{T}))$ can be proved by induction. For depth-1 trees, it does hold because such $J(\mathcal{T})$ is defined equal to \mathcal{V} in (5.23).

Assuming the assertion to be true for all trees of depth up to N , consider a tree \mathcal{T} of depth $N + 1$. Replacing the terminal control law K in (5.24), considering the right-most double summation in such expression, by induction assumption we can assert for each \mathcal{T}' that

$$\begin{aligned} J(\mathcal{T}', A_{\theta(\mathcal{T}')}x + B_{\theta(\mathcal{T}')}K(x, z(\mathcal{T})), \mathcal{U}(\mathcal{T}')) \\ \leq \mathcal{V}(A_{\theta(\mathcal{T}')}x + B_{\theta(\mathcal{T}')}K(x, z(\mathcal{T})), z(\mathcal{T}')) \quad (5.29) \end{aligned}$$

so, we can bound:

$$J(\mathcal{T}, x, \mathcal{U}^K(\mathcal{T})) \leq l(x, z(\mathcal{T}), K(x, z(\mathcal{T}))) + E_+(\mathcal{V}, x, K(x, z(\mathcal{T})), z(\mathcal{T})) \leq \mathcal{V}(x, z(\mathcal{T})) \quad (5.30)$$

where the fact that the tree is strictly complete has been used to replace the summation over all trimmed trees in (5.24) with the summation over all admissible mode-information transitions in (5.7b). Finally, (5.8) bounds the $(l + E_+)$ expression with $\mathcal{V}(x, z(\mathcal{T}))$.

The last step of the proof follows from the fact that further optimisation will improve over the cost of the terminal control law:

$$J^*(\mathcal{T}, x) = \min_{\mathcal{U}(\mathcal{T})} J(\mathcal{T}, x, \mathcal{U}(\mathcal{T})) \leq J(\mathcal{T}, x, \mathcal{U}^K(\mathcal{T}))$$

□

Proposition 5.5. *Given two strictly-complete scenarios \mathcal{T} and \mathcal{T}' , if $\mathcal{T} \subset \mathcal{T}'$, then solution to the scenario predictive control problem in Definition 5.8 over \mathcal{T}' is feasible and $J^*(\mathcal{T}', x) \leq J^*(\mathcal{T}, x)$ for all $x \in \mathcal{K}(\mathcal{T})$.*

Proof: Feasibility is ensured by Proposition 5.3 as $\mathcal{K}(\mathcal{T}) \subset \mathcal{K}(\mathcal{T}')$. Now, consider the three sets of nodes (5.28a)–(5.28c). Using Proposition 5.4, there exists a set of decision variables associated to the trees emanating from nodes in \mathcal{S}_2 (in \mathcal{T}') such that the optimal cost on such sub-trees will be lower than or equal to the terminal cost \mathcal{V} for node state in the terminal sets (which was required in \mathcal{T}). Analogously to the proof of Proposition 5.3, propagating upwards $J^*(\mathcal{T}', x) \leq J^*(\mathcal{T}, x)$ is proven. In fact, as \mathcal{T}' relaxes the terminal constraints in formerly leaf nodes \mathcal{S}_2 , further improvements might be obtained. □

Theorem 5.6. *For any trim-contained policy \mathcal{P} , receding horizon implementation of the MPC problem in Definition 5.8 is recursively feasible and mean square stable if initial conditions (x, z) are such that $x \in \mathbb{K}(\mathcal{T}^{[z]})$.*

Proof: Given initial (x, z) such that $x \in \mathbb{K}(\mathcal{T}^{[z]})$, recursive feasibility comes from Theorem 5.5. Also, as the policy is trim-contained, we have $\mathcal{T}' \subset \mathcal{T}^{[z(\mathcal{T}')]}$; thus, we can assert from Proposition 5.5:

$$J^*(\mathcal{T}', x) \geq J^*(\mathcal{T}^{[z(\mathcal{T}')]}, x) \quad \forall \mathcal{T}' \in \text{trim}(\mathcal{T}^{[z]}) \quad (5.31)$$

because longer trees $(\mathcal{T}^{[z(\mathcal{T}')]})$ have lower cost bound than shorter ones (\mathcal{T}') .

Now, from (5.24), we can assert the following equality:

$$\begin{aligned}
J^*(\mathcal{T}^{[z]}, x) &= \\
l(x, z, u^*(\mathcal{T}^{[z]})) &+ \sum_{\mathcal{T}' \in \text{trim}(\mathcal{T}^{[z]})} \pi_{z(\mathcal{T}')\theta(\mathcal{T}')z(\mathcal{T})} e_{\theta(\mathcal{T}')z(\mathcal{T})} J^*(\mathcal{T}', A_{\theta(\mathcal{T}')}x + B_{\theta(\mathcal{T}')}u^*(\mathcal{T}^{[z]})) \\
&\geq l(x, z, u^*(\mathcal{T}^{[z]})) + \\
&\sum_{\mathcal{T}' \in \text{trim}(\mathcal{T}^{[z]})} \pi_{z(\mathcal{T}')\theta(\mathcal{T}')z(\mathcal{T})} e_{\theta(\mathcal{T}')z(\mathcal{T})} J^*(\mathcal{T}^{[z(\mathcal{T}')]}, A_{\theta(\mathcal{T}')}x + B_{\theta(\mathcal{T}')}u^*(\mathcal{T}^{[z]})) \quad (5.32)
\end{aligned}$$

and (5.31) allows writing the inequality in the expression (5.32). As $\mathcal{T}^{[z]}$ is probabilistic-complete (so all admissible transitions are present in the summations), introducing notation $\mathcal{J}(x, z) := J^*(\mathcal{T}^{[z]}, x)$, we can write (5.32) in the equivalent form:

$$\mathcal{J}(x, z) - E_+(\mathcal{J}, x, u^*(\mathcal{T}^{[z]}), z) \geq l(x, z, u^*(\mathcal{T}^{[z]})) \quad (5.33)$$

so Theorem 5.1 proves mean-square stability if control action $u^*(\mathcal{T}^{[z]})$ is applied in receding horizon. \square

Note that, the mode-dependent predictive controllers in (Patrinos *et al.* 2014) with a full tree are a particular case of our approach, both on the tree structure and on the probabilistic transition model: Theorem 5.6 would reduce to (Patrinos *et al.* 2014, Theorem 24) under their assumptions.

5.6.1. Scenario Generation

In order to generate scenarios fulfilling the above-discussed feasibility and stability properties, it is suggested to, first, generate a set of scenarios (policy) based in, for instance, the probabilistic ranking in (Bernardini and Bemporad 2012), with at least two nodes per tree. Then, in order to make it probabilistic-complete, if there exist non-leaf nodes where (5.17) is not fulfilled, the tree can be easily completed adding the missing admissible child nodes as new leaf nodes; let us denote such operation, carried out over all trees in a policy \mathcal{P} , as *complete*(\mathcal{P}).

Now, $\text{trim}(\mathcal{P})$ will denote the collection of trees arising from trimming each of the policy's trees, in no particular order. Given a policy \mathcal{P} and an arbitrary collection of trees $\{\mathcal{T}_1, \dots, \mathcal{T}_q\}$ for some q , the union $\bar{\mathcal{P}} = \mathcal{P} \cup \{\mathcal{T}_1, \dots, \mathcal{T}_q\}$

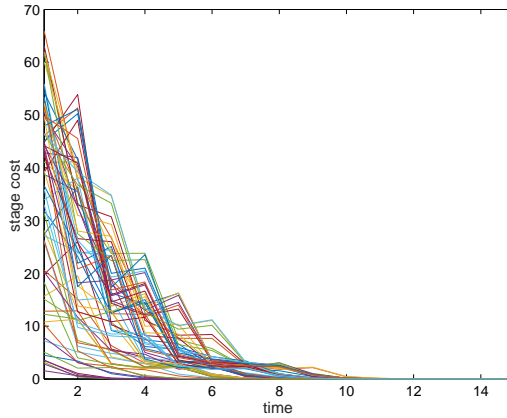


Figura 5-6: Simulation of the scenario-predictive control for trees in figures 5-2 and 5-3.

is defined as the policy $\bar{\mathcal{P}} = \{\bar{\mathcal{T}}^{[1]}, \dots, \bar{\mathcal{T}}^{[Z]}\}$ in which:

$$\bar{\mathcal{T}}^{[z]} := \mathcal{T}^{[z]} \cup \left(\bigcup_{\substack{1 \leq j \leq q \\ z(\mathcal{T}_j) = z}} \mathcal{T}_j \right) \quad (5.34)$$

With these definitions, iterating $\mathcal{P} = \text{complete}(\mathcal{P} \cup \text{trim}(\mathcal{P}))$ until convergence will find a trim-constrained policy fulfilling Theorem 5.6. The above iterations will converge in a finite number of steps, as the maximum depth of trees in \mathcal{P} keeps constant between iterations: for sure they will stop before or, in a worst-case situation, at the moment of reaching full trees.

Example 5.4. From all considerations on the terminal cost, feasible sets, and tree properties (trim-contained) we can ensure that the scenario predictive control problem on the policy given by trees in Figure 5-2, for initial information $z = 1$, and Figure 5-3, for initial information $z = 2$, will be recursively feasible and mean-square stable in receding horizon. The actual setting in terms of quadratic programming is analogous to, for instance, (Bernardini and Bemporad 2012, Patrinos et al. 2014). 100 simulations of the evolution of the quadratic stage cost $l(x, z, u)$ appears on Figure 5-6, for random feasible initial conditions with the model in Example 5.1, tending to zero as expected.

5.7. Conclusions of this Chapter

This chapter has presented a set of terminal ingredients and scenario properties so that scenario predictive control approaches in prior literature can be

proved mean-square stable and receding horizon recursively feasible without the need of enforcing additional Lyapunov level-set constraints. The considered scenarios contemplate a probabilistic model in which a set of different modes (different linear models of the controlled process) switch according to some 'information' variables assumed measurable and governed by a pair of observation/transition probability matrices. In this way, classical results on stability and feasibility of standard predictive control (and associated terminal and feasible sets), as well as those on MJLS, are generalised to scenario-based optimisation under the proposed probabilistic transition model.

Capítulo 6

Reliable MPC for Markov-jump Linear Systems

Nota sobre el capítulo: En los dos anteriores capítulos se formuló el control predictivo basado en escenarios para MJLS asegurando estabilidad y factibilidad para todo tiempo futuro, asumiendo que los árboles de escenarios son completo en el sentido probabilístico (la probabilidad de los nodos sucesores de cada nodo del árbol, excepto los terminales, suman uno). Este tipo de escenario debe ser entendido como no arriesgado, puesto que se tiene 100 % de probabilidad de que el problema de optimización sea factible para árboles que cumplan las condiciones presentes en los capítulos anteriores. Además, se tratan con sistemas inestables en tiempo discreto. El contenido de este capítulo está basado en el trabajo:

- M. Hernández-Mejías, A. Sala, C. Ariño, A. Querol. *Reliable fault-tolerant predictive control for Markov-jump linear systems* . IEEE Conference on Control Application (CCA), 2014. Pp. 1831-1836. Antibes, France.
DOI: 10.1109/CCA.2014.6981579.

Este trabajo corresponde a un primer enfoque en cuanto a la utilización de escenarios arriesgados en MJLS, eliminando secuencias de poca probabilidad, que se mejorará en capítulos posteriores.

Abstract: This chapter considers model-predictive control for discrete-time Markov Jump Linear Systems subject to constraints on state and control variables. Hitting state constraints is a situation which should be avoided, but it is impossible to do that with total certainty when not all possible models are stable, except in trivial initial states (e.g., the origin). Hence, this chapter discusses how to ensure a certain degree of reliability (low likelihood of hitting constraints) in the above unstable-mode case if there exists a terminal controller which, on average, renders the system stochastically stable.

6.1. Introduction

As above discussed *Model Predictive Control (MPC)* is a widely used technique, where the control input is obtained by solving a finite horizon constrained open-loop optimal control problem, (Mayne *et al.* 2000), implemented in a receding horizon policy. One of the main issue in MPC consists in guaranteeing closed-loop stability for the controlled systems, so in the linear case, constraint satisfaction can be guaranteed if the initial state is contained inside a robust control invariant set (Kerrigan 2000).

Chapter 3 consider the extension of classical MPC to MJLS, providing a methodology to design a control law with stochastic stability and guarantee of constraint satisfaction based on extending the usual MPC quadratic programming approach. However, such constraint satisfaction guarantee only yields non-trivial results when all closed-loop modes are stable. For instance, the results presented in Chapter 3 cannot handle the case of unstable systems subject to sporadic loss of control. Alternative options for re-configurable MPC appear in (Hartley and Maciejowski 2014) for redundantly actuated systems.

In this chapter a first approach in order to solve the above problem is presented, extending the results in Chapter 3 to ‘sporadically unstable’ cases, evaluating the reliability of a chosen ‘likely-terminal’ set and, then, proposing it for use in a predictive control strategy which should work ‘most of the times’ when starting from certain ‘likely-feasible’ sets.

6.2. Preliminaries

Consider a Markov-jump linear system (MJLS):

$$x_{k+1} = A_{\theta_k} x_k + B_{\theta_k} u_k \quad (6.1)$$

where $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^m$ are state and input sequences, $\bar{\mathcal{A}} = \{A_1, \dots, A_M\}$ and $\bar{\mathcal{B}} = \{B_1, \dots, B_M\}$ are so-called mode-dependent model matrices and the mode variable $\theta(k) \in \mathcal{M}$, $\mathcal{M} = \{1, \dots, M\}$ is a discrete-time Markov chain with transition probability matrix $\mathbb{P} = [\pi_{ji}]$, with $\pi_{ji} = p(\theta(k+1) = j | \theta(k) = i)$.

We will assume that at time k , the mode $\theta(k)$ is known, as well as the plant state x_k and the Markov chain probability matrix \mathbb{P} .

Similar to Chapter 3, there are M^{N+1} possible realizations of θ_{k+t} from $t = 0$ to $t = N$, $N \geq 0$. An index variable $l \in \mathcal{L}$, $\mathcal{L} = \{1, 2, \dots, M^{N+1}\}$ will be used to single out an individual realization, using a suitable ordering. Such a realization will be denoted by $\theta^l \in \mathcal{M}^{N+1}$; the element of θ^l at position j will be represented by θ_j^l . Also, mode-dependent matrices associated to such element will be denoted by the shorthand $A_j^{[l]} = A_{\theta_j^l}$, $B_j^{[l]} = B_{\theta_j^l}$.

Prediction Model. The prediction model used in this chapter is the same of the Section 3.3.1, *i.e.*:

$$x_{k+t}^{[l]} = \prod_{j=k}^{k+t-1} A_j^{[l]} x_k + \sum_{d=k}^{k+t-1} \left(\prod_{j=d+1}^{k+t-1} A_j^{[l]} \right) B_d^{[l]} u_d^{[l]}, \quad (6.2)$$

6.2.1. Problem Statement

It is assumed that hitting state constraints in the control of the MJLS will represent an undesirable situation to be avoided; such a situation will be considered a non-recoverable failure.

Then, in MPC of systems whose terminal control law cannot stabilize all modes, the actual sets in which there is 100 % guarantee of stochastic stability and not violating the state constraints in the future have zero volume¹⁹. In fact, the sets may even be reduced to a single point (the origin).

However, as the terminal controller is stochastically stable, that means that the actual likelihood of the state being outside any compact set containing the origin tends to zero as time increases. Indeed, stochastic stability means that the unstable modes have low likelihood so its ‘expansion’ is counteracted by more frequent ‘contractions’ from stable modes.

Then, instead of having absolute guarantee that no violation the state constraints in future time, we will try to ensure that the likelihood of hitting such constraints is low enough to have an MPC scheme with ‘acceptable’ reliability

¹⁹Indeed, as the always-unstable-mode sequence is possible, the projection of the terminal/feasible sets on its unstable eigenvectors subspace must be zero, otherwise it could expand forever.

in practice.

The aim of this chapter is, hence, assessing the reliability of a chosen ‘likely-terminal’ set (with non-zero volume) and, if its probability of violating constraints is deemed low enough, proposing it for later use in a predictive control strategy which should work ‘most of the times’ when starting from some ‘likely-feasible’ sets.

6.3. Reliability analysis of a stochastically stable Markov-jump system

Stochastic stability of a Markov process implies that, on average, the behaviour is ‘contractive’. That, subsequently, means that starting close enough to the origin, the likelihood of the state reaching a particular constraint tends to zero as time increases. In particular, the smaller the set of initial conditions, the lower the likelihood of having hit the constraints up to a particular time. Indeed, it will require a greater deal of expansion which is more unlikely given that in the long run, the average behaviour contracts.

Based on the above, given an arbitrary model²⁰:

$$x_{k+1} = A_{\theta_k} x_k \quad (6.3)$$

and a constraint set $\Omega_{max} \subset \mathbb{X}$, the objective of this section is computing a worst-case bound of the ‘probability of failure at H or less steps’ (reaching past the boundary of Ω_{max}) if initial conditions are at a particular set Ω (assumed given *a priori*). It is a worst-case bound, because if one of the images of any vertex of Ω hits the constraints, the whole mode sequence is considered to be leading to failure.

Specifically, given an horizon H , an outer set Ω_{max} , a central candidate set $\Omega \subset \Omega_{max}$, and an inner set $\Omega_{in} \subset \Omega$, we will produce an algorithm to compute the following probabilities:

- **probability of failure (\mathcal{P}_1):** The cumulative probability of all mode sequences for which there exists a vertex in Ω so that a trajectory from it leaves the interior of Ω_{max} in H or less steps.
- **probability of contraction (success) (\mathcal{P}_2):** The cumulative probability of all mode sequences which do not leave Ω_{max} and map all vertices of Ω

²⁰The model might be an open-loop one or, as later proposed, a closed-loop –terminal controller– set-up

into Ω_{in} , after exactly H steps.

- **probability of uncertain status (\mathcal{P}_3):** One minus the other two probabilities, represents the cumulative probability of all sequences for which no vertex leaves Ω_{max} in H or less steps, but there is at least one vertex which did not enter Ω_{in} .

Algorithm 4 is used to compute the above-described probabilities. In the referred algorithm, the input *modelsini* is the list of matrices of the model (6.3), jointly with its associated initial probabilities²¹. Then, the *children* function generates, given a list of (M : matrix, *prob*: probability) pairs, a new list of (matrix, probability) pairs, obtained by multiplying every matrix in the input list by every possible mode matrix in (6.3), and the associated probability by the corresponding transition probability from Markov matrix \mathbb{P} .

For instance, in Figure 6-1 it can be seen that a two-mode system has two children for each one of the considered matrices at previous iteration. At $t = 1$ four children are obtained, two of them from A_1 and the other two from A_2 at $t = 0$. Once *childrenList* is computed, after suitable vertex checking in the algorithm, it might result that the last mode sequence yields some vertex $\mathbb{V}(j)$ so $Image = A_2 A_2 \mathbb{V}(j) \notin \Omega_{max}$, therefore, for the next instant, this sequence's children will not be considered. Then, six children appear at next instant, and so on until the final horizon H is reached.

For reliable practical operation, the probability of failure (\mathcal{P}_1) must be low and the probability of contraction (\mathcal{P}_2) must be high enough. In a worst-case analysis, all uncertain-status points might eventually lead to failure (which, of course, is conservative). Note that the points forced inside Ω_{in} will have in the future a lower probability of failure, and a greater fraction of those points will enter in a subsequent smaller set if the computations were repeated because Ω_{in} is, by assumption, smaller than Ω .

6.3.1. Choice of high-likelihood terminal and feasible sets for MPC technique

When considering stability of MPC set-ups for MJLS, the above study must be carried out for a candidate 'likely-terminal' set with the Riccati stochastically-stable optimal control law (terminal controller). From an engineering point of view, a choice of the mentioned 'likely-terminal' set \mathbb{X}_f will be reliable enough

²¹user-defined, for instance equal to the stationary Markov probabilities, or equal to one if it is assumed that we know the starting mode perfectly at time of the set computation.

Algoritmo 4 Reliability analysis for polytopic sets

Inputs: $\Omega_{max}, \Omega, \Omega_{in}, H, modelsit, \mathbb{P}$ **Outputs:** $\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3$

```

1:  $\mathcal{P}_1 = \mathcal{P}_2 = \mathcal{P}_3 = 0;$ 
2:  $\mathbb{V} = vertices(\Omega), \quad modelsit = modelsit;$ 
3: para  $i = 0$  to  $H$  hacer
3:    $childrenList = children(modelsit, \mathbb{P});$ 
3:    $index = 0;$ 
4:   para  $j = 0$  to  $size(childrenList)$  hacer
4:     success=true;
5:     para  $k = 0$  to  $size(\mathbb{V})$  hacer
5:        $Image = childrenList(j).M * \mathbb{V}(k);$ 
6:       si not  $(Image \in \Omega_{max})$  entonces
6:          $\mathcal{P}_1 = \mathcal{P}_1 + childrenList(j).prob;$ 
6:         success=false;
6:         break;
7:       fin si
8:     fin para
9:     si success entonces
9:        $modelsit(index) = childrenList(j);$ 
9:        $index = index + 1;$ 
10:    fin si
11:  fin para
12: fin para
13: para  $i = 0$  to  $size(modelsit)$  hacer
13:  success=true;
14:  para  $j = 0$  to  $size(\mathbb{V})$  hacer
14:     $Image = modelsit(i).M * \mathbb{V}(j);$ 
15:    si not  $(Image \in \Omega_{in})$  entonces
15:    success=false;
15:    break;
16:  fin si
17:  fin para
18:  si success entonces
18:     $\mathcal{P}_2 = \mathcal{P}_2 + modelsit(i).prob;$ 
19:  fin si
20: fin para
21:  $\mathcal{P}_3 = 1 - (\mathcal{P}_1 + \mathcal{P}_2);$ 

```

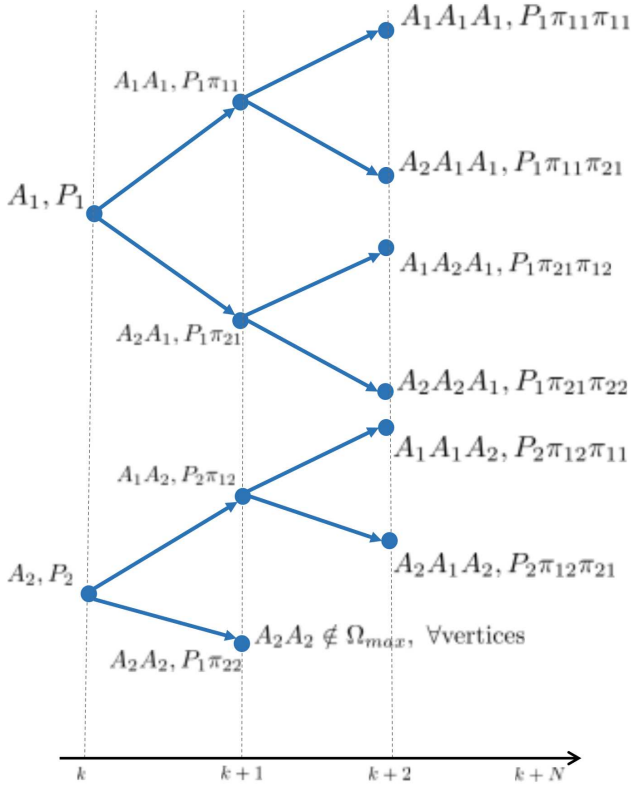


Figura 6-1: Possible mode sequences trajectories.

when the probability of failure (\mathcal{P}_1) is lower than a specified threshold and the probability of success (\mathcal{P}_2) is higher than another high-enough threshold.

Once a reliable enough terminal set \mathbb{X}_f has been chosen, starting with known mode, say i , if we take the prediction model until horizon N_1 , the constraints, for all possible mode sequences, are:

$$\begin{aligned}
 R_{\mathbb{X}}x + S_{\mathbb{X}} &\leq 0 \\
 R_{\mathbb{X}}(A_i x + B_i u_1^{[i]}) + S_{\mathbb{X}} &\leq 0 \\
 R_{\mathbb{X}}(A_1 A_i x + A_1 B_i u_1^{[i]} + B_1 u_2^{[1]}) + S_{\mathbb{X}} &\leq 0 \\
 R_{\mathbb{X}}(A_2 A_i x + A_2 B_i u_1^{[i]} + B_2 u_2^{[2]}) + S_{\mathbb{X}} &\leq 0 \\
 &\vdots \\
 R_{\mathbb{X}}(A_2 A_1 A_i x + A_2 A_1 B_i u_1^{[i]} + A_2 B_1 u_2^{[1]} + B_2 u_3^{[2]}) + S_{\mathbb{X}} &\leq 0 \\
 &\vdots \\
 R_{\mathbb{X}_f}(A_1 A_1 \dots A_i x + \dots) + S_{\mathbb{X}_f} &\leq 0 \\
 R_{\mathbb{X}_f}(A_M A_M \dots A_i x + \dots) + S_{\mathbb{X}_f} &\leq 0
 \end{aligned} \tag{6.4}$$

where R and S are given by Definition 1.19.

Let us define the N_1 -Step Robust Controllable Set $\mathcal{K}_{N_1}(\mathbb{X}, \mathbb{X}_f)$ as, the set of states in \mathbb{X} for which there exist a control law (the actions $u_k^{[l]}$, $l \in \mathcal{L}$, $k = 1, \dots, N_1$) fulfilling the constraints (6.4) for all i , such that a set \mathbb{X}_f is reached in at most N_1 steps. Obviously, it can be obtained by iterating a one-step fault-tolerant set given by Definition 3.2 and, using the Algorithm 1, but we express it here in multiple steps because a modification will be proposed below. Therefore, we could obtain the **Robust controllable set** at a multiple of N_1 steps via²² the iteration:

$$\mathcal{K}_{(p+1)*N_1}(\mathbb{X}, \mathbb{X}_f) = \mathcal{K}_{N_1}(\mathbb{X}, \mathcal{K}_{p*N_1}(\mathbb{X}, \mathbb{X}_f)) \tag{6.5}$$

starting with $\mathcal{K}_0(\mathbb{X}, \mathbb{X}_f) = \mathbb{X}_f$.

Initial conditions in such controllable set will guarantee that whatever the mode sequence, the likely-terminal set is reached in N_1 steps, so the reliability guarantees of such set hold, too. However, considering repeated unstable combinations might lead to a small \mathcal{K}_{p*N_1} (Figure 6-3 in the example section).

In a probabilistic case, if the probability of some of the mode sequences in (6.4) are very small, then we can compute larger N_1 -step controllable sets by removing unlikely sequences from the constraints (6.4); the resulting set, denoted as $\tilde{\mathcal{K}}_{N_1}(\mathbb{X}, \mathbb{X}_f)$, will yield a trajectory ending in the terminal set ‘in most cases’ when initial conditions are in $\tilde{\mathcal{K}}_{N_1}$.

Adapting the iterations above to an horizon multiple of N_1 , in this ‘likely

²²Again, this is not actually needed as everything can be obtained from the standard one-step set, but motivates later expressions.

controllable' case we have:

$$\tilde{\mathcal{K}}_{(p+1)*N_1}(\mathbb{X}, \mathbb{X}_f) = \tilde{\mathcal{K}}_{N_1}(\mathbb{X}, \tilde{\mathcal{K}}_{(p)*N_1}(\mathbb{X}, \mathbb{X}_f)) \quad (6.6)$$

initializing the iterations with $\tilde{\mathcal{K}}_0(\mathbb{X}, \mathbb{X}_f) = \mathbb{X}_f$.

So, with the above set-up for predictive control, the proposal achieves a controller which works in the majority of cases, as once the state is inside the inner sets it is progressively more unlikely to reach the constraints as time increases. Hence, the proposed approach may, in some cases, be able to design reliable enough MPC in practice for MJLS with unstable systems.

6.4. Numerical Example

Let us consider, as an example, a linearised inverted pendulum, with two states (position and speed) and one input. In the discrete-time Markov-jump model, two operating modes will be considered: Mode 1 means that the actuator is working properly; mode 2, the actuator fails with a complete loss of the control capabilities. So, the two models are:

$$[A_1|B_1] = \left[\begin{array}{cc|c} 1,0094 & 0,3009 & 0,0643 \\ 0,0632 & 1,0094 & 0,4299 \end{array} \right],$$

$$[A_2|B_2] = \left[\begin{array}{cc|c} 1,0094 & 0,3009 & 0 \\ 0,0632 & 1,0094 & 0 \end{array} \right]$$

Note that, the system in mode 2 is not stabilisable, *i.e.*:

$$B_2 = 0, \text{ eig}(A_2) = [1, 1474, 0, 8714]$$

The example has been chosen as the complete loss of an actuator in an unstable SISO system, because is a very relevant problem for this kind of process, so assessing its reliability (obviously, when there is a chance of recovering functionality in future samples) demonstrates the main features of our proposal. The relevant sets and reliability measures in stable systems and in those with redundancy in actuators would be larger.

The transition probability matrix between the two operating modes will be set to:

$$\mathbb{P} = \left[\begin{array}{cc} 0,95 & 0,9 \\ 0,05 & 0,1 \end{array} \right],$$

which models modes with a high likelihood of working properly, and sporadic faults which recover soon to normal behaviour (obviously, unstable systems without control during a large time interval would fail, this is the reason that the Markov matrix has been chosen this way). The Markov stationary probabilities are: $p(mode1) = 0,94736$ and $mode2 = p(0,0526)$.

Actuator constraints will be $-5 \leq u \leq 5$; state constraints will be: $-\pi/2 \leq x_1 \leq \pi/2$ and $-4 \leq x_2 \leq 4$, considering that the system has ultimately failed if state constraints are hit.

The weighting matrices in the MPC cost (3.20) will be set to: $Q = \mathbf{diag}(2, 2)$ and $R = 0,3$.

Recursively solving equation (2.9) the *terminal control law* u is obtained. The resulting terminal control gains are:

$$[K^1|K^2] = \left[\begin{array}{cc|cc} 1,48344 & 1,96784 & 0 & 0 \end{array} \right]$$

and the system with such terminal controller is, of course, stochastically stable as the Riccati equations have converged to a finite cost.

Once the terminal Riccati equations have been solved, suitable sets for predictive control implementation need to be set up.

If we keep all possibilities of mode transitions, the terminal invariant set \mathbb{X}_f discussed in Section 6.3.1 shrinks to the origin (intuitively expected, because a trajectory always in Mode 2 is unstable). The feasible sets for predictive control are also the origin. Hence, the empty sets obtained in Section 3.3.2 are basically useless.

Let us now apply Algorithm 4 to evaluate the reliability of the terminal controller in the following setting, see Figure 6-2:

- Outer set Ω_{max} is the input-output admissible set intersection of the state constraints and constraints arising from saturation of the terminal control law.
- Central set Ω is defined to be the scaling of Ω_{max} with a factor $0 < \lambda_1 < 1$.
- Inner set Ω_{in} is defined to be the scaling of Ω with factor $0 < \lambda_2 < 1$.
- Algorithm horizon $N = 10$.

The algorithm results, for $\lambda_1 = \lambda_2 = 0,7$ are shown in Table 6.1 below. The left numeric column presents the results when the probability of the mode at $t = 0$ is that from the Markov stationary state, the middle one when it is

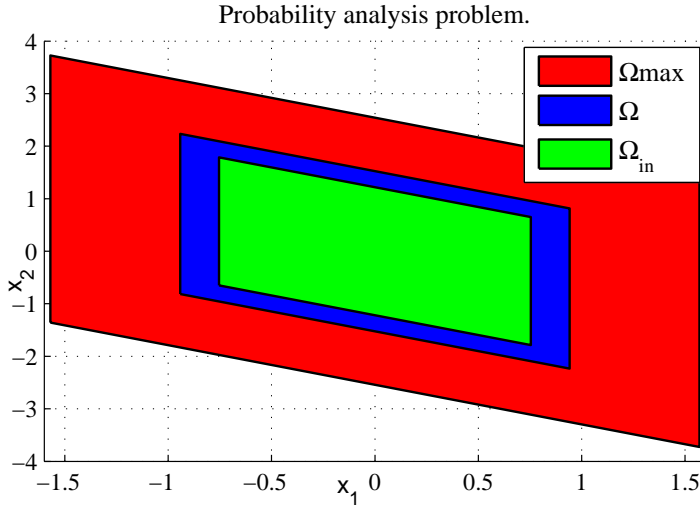


Figura 6-2: Ω_{max} is the maximal input-output admissible set, Ω represents a polytope with $zoom = 0,70$ of Ω_{max} $\Omega_{out} = \Omega_{max}$.

known that the initial mode is Mode 1, and the right one when starting from Mode 2.

Cuadro 6.1: Probability analysis ($\lambda_1 = \lambda_2 = 0,7$, $N = 10$)

	Stationary	Mode 1	Mode 2
\mathcal{P}_1	0.005263	$5,0 \cdot 10^{-06}$	0.1000
\mathcal{P}_2	0.994736	0.999994	0.89995
\mathcal{P}_3	$6,866 \cdot 10^{-7}$	$4,77250 \cdot 10^{-7}$	$4,4550 \cdot 10^{-6}$

The table shows that the probability of failure if initial state is in Ω and initial mode is Mode 2 is 10%. As we consider the probability of failure to be too high, another choice of $\lambda_1 = 0,6$, $\lambda_2 = 0,8$ has been fed to Algorithm 4; results appear in Table 6.2.

Cuadro 6.2: Probability analysis ($\lambda_1 = 0,6$, $\lambda_2 = 0,8$, $N = 10$)

	Stationary	Mode 1	Mode 2
\mathcal{P}_1	$5,264 \cdot 10^{-4}$	$5,00 \cdot 10^{-06}$	0.0100
\mathcal{P}_2	0.99994	0.999999	0.989999
\mathcal{P}_3	$5,312 \cdot 10^{-7}$	$4,9975 \cdot 10^{-7}$	$1,098 \cdot 10^{-6}$

The referred table shows that the worst-case probability of failure (starting

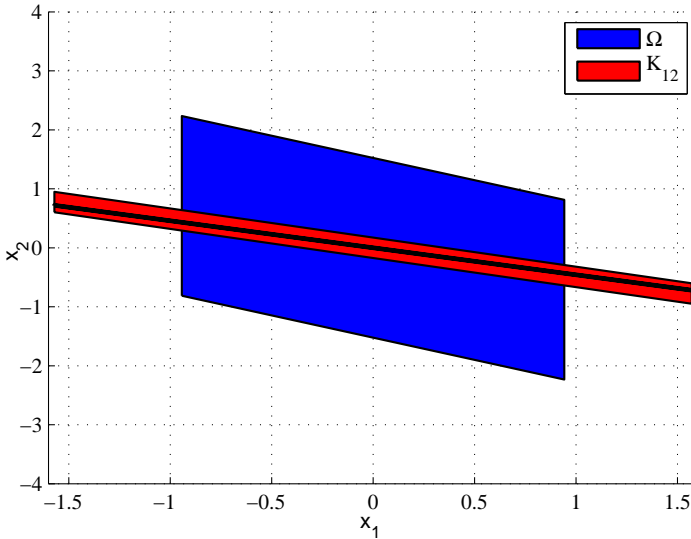


Figura 6-3: Ω is the terminal set \mathbb{X}_f and \mathcal{K}_{12} is the Fault-Tolerant Robust controllable set for all possible modes.

already in failed mode) is 1 %, and actually, such start situation in stationary state is quite unlikely (stationary prob. of failure is $5,3 \cdot 10^{-4}$), these choice of sets is deemed reliable enough, hence Ω it is proposed as terminal set \mathbb{X}_f . Indeed, the terminal controller, in steady state, will drive the state to Ω_{in} in 99.994 % of the cases.

Now, a full predictive control scheme with prediction horizon $N_1 = 12$ is set up using the above terminal set. For this scheme, the set of controllable initial states \mathcal{K}_{12} is computed for all possible mode transitions; the resulting sets are depicted in Figure 6-3.

Note that the feasible set does not include the whole terminal set: the feasible set shrinks in the direction associated to the unstable eigenvector (remaining a set basically aligned in the direction of the stable eigenvector, depicted with solid line in the figure).

However, if we compute the controllable set excluding them, following the ideas in section 6.3.1. If we take $N_1 = 3$, and remove sequences $\{122\}$ and $\{222\}$ we keep a 99 % likelihood of reaching the terminal set in 3 steps. Iterating four times, we obtain $\tilde{\mathcal{K}}_{12}$ (see Figure 6-4) as an initial condition set in which our predictive scheme will work reliably enough (99 % likelihood of entering a smaller $\tilde{\mathcal{K}}_9$, and so on).

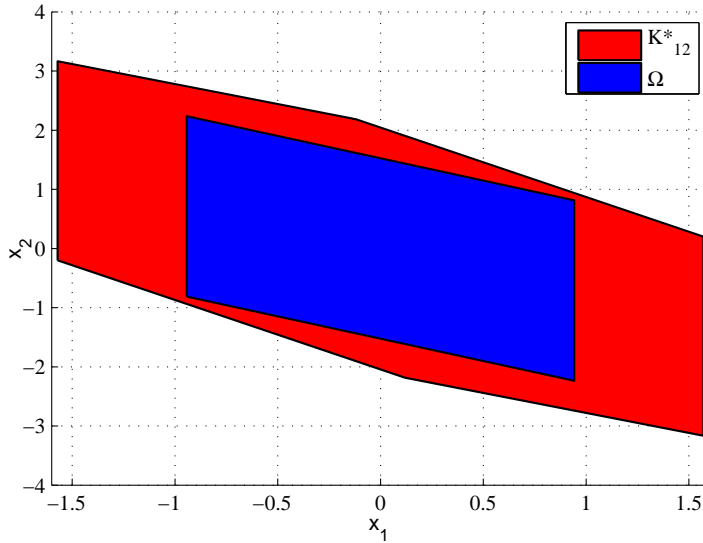


Figura 6-4: Ω is the terminal set \mathbb{X}_f and $K_{12}^* = \tilde{K}_{12}$, is the Fault-Tolerant Robust controllable set, removing unlikely sequences.

6.5. Conclusions of this Chapter

This chapter has presented a reliability analysis of model predictive control set-ups for systems with fault transitions modelled by a Markov chain with unstable modes. An algorithm computes the probability of hitting constraints and that of entering a smaller set after a prefixed horizon. This allows increasing the volume of the initial condition sets in which MPC strategies will work ‘in the majority of cases’ with respect to the robust ‘all cases’ sets.

Capítulo 7

Reliable Controllable Sets for Constrained Markov-Jump Linear Systems

Nota sobre el capítulo: En este capítulo, se presenta las contribuciones sobre conjuntos *confiables* basado en escenario para MJLS. Se consideran políticas arriesgadas en el árbol de escenario. Aquí, se obtienen conjuntos de estados que dependen de una realización en particular. Para estados perteneciente a estos conjuntos, puede garantizarse, con una cierta probabilidad menor del 100 %, que dichos estados alcancen a una región invariante en donde la convergencia del sistema al origen está garantizada. Este capítulo está basado en el trabajo:

- M. Hernández-Mejías, A. Sala, C. Ariño, A. Querol. *Reliable controllable sets for constrained Markov-Jump Linear Systems*. International Journal Of Robust and Nonlinear Control, 2016. Volumen 26, páginas 2075-2089.

Abstract: Robust λ -contractive sets have been proposed in previous literature for uncertain polytopic linear systems. It is well known that, if initial state is inside such sets, it is guaranteed to converge to the origin. This work presents the generalisation of such concepts to systems whose behaviour changes among different linear models with probability given by a Markov chain. We propose sequence-dependent sets and associated controllers which can ensure a reliability bound when initial conditions are outside the maximal λ -contractive set. Such reliability bound will be understood as the probability of actually reaching the origin from a given initial condition without violating constraints. As initial condition are further away from the origin, the likelihood of reaching the origin decreases.

7.1. Introduction

Control of constrained systems needs assessing the set of initial conditions which can be steered to the origin without violating such constraints during the transient. To that purpose, robust λ -contractive controllable sets have been proposed in priori literature for uncertain polytopic linear systems (Gilbert and Tan 1991, Blanchini 1994, Blanchini 1999, Kerrigan 2000).

Reliability analysis of a control strategy for constrained systems in presence of mode changes, is a problem of significant interest, motivating this work. Indeed, problems arise when initial conditions are outside the proven maximal sets. In such a case, the controller might stabilize the system without constraint violation only for *some* mode sequences, i.e., with probability lower than 1. A first approach has been presented in Chapter 6 for closed-loop unstable system.

This chapter focuses on the above issue, proposing an algorithm which determines sequence-dependent sets for which there exists a control law such that the system can be steered to the origin, but only for a *particular sequence* θ . If the algorithm converges, then, states outside such sets cannot be driven to the origin without violating constraints, whichever the mode sequence: the likelihood of reaching the origin from them is zero. Obviously, initial conditions in a robust λ -contractive controllable set have 100 % likelihood of reaching the origin. If the state lies in '*intermediate*' regions, this work proposes a control law which will steer the state to the origin with a certain probability bound p , to be denoted as *reliability bound*.

Two settings will be considered regarding the mode information available to the controller at the moment of computing the control action in on-line

operation. The first one will discuss the case in which the current mode is available; the second one will discuss controllers without such mode information. The former is a widely used assumption in control of MJLS (present in most of the above-cited works); the latter (no mode information) will yield controllers with lower reliability, as intuitively expected.

7.2. Preliminaries, Problem Statement

Consider a non-autonomous linear time-varying system, for $k \geq 0$:

$$x_{k+1} = A_{\theta_k} x_k + B_{\theta_k} u_k, \quad \theta_k \in \mathcal{M}, \quad (7.1)$$

The current mode θ_k can be either known at the instant of computing u_k or not. That will give rise to different control action possibilities later in this work.

7.2.1. Robust h -step Sets

In the case there is no mode information available to the controller, system (7.1) can be considered to be an uncertain linear system with polytopic uncertainty. In that context, let us assume the state must be constrained into a region Ω and that the control action must, too, belong to a known set \mathbb{U} .

The following definition is well known:

Definition 7.1 (Robust 1-step set (Blanchini 1994, Kerrigan 2000)).

$$Q(\mathcal{C}) := \{x \in \mathbb{R}^n : \exists u \in \mathbb{U} \text{ such that } A_i x + B_i u \in \mathcal{C} \forall i \in \mathcal{M}\} \quad (7.2)$$

With the above definition, following (Kerrigan 2000), given a contraction rate $0 \leq \lambda \leq 1$, the recursively-defined set:

$$\mathcal{K}_h^\lambda = Q(\lambda \mathcal{K}_{h-1}^\lambda) \cap \Omega, \quad (7.3)$$

starting with $\mathcal{K}_0^\lambda = \Omega$, will be denoted as the **robust h -step λ -contractive controllable set**. Indeed, as shown in the cited reference, for all $x_0 \in \mathcal{K}_h^\lambda$ there exists an admissible control action \mathbf{u} which, without knowledge of the actual mode, keeps the future states inside Ω at least h time steps, and $x_k^\theta(x_0, \mathbf{u}) \in \lambda^k \Omega$ for all $\theta \in \mathcal{M}^h$.

When super-index λ is omitted, it will be assumed to take the value of

$\lambda = 1$, denoting the so-called *invariant* sets. Such convention ($\lambda = 1$) when omitted will also be used in later variations of the above invariant concepts.

If the iteration (7.3) converges for $h \rightarrow \infty$, the resulting set will be labelled as $\mathcal{K}_\infty^\lambda$. This is largest set of initial states in Ω for which there exists a mode-independent controller making $\mathcal{K}_\infty^\lambda$ to be λ -contractive (**maximal λ -contractive controllable set**), see (Kerrigan 2000) for details.

If the initial state belongs to the above maximal robust λ -contractive sets, it will robustly converge to the origin (irrespective of the mode sequence, without information on current mode).

7.2.2. Problem Statement and Auxiliary Notation

The objective of this chapter is to extend the above λ -contractiveness concepts to initial states outside the maximal robust set. Obviously, such states will not converge to the origin under some sequences; this work will propose suitable control laws for them, and will estimate the probability of success in driving the state to the origin.

Two information scenarios (either known or unknown current mode) will be considered. When current mode is known to the controller, a more flexible definition of $\mathcal{Q}(\mathcal{C})$ will be proposed, allowing for mode-dependent u^i in (7.2), and a higher probability of success will be guaranteed.

In order to address the problem with more generality, we will consider mode-dependent constraints, as well as the possibility of some transitions having zero probability to enlarge the obtained guaranteed sets.

The mode-dependent constraints and essential notation to be used throughout the paper will be defined next.

Constraints. Some mode-dependent state and input constraints will be considered. In particular, a list of sets $\bar{\Omega} := \{\Omega^{[1]}, \dots, \Omega^{[M]}\}$, $\bar{\mathbb{U}} := \{\mathbb{U}^{[1]}, \dots, \mathbb{U}^{[M]}\}$ will be specified as the constraint for each mode (x_k must belong to $\Omega^{[\theta_k]}$, and u_k must belong to $\mathbb{U}^{[\theta_k]}$). The origin $x = 0$ will be assumed to belong to all $\Omega^{[i]}$, and likewise $u = 0$ will be assumed to belong to all $\mathbb{U}^{[i]}$, for all $i \in \mathcal{M}$.

The notation Ω^* will denote the set in the augmented space $\mathbb{R}^n \times \mathcal{M}$ given by $\Omega^* := \{(x, \theta) \in \mathbb{R}^n \times \mathcal{M} : x \in \Omega^{[\theta]}\}$. Hence, the mode-dependent constraint can be stated, for all $k \geq 0$, as $(x_k, \theta_k) \in \Omega^*$, and $(u_k, \theta_k) \in \mathbb{U}^*$ being \mathbb{U}^* likewise defined. Analogously, for any set $\mathcal{C}^* \subset \Omega^*$, the ‘cuts’ for a particular mode will be understood as $\mathcal{C}^{[\theta]} := \{x \in \Omega^{[\theta]} : (x, \theta) \in \mathcal{C}^*\}$. Given an arbitrary set $\mathcal{C} \in \mathbb{R}^n$, the notation $\lambda\mathcal{C}$ will refer to a linear scaling of \mathcal{C} with scaling factor

$0 \leq \lambda \leq 1$. Given $\mathcal{C}^* \subset \Omega^*$, the scaling $\lambda\mathcal{C}^*$ will be understood as $\lambda\mathcal{C}^* := \{(x, \theta) : x \in \lambda\mathcal{C}^{[\theta]}\}$.

We will denote as $\Omega = \cap_i \Omega^{[i]}$, $\mathbb{U} = \cap_i \mathbb{U}^{[i]}$ which will be ‘safe’ sets fulfilling the constraints for all modes. If no mode information is available when computing the control action, the controller should obviously assume single Ω and \mathbb{U} as state and input constraints (mode-independent), as considered in Section 7.2.1.

Sequence Notation. For a particular $\theta \in \mathcal{M}^h$, $\theta = \{\theta_0, \theta_1, \dots, \theta_{h-1}\}$, denote by $\theta_{i:j}$, for $0 \leq i \leq j \leq h-1$, the sub-sequence $\{\theta_i, \theta_{i+1}, \dots, \theta_j\} \in \mathcal{M}^{j-i+1}$. Given $\eta \in \mathcal{M}^{h_1}$, $\xi \in \mathcal{M}^{h_2}$, denote as $\eta\xi$ the concatenated sequence

$$\{\eta_0, \dots, \eta_{h_1-1}, \xi_0, \dots, \xi_{h_2-1}\} \in \mathcal{M}^{h_1+h_2}$$

A sequence θ will be said to be admissible if $\prod_{h=1}^{l-1} \pi_{\theta_h \theta_{h-1}} \neq 0$. Given the transition probabilities, the so-called incidence matrix \mathcal{I} will be defined as the one whose element at position (i, j) is equal to 1 if $\pi_{ij} \neq 0$, and equal to zero otherwise.

The set of all admissible sequences of length h will be denoted as $\overline{\mathcal{M}}^h$. Evidently, a sequence is admissible if and only if all its sub-sequences are, too, admissible. Given a particular mode, its admissible successors will be denoted by $\mathcal{S}_i := \{j : \pi_{ji} \neq 0\}$. Of course, if $\mathcal{S}_i = \mathcal{M}$ for all i , then any arbitrary switching is admissible.

Note that if sequence switching has a minimum dwell-time constraint, such as the case considered in (Dehghan and Ong 2012), it can be trivially considered the output of a Markov process with a larger number of states, so the results to be presented here can also be applied to such a case.

Sets Associated to Sequences. The notation $\mathbf{u} \in \mathbb{U}_\theta$, $\mathbf{u} = \{u_0, u_1, \dots, u_{h-1}\}$ will be used for input sequences of length h such that constraints are not violated, *i.e.*, $u_k \in \mathbb{U}^{[\theta_k]}$. An analogous notation to state sequences $x \in \Omega_\theta$ will be used, too.

Predicted State Sequences. Given a mode sequence θ , for the controlled system (7.1), with an input sequence $\mathbf{u} \in \mathbb{U}_\theta$, let us define the associated state

sequence as x^θ , abusing of notation:

$$\begin{aligned} x_0, x_1^\theta &:= A_{\theta_0}x_0 + B_{\theta_0}u_0, x_2^\theta := A_{\theta_1}x_1^\theta + B_{\theta_1}u_1, \dots, \\ x_h^\theta &:= A_{\theta_{h-1}}x_{\theta_{h-1}}^\theta + B_{\theta_{h-1}}u_{h-1}, \end{aligned} \quad (7.4)$$

where actually x_k^θ is a function $x_k^\theta(x_0, \mathbf{u})$.

An input sequence (and the associated predicted state trajectory) will be denoted as admissible for initial state x_0 and sequence θ if mode-dependent constraints are not violated, i.e., if $\mathbf{u} \in \mathbb{U}_\theta$ and $\mathbf{x}^\theta(x_0, \mathbf{u}) \in \Omega_\theta$.

7.3. Augmented and Sequence-dependent Sets

In this section, generalizations of (7.2) will be proposed. First, the extension to the case when the controller knows the current operating mode and the admissible successors. Later, sequence-dependent sets will be defined so that, in Section 7.4, the knowledge of the probabilities of future sequences will allow the controller to ‘bet’ over the most likely outcome to decide the course of action.

7.3.1. Augmented h -step Sets

Let us redefine the one-step set in the augmented space Ω^* as:

$$\mathcal{Q}^*(\mathcal{C}^*) := \{(x_0, i) \in \Omega^* : \exists u \in \mathbb{U}^{[i]} \text{ such that } (A_i x_0 + B_i u, j) \in \mathcal{C}^* \forall j \in \mathcal{S}_i\}, \quad (7.5)$$

where knowledge of the mode i by the controller is implicitly embedded in the definition. Then, the iterative algorithm for h -step sets can be rewritten as Algorithm 5.

Theorem 7.1. *If Algorithm 5 converges (success=true), the set $\mathcal{K}_\infty^{\lambda,*}$ is the maximal λ -contractive controllable set contained in Ω^* .*

Proof: Omitted, as would be almost identical to those in the non-augmented space (Blanchini 1994, Theorem 3.1),(Kerrigan 2000). \square

Remark 7.1. *Convergence of the algorithm in a finite number of steps cannot be guaranteed in general. However, if there exists any λ such that the above algorithm converges (in maybe an infinite number of steps), there exists a finite h such that $\mathcal{K}_h^{\lambda,*}$ is $\hat{\lambda}$ -contractive for any $\lambda < \hat{\lambda} \leq 1$, following identical argumentations to*

Algoritmo 5 Recursive computation of h -step sets in augmented space

```

1: Set  $\mathcal{K}_0^{\lambda,*} = \Omega^*$ ;
2: Let  $h = 1$ ;
3: Compute  $\mathcal{K}_h^{\lambda,*} = \mathcal{Q}^*(\lambda\mathcal{K}_{h-1}^{\lambda,*})$ ;
4: si  $\mathcal{K}_h^{\lambda,*} = \mathcal{K}_{h-1}^{\lambda,*}$  entonces
4:   success=true;
4:   let  $\mathcal{K}_\infty^{\lambda,*} = \mathcal{K}_h^{\lambda,*}$ , STOP;
5: fin si
6: Let  $h = h + 1$ ;
7: si  $h < h_{MAX}$  entonces
7:   go to step 3;
8: si no
8:   let success=false STOP;
9: fin si

```

(Blanchini 1994, Theorem 3.2). Hence, if h_{MAX} is reached, bisection on $\hat{\lambda}$ can be used to find the smallest $\hat{\lambda}$ such that $\mathcal{K}_{h_{MAX}}^{\lambda,*}$ is $\hat{\lambda}$ -contractive, by testing if $\mathcal{K}_{h_{MAX}}^{\lambda,*} \subset \mathcal{Q}^*(\hat{\lambda}\mathcal{K}_{h_{MAX}}^{\lambda,*})$.

From the above results, for any $(x_0, \theta_0) \in \mathcal{K}_\infty^{\lambda,*}$ there exists a causal $u_0 = u_0(x_0, \theta_0)$ such that the successor $(x_1 = A_{\theta_0}x_0 + B_{\theta_0}u_0, \theta_1)$ is admissible for any possible successor mode $\theta_1 \in \mathcal{S}_{\theta_0}$, i.e., $x_1 \in \cap_{\theta_1 \in \mathcal{S}_{\theta_0}} \lambda\mathcal{K}_\infty^{\lambda, [\theta_1]}$.

Now, the following set in the original (non-augmented) state space can be defined:

Definition 7.2. Let us define h -step controlled safe set as:

$$\mathbb{S}_h^\lambda := \cap_{i \in \mathcal{M}} \mathcal{K}_h^{\lambda, [i]} \quad (7.6)$$

This set denotes the set of initial states for which there exists a control action (causal, dependent on current mode) that fulfils $(x_k, \theta_k) \in \lambda^k \Omega^*$ for $0 \leq k \leq h$ under any admissible sequence (irrespective of the starting mode θ_0).

The name comes because starting in such state is ‘safe’ (constraints not violated) at least for h steps, even if θ_0 is not yet known (assuming it will later be, of course).

After convergence, the sets $\mathcal{K}_\infty^{\lambda, [i]}$ are those for which an initial state in them, with initial mode i can be kept in the admissible regions forever, and steered to the origin with decay-rate λ . Then, $\mathbb{S}_\infty^\lambda := \cap_{i \in \mathcal{M}} \mathcal{K}_\infty^{\lambda, [i]}$ (to be named as *maximal controlled safe set*) denotes the set of states for which a causal admissible controller exists such that constraints will never be violated in the future,

irrespective of the starting mode.

Remark 7.2. Note that the maximal safe set $\mathbb{S}_\infty^\lambda$ is, in general, not λ -contractive, in the sense of (Kerrigan 2000). Indeed, the truly λ -contractive set is $\mathcal{K}_\infty^{\lambda,*}$; so, any state starting in $\mathbb{S}_\infty^\lambda$ will be kept at a future time k inside $\cup_{i \in \mathcal{M}} \lambda^k \mathcal{K}_\infty^{\lambda,[i]}$. A particular case in which $\mathbb{S}_\infty^\lambda$ is λ -contractive is discussed below.

Lemma 7.1. Under arbitrary switching (i.e., $\pi_{ji} \neq 0$ for all i, j), the maximal controlled safe set $\mathbb{S}_\infty^\lambda$ is control λ -contractive. It is, actually, the maximal control λ -contractive set with mode-dependent control action.

Proof: Consider $\mathcal{K}_\infty^{\lambda,*}$ which is maximal and fulfils, due to convergence:

$$\mathcal{K}_\infty^{\lambda,*} = \mathcal{Q}^*(\mathcal{K}_\infty^{\lambda,*}) = \{(x_0, i) \in \Omega^* : \exists (u, i) \in \mathbb{U}^* \text{ such that } A_i x_0 + B_i u \in \lambda \left(\bigcap_{j \in \mathcal{M}} \mathcal{K}_\infty^{\lambda,[j]} \right)\}, \quad (7.7)$$

but, from (7.6), trivially (replacing the last intersection):

$$\mathcal{K}_\infty^{\lambda,*} = \mathcal{Q}^*(\mathcal{K}_\infty^{\lambda,*}) = \{(x_0, i) \in \Omega^* : \exists (u, i) \in \mathbb{U}^* \text{ such that } A_i x_0 + B_i u \in \lambda \mathbb{S}_\infty^\lambda\} \quad (7.8)$$

Note that, if $x \in \mathbb{S}_\infty^\lambda$, then $(x, i) \in \mathcal{K}_\infty^{\lambda,*}$ for all i . Then, (7.8) entails that, whichever the mode, there exists a (mode-dependent) control action driving the successor state to the scaled set $\lambda \mathbb{S}_\infty^\lambda$. Thus, $\mathbb{S}_\infty^\lambda$ is control λ -contractive.

Note that, if $x \notin \mathbb{S}_\infty^\lambda$ then there exists j such that $(x, j) \notin \mathcal{K}_\infty^{\lambda,*}$. Due to maximality of $\mathcal{K}_\infty^{\lambda,*}$, then there does *not* exist a control action keeping all the future states admissible; hence, $\mathbb{S}_\infty^\lambda$ is maximal, too. \square

7.3.2. Sequence-dependent h -step Sets

The previous section enlarges the possible initial conditions with guaranteed convergence to the origin (with respect to the original robust sets in section 7.2.1). However, if the initial state and mode are not in $\mathcal{K}_\infty^{\lambda,*}$, then there exist a sequence of modes which will ultimately violate the constraints, whichever the controller. However, there might be some sequences for which a controller exists such that the state can be steered to $\mathcal{K}_\infty^{\lambda,*}$. So, there might be some probability of success in driving the state to the origin. This motivates the definition of the following l -step sets in the (non-augmented) state space for a particular sequence.

Given the system under study (7.1) and an arbitrary set \mathcal{C} , with associated mode-dependent state and input constraints, we will define the h -step

sequence-set $\mathcal{Q}_\theta(\mathcal{C})$, $\theta \in \mathcal{M}^h$ as:

$$\mathcal{Q}_\theta(\mathcal{C}) := \{x_0 \in \Omega^{[\theta_0]} : \exists \mathbf{u} \in \mathbb{U}_\theta \text{ such that } (x_k^\theta(x_0, \mathbf{u}_{0:k-1}), \theta_k) \in \Omega^* \\ \forall k \in \{1, 2, \dots, h-1\}, x_h^\theta(x_0, \mathbf{u}) \in \mathcal{C}\}, \quad (7.9)$$

where x_k^θ is obtained with (7.4). Hence, $\mathcal{Q}_\theta(\mathcal{C})$ is the set of initial states for which there exists an admissible control actions for a particular sequence θ , such that, the state is driven in an admissible trajectory to \mathcal{C} in $\text{len}(\theta)$ steps, while keeping the evolution of the state inside Ω^* for the first $h-1$ step.

When the sequence length is 1, i.e., $\eta = \{i\}$, $i \in \mathcal{M}$ the above expression will actually be a ‘classical’ 1-step set, represented, with a slight abuse of notation, by:

$$\mathcal{Q}_i(\mathcal{C}) := \{x_0 \in \Omega^{[i]} : \exists u_0 \in \mathbb{U}^{[i]} \text{ such that } A_i x_0 + B_i u_0 \in \mathcal{C}\} \quad (7.10)$$

For instance, the one-step set in (7.5) could have been written as $\mathcal{Q}^*(\mathcal{C}^*) = \{(x, i) \in \Omega^* : x \in \mathcal{Q}_i(\cap_{j \in \mathcal{S}_i} \mathcal{C}^{[j]})\}$.

Lemma 7.2. Consider mode sequences $\eta \in \mathcal{M}^{h_1}$, $\xi \in \mathcal{M}^{h_2}$, $h_1 \geq 1$, $h_2 \geq 1$. Then, $\mathcal{Q}_{\eta\xi}(\mathcal{C}) = \mathcal{Q}_\eta(\mathcal{Q}_\xi(\mathcal{C}))$.

Proof (omitted) is straightforward from the definition. Such property will allow recursive definitions of the above sets, similar to standard algorithms.

Corollary 7.1. A particular case of the above lemma when $l_1 = 1$, i.e., $\eta = \{i\}$, $i \in \mathcal{M}$ gives:

$$\mathcal{Q}_{i\xi}(\mathcal{C}) = \mathcal{Q}_i(\mathcal{Q}_\xi(\mathcal{C})) \quad (7.11)$$

As possible future mode sequences are infinite, there might be an infinite number of the above sequence-dependent sets. However, in some cases, the number of sets are finite and depending only in a few of the first sequence elements, motivating the lemma below.

Lemma 7.3. If $\mathcal{Q}_{\eta\xi_1}(\mathcal{C}) = \mathcal{Q}_\eta(\mathcal{C})$ for all $\eta\xi_1 \in \overline{\mathcal{M}}^{h+1}$ then $\mathcal{Q}_{\eta\psi}(\mathcal{C}) = \mathcal{Q}_\eta(\mathcal{C})$ for any admissible sequence ψ of any arbitrary length.

Proof: Consider a sequence of length $h+2$ expressed as $\xi_1 \in \mathcal{M}$, $\xi_2 \in \mathcal{M}$, $\eta \in \overline{\mathcal{M}}^h$, such that $\eta\xi_1\xi_2 \in \overline{\mathcal{M}}^{h+2}$. Consider now the associated sequence set $\mathcal{Q}_{\eta\xi_1\xi_2}$. Applying first Corollary 7.1, splitting out the first element of η , i.e., $\eta\xi_1\xi_2 = \eta_0\eta_{1:h-1}\xi_1\xi_2$, we have:

$$\mathcal{Q}_{\eta\xi_1\xi_2}(\mathcal{C}) = \mathcal{Q}_{\eta_0\eta_{1:h-1}\xi_1\xi_2}(\mathcal{C}) = \mathcal{Q}_{\eta_0}(\mathcal{Q}_{\eta_{1:h-1}\xi_1\xi_2}(\mathcal{C})) \quad (7.12)$$

As $\eta_{1:h-1}\xi_1\xi_2 \in \overline{\mathcal{M}}^{h+1}$, $\mathcal{Q}_{\eta_{1:h-1}\xi_1\xi_2} = \mathcal{Q}_{\eta_{1:h-1}\xi_1}$ by assumption. Hence (7.12) can be expressed as:

$$\mathcal{Q}_{\eta_0}(\mathcal{Q}_{\eta_{1:h-1}\xi_1\xi_2}(\mathcal{C})) = \mathcal{Q}_{\eta_0}(\mathcal{Q}_{\eta_{1:h-1}\xi_1}(\mathcal{C})) = \mathcal{Q}_{\eta_{\xi_1}}(\mathcal{C}) \quad (7.13)$$

As $\mathcal{Q}_{\eta_{\xi_1}} = \mathcal{Q}_\eta$ for all admissible η_{ξ_1} , (7.13) implies $\mathcal{Q}_{\eta_{\xi_1}\xi_2} = \mathcal{Q}_{\eta_{\xi_1}} = \mathcal{Q}_\eta$ for all admissible sequences. An induction argumentation makes $\mathcal{Q}_{\eta\psi} = \mathcal{Q}_\eta$ for any finite-length sequence ψ . \square

Recursive Algorithm for \mathcal{Q}_η . The finite collection of sets arising from the above lemma can be computed by Algorithm 6 (if it succeeds), with a target set \mathcal{C} used to ‘seed’ the recursions.

Algoritmo 6 Recursive computation of sequence-dependent sets $\mathcal{Q}_\eta(\mathcal{C})$

Inputs: \mathcal{C} .

1. Start with all $\psi \in \mathcal{M}$. Compute $\mathcal{Q}_\psi(\mathcal{C})$. **If** all $\mathcal{Q}_\psi = \mathcal{C}$, let success=true; **STOP**.
 2. Let $h = 1$.
 3. For all $\xi \in \mathcal{M}$, $\eta \in \overline{\mathcal{M}}^h$ such that $\xi\eta$ is admissible, compute $\mathcal{Q}_{\xi\eta}(\mathcal{C})$ using the one-step set in Lemma 7.2. $\mathcal{Q}_{\xi\eta}(\mathcal{C}) = \mathcal{Q}_\xi(\mathcal{Q}_\eta(\mathcal{C}))$.
 4. **If** $\mathcal{Q}_{\eta\psi}(\mathcal{C}) = \mathcal{Q}_\eta(\mathcal{C})$ for all $\psi \in \mathcal{M}$, $\eta \in \overline{\mathcal{M}}^h$ such that $\psi\eta$ is admissible, **then** let success=true; $h_c = h$; **STOP**.
 5. Let $h = h + 1$, **If** $h < h_{MAX}$ go to step 3 **else**, let success=false; $h_c = h_{MAX}$; **STOP**.
-

If ‘success’ ends up having a true value, the result is a list of converged sets $\mathcal{Q}_\eta(\mathcal{C})$, covering all $\eta \in \overline{\mathcal{M}}^{h_c}$. Also, intermediate sets $\mathcal{Q}_\eta(\mathcal{C})$ with $len(\eta) < h_c$ should also be stored for further interpretation, see Section 7.4. IF ‘success’ were not true, the algorithm wouldn’t have found the final converged sets; anyway, they can still be useful in some cases, as discussed in Section 7.4.3.

The interpretation of the converged set $\mathcal{Q}_\eta = \mathcal{Q}_{\eta\xi}$ is that for all x in \mathcal{Q}_η there exists a control action driving the state to \mathcal{C} under any sequence starting with η . However, note that the control action might be sequence-dependent, so it would need a ‘non-causal’ controller. Hence, further developments are needed in order to devise causal controllers, as discussed next.

7.4. Reliable On-line Controller Design

The above sequence-dependent sets can be used in order to decide a course of action to reach the origin with high likelihood. Let us consider each of the two mode information set-ups, i.e., controller knows current mode or not.

7.4.1. Controller with known Current Mode in On-line Operation

Of course, if at a particular instant, say $t = 0$, we have $(x_0, \theta_0) \in \mathcal{K}_\infty^{\lambda,*}$, there exists a control law keeping the next state and mode in $\mathcal{K}_\infty^{\lambda,*}$, so that this controller is the one to be applied ensuring 100 % likelihood of being admissible at all future times.

When $(x_0, \theta_0) \notin \mathcal{K}_\infty^{\lambda,*}$, the following corollary combines the two above algorithms by using the results of Algorithm 5 as seeds \mathcal{C} of Algorithm 6.

Corollary 7.2. *Given $\gamma \in \overline{\mathcal{M}}^{h+1}$, $\gamma = \eta\psi$, for all $x_0 \in \mathcal{Q}_\eta(\mathcal{K}_\infty^{\lambda,[\psi]})$, there exists a control law driving the state to $\mathcal{K}_\infty^{\lambda,*}$ from x_0 under sequence η , and remaining in $\mathcal{K}_\infty^{\lambda,*}$ thereafter for any subsequent mode sequence beginning with ψ , converging to the origin with contraction rate λ .*

Proof: The proof is trivial from the definition of $\mathcal{Q}_\eta(\mathcal{K}_\infty^{\lambda,[\psi]})$, the λ -contractiveness of $\mathcal{K}_\infty^{\lambda,*}$ and the fact that $(x, \psi) \in \mathcal{K}_\infty^{\lambda,*}$ for any $x \in \mathcal{K}_\infty^{\lambda,[\psi]}$. \square

Note that Corollary 7.2 does *not* require convergence of Algorithm 6, but only convergence of Algorithm 5. Note, too, that the sets $\mathcal{Q}_\eta(\mathcal{K}_\infty^{\lambda,[\psi]})$ get ‘larger’ as $len(\eta)$ increases. In precise terms:

Corollary 7.3. *Given $\gamma \in \overline{\mathcal{M}}^h$, $\gamma = \eta\psi_1\psi_2$, then $\mathcal{Q}_\eta(\mathcal{K}_\infty^{\lambda,[\psi_1]}) \subset \mathcal{Q}_{\eta\psi_1}(\mathcal{K}_\infty^{\lambda,[\psi_2]})$.*

Proof: Indeed, the set of x_0 which can be driven to the origin with any arbitrary sequence starting with $\eta\psi_1$ –i.e., $\mathcal{Q}_\eta(\mathcal{K}_\infty^{\lambda,[\psi_1]})$ from Corollary 7.2– is a subset of those which can be driven with the particular case $\eta\psi_1\psi_2$ –i.e., $\mathcal{Q}_{\eta\psi_1}(\mathcal{K}_\infty^{\lambda,[\psi_2]})$ –. \square

So, as Algorithm 6 progresses, a larger family of nested sets (in the above sense) is obtained, until convergence (if it happens to occur) or exhaustion of computational resources. A practical interpretation of the above corollary means that progressively longer sequences will reach points further away from the origin. As such sequences will be less likely, initial conditions far away from the origin will have lower likelihood of being steered to the origin than those close to it, as intuitively expected.

Reliable Control Algorithm.

The above results can be used in on-line control to obtain a control action which drives the state to the maximal set $\mathcal{K}_\infty^{\lambda,*}$ with a given probability (associated to the most likely sequence).

As above discussed, if $(x_0, \theta_0) \in \mathcal{K}_\infty^{\lambda,*}$, 100 % reliability can be guaranteed. Otherwise, the control action resulting from Algorithm 7 below can be easily computed (at least when all involved sets are polytopic ones: everything amounts to linear constraints). Let us first present the algorithm, and discuss later the meaning of the involved steps.

Algorithm inputs are current state, current mode and all the sets resulting from Algorithm 6, either until convergence or until $h_c = h_{MAX}$ has been reached without success. As h_c can be different for each $\mathcal{K}_\infty^{\lambda, [\xi]}$, $\xi \in \mathcal{M}$, when Algorithm 6 is run, it will either converge to a finite $h_{c, \xi}$ or reach $h_{c, \xi} = h_{max}$ without success.

Algoritmo 7 Reliable On-line Controller

Inputs: $x_0, \theta_0, \mathcal{Q}_\eta(\mathcal{K}_\infty^{\lambda, [\xi]}), \eta \xi \in \overline{\mathcal{M}}^{h_{c, \xi}}$

- 1: **si** $x_0 \in \mathcal{K}_\infty^{\lambda, [\theta_0]}$ **entonces**
- 1: Compute any control such that: $x_1 \in \bigcap_{j \in \mathcal{S}_{\theta_0}} \mathcal{K}_\infty^{\lambda, [j]}$. **Exit;**
- 2: **si no**
- 2: Compute the set of sequences:

$$\Gamma(x_0, \theta_0) := \{\gamma : \gamma = \eta \xi, 1 \leq \text{len}(\eta) \leq h_{c, \xi}, \xi \in \mathcal{M}, x_0 \in \mathcal{Q}_\eta(\mathcal{K}_\infty^{\lambda, [\xi]}), \gamma \text{ is admissible}, \gamma_0 = \theta_0\} \quad (7.14)$$

- 3: **fin si**
- 4: **si** $\Gamma(x_0, \theta_0) = \emptyset$ **entonces**
- 4: Throw('Failure'), **Exit;**
- 5: **si no**
- 5: determine the sequence β in $\Gamma(x_0, \theta_0)$ with highest probability (conditioned to $\gamma_0 = \theta_0$), i.e.,

$$\beta := \arg \max_{\gamma \in \Gamma(x_0, \theta_0)} Pr(\gamma | \gamma_0 = \theta_0) \quad (7.15)$$

- 6: **fin si**
 - 7: Let $v := \text{len}(\beta)$. Compute a control steering the next state to $\mathcal{Q}_{\beta_{1:v-2}}(\mathcal{K}_\infty^{\lambda, [\xi]}),$ being $\xi = \beta_{v-1}$, i.e., the last element of β , **Exit;**
-

Depending on the initial state, the presented algorithm can succeed in obtaining a stabilizing controller or not. Let us discuss such situations, detected at Step 3.

Interpretation of Step 4.

Proposition 7.1. *If $\Gamma(x_0, \theta_0) = \emptyset$, and $h_{c,\xi} < h_{MAX}$ for all ξ , then there does not exist a control law which drives the state to the origin, irrespective of the mode sequence: the probability of success is zero.*

Proof: Indeed, there is no $\mathcal{Q}_\eta(\mathcal{K}_\infty^{\lambda, [\xi]})$ with η starting with θ_0 to which x_0 belongs, so, as Algorithm 6 converged, there does not exist any arbitrary-length sequence for which there exists a control sequence bringing it to $\mathcal{K}_\infty^{\lambda, *}$. \square

The statement `Throw('Failure')` indicates that the situation in Proposition 7.1 did occur and, hence, there is no possibility of driving the state to the origin violating constraints: suitable abort or constraint relaxation procedures should be handled elsewhere.

Remark 7.3. *Actually, in the algorithm implementation, if $h_{c,\xi} = l_{MAX}$ for some ξ (i.e., non-convergence of the prior Algorithm 6 for at least one $\mathcal{K}_\infty^{\lambda, [\xi]}$) the meaning would be slightly different: the allowed computational resources would not prove existence of a stabilizing controller for the explored sequences; however, there might be longer ones for which such controller exists. Again, backup controllers for such a case are out of the scope of this work.*

Interpretation of Steps 5 and 6 (success). Regarding steps 5 and 6, note that if $\Gamma(x_0, \theta_0) \neq \emptyset$, a controller which steers the state to the origin for any sequence in $\Gamma(x_0, \theta_0)$ exists. However, the control action might be different for each sequence, of course; then, as the controller must be causal, betting on the most probable one for the future is the choice proposed in Algorithm 7.

Indeed, being $v = \text{len}(\beta)$, consider any controller which steers the next state to the 'successor set' $\mathcal{Q}_{\beta_{1:v-2}}(\mathcal{K}_\infty^{\lambda, [\xi]})$, being $\xi = \beta_{v-1}$, i.e., ξ is the last element of β . By definition, such controller does exist and its computation requires just a one-step feasibility condition, trivial to solve for in the linear case.

Note also that, as sequence-dependent sets are nested, in the sense precised in Corollary 7.3, if $\gamma \in \Gamma(x_0, \gamma_0)$ then any longer sequences starting with γ also belong to $\Gamma(x_0, \gamma_0)$; however, their probability will be, evidently, lower. To avoid searching for irrelevant sequences, we will denote as $\text{Trim}(\Gamma)$ the result of removing from a set of sequences Γ any sequence whose starting sub-sequence is already in Γ . Obviously, Γ should actually be replaced, then, by $\text{Trim}(\Gamma)$ in (7.15).

Off-line Reliability Computation. If the initial mode is not known at the time of reliability determination (i.e., off-line, prior to actual operation), then

100 % reliability can be guaranteed only for $x_0 \in \mathbb{S}_\infty^\lambda$. Otherwise, i.e., considering a particular $x_0 \notin \mathbb{S}_\infty^\lambda$, let us assume that only an *a priori* probability vector for each mode $p = \{p_1, \dots, p_M\}$ with $p_i := Pr(\theta_0 = i)$, $i \in \mathcal{M}$ is available.

To incorporate the concepts in Step 1 without knowledge of initial mode, instead of (7.14) we must consider:

$$\Gamma_2(x_0) := \{\gamma : \gamma = \eta\xi, 0 \leq \text{len}(\eta) \leq h_{c,\xi}, \xi \in \mathcal{M}, x_0 \in \mathcal{Q}_\eta(\mathcal{K}_\infty^{\lambda, [\xi]}), \gamma \text{ is admissible}\} \quad (7.16)$$

abusing the notation by understanding $\mathcal{Q}_\eta(\mathcal{K}_\infty^{\lambda, [\xi]}) := \mathcal{K}_\infty^{\lambda, [\xi]}$ for zero-length η .

Analogously to Proposition 7.1, if $\Gamma_2(x_0) = \emptyset$ and $h_{c,\xi} < h_{MAX}$ for all $\xi \in \mathcal{M}$ (convergence) there is no sequence for any initial mode for which the state could be steered to the origin. Otherwise, a controller which steers the state to the origin for some sequences exists. A lower bound for the probability of success of the above algorithm can be obtained with the result below:

Lemma 7.4. *The controller in Algorithm 7 has a probability of driving the state to the origin²³ of, at least:*

$$\underline{p} := \sum_{i \in \mathcal{M}} \left(p_i \cdot \max_{\gamma \in \text{Trim}(\Gamma_2(x_0))} Pr(\gamma | \gamma_0 = i) \right) \quad (7.17)$$

Proof: Indeed, the sequence $\beta_2(x_0, i)$ from $\Gamma_2(x_0)$, given by

$$\beta_2(x_0, i) := \arg \max_{\gamma \in \Gamma_2(x_0)} Pr(\gamma | \gamma_0 = i) \quad i \in \mathcal{M} \quad (7.18)$$

would be the sequence β that Algorithm 7 would produce if $\theta_0 = i$ at the moment of computing the control action. It is straightforward to see that summation of the probability of each of the $\beta_2(x_0, i)$ conditioned to its starting element, multiplied by the probability of each starting mode gives (7.17). \square

7.4.2. Controller With no On-line Mode Information

If the current mode is not available to the controller, a controller can be crafted using as seed to Algorithm 6 the ‘classical’ maximal λ -contractive controllable set from Section (7.2.1), i.e., using $\mathcal{K}_\infty^\lambda$ in Step 1 (instead of the previously used augmented $\mathcal{K}_\infty^{\lambda,*}$). Also, (7.14) in the controller algorithm must be redefined, changing Γ to:

$$\Gamma_3(x_0) := \{\gamma : 1 \leq \text{len}(\gamma) \leq h, x_0 \in \mathcal{Q}_\gamma(\mathcal{K}_\infty^\lambda), \gamma \text{ is admissible}\} \quad (7.19)$$

²³actually, with geometric contraction rate λ after h steps

and (7.15) should be changed to finding the sequence with maximal probability in (7.19):

$$\beta := \arg \max_{\gamma \in \text{Trim}(\Gamma_3(x_0))} Pr(\gamma) \quad (7.20)$$

where probability $Pr(\gamma)$ is, obviously, computed using whatever prior information p available (or, for instance, assuming stationary state of the Markov chain).

Then, the control law steering the state to the *successor set* $\mathcal{Q}_{\gamma(1:end)}(\mathcal{K}_{\infty}^{\lambda})$ has at least probability $p = \text{prob}(\beta(x_0))$ of success²⁴. Further details are omitted as they are almost identical to the known-mode case. Obviously, the resulting controller has lower reliability than the one in Section 7.4.1.

7.4.3. Discussion and Computational Issues

Obviously, the applicability of the presented results is limited by the required computational load for the proposed algorithms. In the polyhedral case, computational issues arise because of two reasons:

- on the one hand, due to the possible increase of the number of vertices of the M sets which get iteratively refined in Algorithm 5: the time taken to compute each step increases with h (due to a larger number of vertices in the involved polytopes),
- on the other hand, because of the heavy increase of the number of sequences in Algorithm 6.

The computational burden of Algorithm 5 is well studied, as it is a higher-dimensional version of standard control-invariant set algorithms; such algorithms and their computational issues are described in (Kerrigan 2000, Kvasnica *et al.* 2004) and references therein: although the number of involved sets in Algorithm 5 does not increase with the number of iterations, the basic drawback lies in vertex enumeration/projection steps whose computational demands make them, in particular, very hard to solve for systems with a large number of input variables.

Regarding the sequence-dependent sets in Algorithm 6, the number of possible sequences to consider grows exponentially with the number of steps h in

²⁴ Indeed, the summation in (7.17) cannot be made because the controller will not be able to make the choice between the different $\beta(x_0, i)$ that (7.17) assumes, as mode i will *not* be known. Hence, the proposed option in (7.20) is betting for the (single) most likely sequence driving to the robust λ -contractive controllable set. Note that, due to the lack of mode information, the probability of β in (7.20) is lower than the bound p in (7.17).

most cases (for instance, for all regular Markov chains). In particular, it can be easily proved that the number of sets for a given value of h (in the M needed runs of Algorithm 6) is given by the sum of the elements of the h -th power of the incidence matrix \mathcal{I}^l . So, denoting by $\bar{\lambda}$ the largest eigenvalue of \mathcal{I} , for long enough sequences, the number of sequence-dependent sets may increase with $\mathcal{O}(\bar{\lambda}^l)$. Note however, that some of the sets obtained with (7.11) might be coincident with previously-computed ones so further one-steps sets from them can be skipped –in fact, when such thing happens with all sets, we have the conditions in Lemma 7.3–.

A last issue is the existence of multiple control actions steering the state to the successor set: the on-line control action can be computed, for instance, minimizing a suitable cost index, say, one-step control effort, if the solution were not unique (using linear or quadratic programming).

Algorithm Modifications. From the above discussion, there might be not enough computational resources to achieve full convergence of Algorithm 6 (or the algorithm might not converge in a finite number of iterations). When not converged, if initial state is outside all of the obtained sets on-line finite-horizon controllers might be conceived trying to reach (or land ‘close enough’ to) some of the computed sets.

A second modification can be suggested if Algorithm 5 does not converge: in such a case, Algorithm 6 can be seeded with any subset of either $\mathcal{K}_\infty^\lambda$ (unknown mode) or $\mathcal{K}_\infty^{\lambda,*}$ (known mode), computed, for instance, stopping Algorithm 5 when a contractive set is found (Remark 7.1), or via Lyapunov functions and LMIs (Zhang and Boukas 2009, Ma and Boukas 2011) or, plainly, the origin (safe, for sure, even for unstable plants).

Also, alternative controllers in step 5 of Algorithm 7 could be crafted, trying to steer the state to the intersection of several of the \mathcal{Q}_η sets arising from Γ , if possible. In that way, the controller would ‘bet’ on several sequences –at least for one sample–. Finding common control actions for several sequences in multi-step settings might also be addressable. However, the number of possible sequence groups is $2^q - 1$ being q the number of elements of Γ ; also, projections involved in computing the controllable sets severely hinder computation of multi-step feasible sets. Hence, the computational complexity of determining the maximum-reliability controller and its associated controllable sets appears to be insurmountable except for simplistic cases.

As a last remark, if transition probabilities (or prior information) were uncertain, the bound \underline{p} should be computed maximizing the worst-case proba-

Cuadro 7.1: Example parameters.

Parameters	Operation modes		
	i=1	i=2	i=3
A_i	$\begin{bmatrix} -0,8891 & -0,8149 \\ 0,8149 & -0,8891 \end{bmatrix}$	$\begin{bmatrix} -0,4446 & 0,4074 \\ -0,4074 & -0,4446 \end{bmatrix}$	$\begin{bmatrix} 0,5636 & -0,7240 \\ 0,7240 & 0,5636 \end{bmatrix}$
B_i	$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0,5 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1,2 \end{bmatrix}$
$\Omega^{[i]}$	$\begin{bmatrix} -10 \leq x_1 \leq 10 \\ -10 \leq x_2 \leq 10 \end{bmatrix}$	$\begin{bmatrix} -5 \leq x_1 \leq 5 \\ -12 \leq x_2 \leq 12 \end{bmatrix}$	$\begin{bmatrix} -5 \leq x_1 \leq 5 \\ -14 \leq x_2 \leq 8 \end{bmatrix}$

$$\mathbb{U} = [-9 \leq u \leq 9], \quad \mathbb{P} = \begin{bmatrix} 0,98 & 0,5 & 0 \\ 0,02 & 0,1 & 0,05 \\ 0 & 0,4 & 0,95 \end{bmatrix}, \quad \mathcal{I} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

bility estimate. Anyway, as the shape of sequence-dependent sets is independent of the probability computation, the ‘uncertain probability’ case does not modify any essential aspect of the procedures presented in this manuscript so details are left to the reader.

7.5. Numerical Example

Consider the second-order MJLS (7.1) with three operating modes, $i = \{1, 2, 3\}$. The corresponding parameters are given in Table 7.1. Also, a geometric contraction rate specification $\lambda = 0,98$ was chosen. For off-line reliability computations, the prior probability of each mode $Pr(\theta_0 = i)$ has been computed assuming stationary state of the mode Markov chain, resulting in $p_1 = 0,7353$, $p_2 = 0,0294$ and $p_3 = 0,2353$.

First, the sets $\mathcal{K}_\infty^{\lambda, [i]}$ are computed by executing Algorithm 5 until convergence using the one-step set (7.10). This initial algorithm converged in four steps. The maximal safe set $\mathbb{S}_\infty^\lambda$ is determined, according to Definition 7.2. So, this set has probability 1 of success.

Then, Algorithm (6) is run three times, setting for each execution, the corresponding $\mathcal{K}_\infty^{\lambda, [i]}$ as seed. The result is a list of sets, i.e., $\mathcal{Q}_\eta(\mathcal{K}_\infty^{\lambda, [1]})$, $\mathcal{Q}_\xi(\mathcal{K}_\infty^{\lambda, [2]})$ and $\mathcal{Q}_\phi(\mathcal{K}_\infty^{\lambda, [3]})$. The converged lengths are, too, $h_{c,1} = h_{c,2} = h_{c,3} = 4$. The number of possible sequences, from the powers of the incidence matrix \mathcal{I} , are, for l ranging from zero (the 3 maximal sets $\mathcal{K}_\infty^{\lambda, [i]}$) to 4 is $\{3, 7, 17, 41, 99\}$. However, the number of *new* sequence-dependent sets appearing at each step (i.e., different from those sets present in previous steps, with default MPT tolerances) is only $\{3, 6, 11, 3, 0\}$, so we have after convergence (Lemma 7.3) a total of 23 sets. The new sets at each step appear in Figure 7-1.

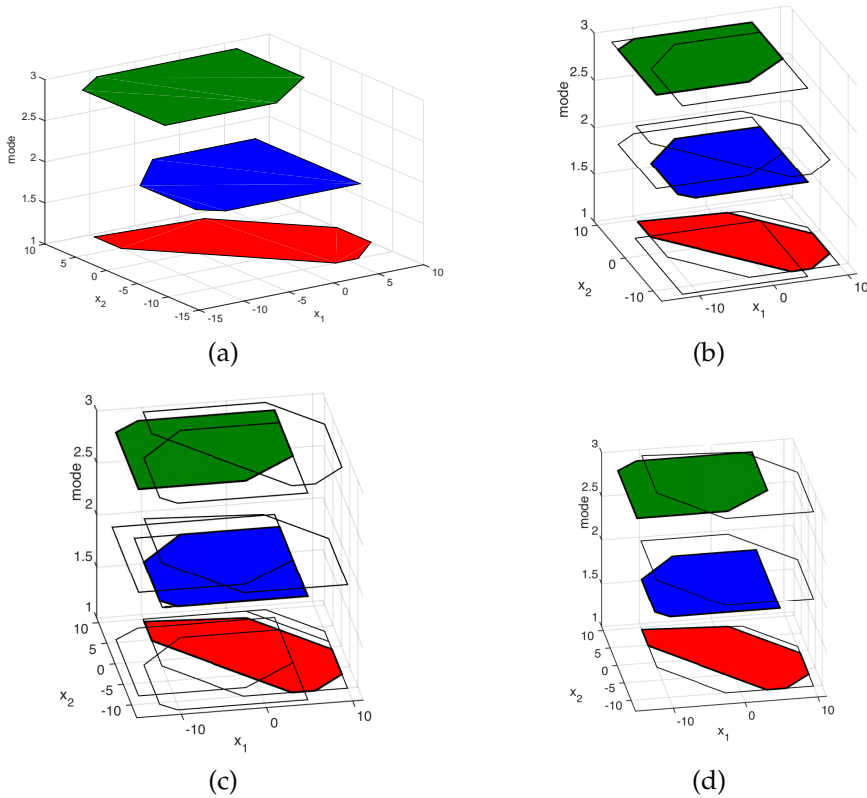


Figure 7-1: (a) Maximal controllable sets for each mode. [The projection of these sets onto the x -space are the dashed lines in Figure 7-3];
 (b) 1-step sets from each of the sets in plot (a), for each mode (six new sets appearing);
 (c) 2-step sets (11 new sets appearing);
 (d) 3-step sets (three new sets appearing).

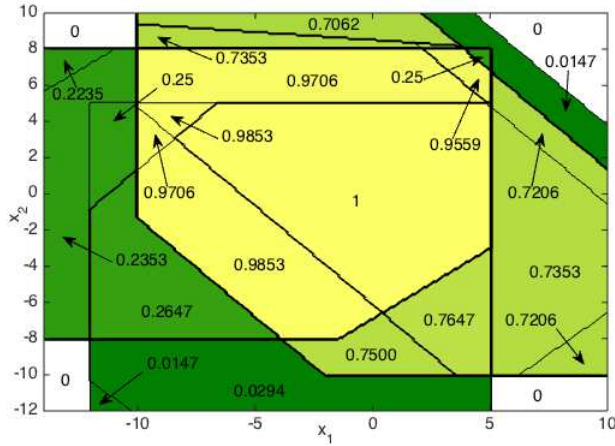


Figura 7-2: Reliability bound for different initial states, under stationary mode assumption.

Using the 23 obtained sets, their projection on x -space would lead to Figure 7-2; however, in the figure, a colour map has been superimposed representing the reliability bound \underline{p} from (7.17) (both numerical figures and yellow (high reliability) to green (low reliability) fillings appear). Regions for which no stabilizing controller (fulfilling the required constraints) exists for any sequence are filled in white. The set $\mathbb{S}_\infty^\lambda$ is, obviously, labelled with reliability bound equal to one in the figure.

In order to detail the probability computation, consider, for instance, an arbitrarily chosen point $x_0 = [3,433; 7,157]$, which does *not* belong to $\mathbb{S}_\infty^\lambda$. However, x_0 belongs to (trimming longer sequences –Corollary 7.3–):

$$\mathcal{Q}_\eta(\mathcal{K}_\infty^{\lambda,[1]}) \quad \text{for } \eta \in \{\boxed{\{1\}}, \{12\}, \{32\}\} \quad (7.21)$$

$$\mathcal{Q}_\xi(\mathcal{K}_\infty^{\lambda,[2]}) \quad \text{for } \xi \in \{\{1\}, \{3\}, \{11\}, \{33\}, \{111\}\} \quad (7.22)$$

$$\mathcal{Q}_\phi(\mathcal{K}_\infty^{\lambda,[3]}) \quad \text{for } \phi \in \{\{12\}, \{112\}\} \quad (7.23)$$

$$\mathcal{K}_\infty^{\lambda,[3]} \quad (7.24)$$

Let us consider an on-line operation measuring $\theta_0 = 1$. Then, the most likely sequence γ in $\Gamma(x_0, 1)$ from (7.14), i.e., conditioned to initial mode equal to 1, is $\{11\}$ (probability 0.98, from the boxed η in (7.21) above). Then, a control action steering the successor state to $\mathcal{K}_\infty^{\lambda,[1]}$ would be proposed by Algorithm 7.

For off-line reliability computation (a priori), apart from the case $\theta_0 = 1$, the other remaining two possibilities should be considered. For $\theta_0 = 2$, there

reliability bound, understood as likelihood of driving the state to the origin without violating constraints. The basic idea is the fact that, as soon as the state lands at particular 'safe' sets, reliability is 100 %. Then, some probability of success \underline{p} can be asserted for initial conditions in the (larger) sets for which there exists a finite-length sequence with a controller steering the state to such safe sets.

The proposed sets are a probabilistic generalization of the maximal robust λ -contractive sets in prior literature. Mode-dependent constraints are incorporated, too. As intuitively expected, the further away initial conditions are from the origin, the lower the computed (state-dependent) reliability bounds are.

Capítulo 8

Reliability and Time-to-Failure Bounds for Constrained MJLS

Nota sobre el capítulo: En este capítulo se extienden los resultados anteriores para MJLS con perturbaciones. Además, para el caso de MJLS donde uno o varios modos de funcionamiento hacen inestable al sistema, se presenta el concepto de tiempo medio a fallar, entendiéndose, como el tiempo –en media– en que se violará las restricciones del sistema. En el capítulo anterior, la ley de control obtenida es subóptima, mientras que la acción de control propuesta aquí es óptima para el caso de MJLS sin perturbaciones.

Abstract: This chapter presents a methodology to obtain a guaranteed-reliability controller for constrained linear systems which switch between different modes according to a Markov chain (Markov-jump linear systems). Inside the classical maximal robust controllable set, there is 100% guarantee of never violating constraints at future time. However, outside such set, some sequences might make hitting constraints unavoidable for some disturbance realisations. A guaranteed-reliability controller based on a *greedy* approach was proposed in Chapter 7 for disturbance-free, robustly stabilisable Markov-jump linear systems. Here, extensions are presented by, first, considering bounded disturbances and, second, presenting an iterative algorithm based on dynamic programming. In non-stabilisable systems, reliability is zero; therefore, prior results cannot be applied; in this case, optimisation of a mean-time-to-failure bound is proposed, via minor algorithm modifications. Optimality can be proved in the disturbance-free finitely-generated case.

8.1. Introduction

This chapter extends the set-invariance ideas (Blanchini 1999, Kerrigan 2000) to provide some robust *reliability* and *mean time to failure* (MTTF) bounds (and associated controllers), in processes where disturbances are bounded and parameter changes can be modelled as a set of discrete-time linear systems (operation *modes*), with mode transitions governed by a Markov chain (Markov-jump linear systems (MJLS)). In this MJLS case, the maximal sets for which a mode-dependent control action exists making them robustly invariant can, too, be computed by adapting the above concepts, see (Patrinos *et al.* 2014, Lemma 16), or (Grieder *et al.* 2005, Dehghan and Ong 2012) for switching systems cases. Of course, linear matrix inequalities (LMI) may be used to handle ellipsoidal invariant sets, see Section 2.5 which is based on the work (Lu *et al.* 2013).

Chapter 7 defined *reliability* as the probability of avoiding future constraint violations, in a disturbance-free setting. Basically, the main idea proposed there is the fact that, if the initial state is outside the maximal robust controllable set, there might exist mode sequences such that constraint violation is unavoidable ('failure'): if we are 'close' to the maximal invariant set, such failure can be avoided with high probability, but it will not be the case 'far away' from it. The cited work proposed a solution, based on so-called 'sequence-dependent' sets; a sub-optimal 'greedy' action was chosen to be the one associated to the most likely sequence with which the invariant set was reachable.

The above-cited results apply only to *stabilisable, undisturbed* MJLS. Apart from the absence of disturbances, there might be better controller options than the greedy controller, exploiting the possibility of ‘betting’ on more than one sequence. Also, if robust invariant sets do not exist (some cases of disturbed/non-stabilisable systems), sooner or later constraints will be violated, so mean-time-to-failure ideas should be considered, as reliability will be zero in such situation.

The objective of this chapter is to extend the above-discussed shortcomings of Chapter 7, by considering dynamic-programming argumentations for MJLS with bounded disturbances and, in regions where reliability is zero, considering a controller which maximises a MTTF bound. No probabilistic description over uncertainty will be assumed, so the proposed reliability/MTTF measures will be robust (worst-case) bounds regarding disturbances. Numerical results show improvements over Chapter 7 and, actually, optimality can be proved in some cases.

8.2. Preliminaries

Consider a non-autonomous discrete time-varying linear system:

$$x_{k+1} = A_{\theta_k} x_k + B_{\theta_k} u_k + E_{\theta_k} w_k, \quad \theta_k \in \mathcal{M}, \quad k \geq 0 \quad (8.1)$$

where $x_k \in \mathbb{R}^n$ represents the state vector, $u_k \in \mathbb{R}^m$ the control actions, and $w_k \in \mathbb{R}^s$ are disturbances, being $\mathcal{M} = \{1, 2, \dots, M\}$ a set of possible ‘modes’. System (8.1) switches between these M different operation modes, i.e., $A_{\theta_k} \in \bar{\mathcal{A}} := \{A_1, \dots, A_M\}$, $B_{\theta_k} \in \bar{\mathcal{B}} := \{B_1, \dots, B_M\}$, $E_{\theta_k} \in \bar{\mathcal{E}} := \{E_1, \dots, E_M\}$.

In this work, it is assumed that, at time k , the current mode θ_k is known, as well as the state x_k . Additionally, the operation mode θ_k follows a discrete-time Markov chain with transition probabilities matrix $\mathbb{P} = (\pi_{ji}) \in \mathbb{R}^{M \times M}$, i.e., $Pr\{\theta_{k+1} = j | \theta_k = i\} = \pi_{ji}$, were $\pi_{ji} \geq 0, \forall i, j \in M$ and $\sum_{j=1}^M \pi_{ji} = 1$. The mode matrices will be assumed time-invariant ($\bar{\mathcal{A}}$ and $\bar{\mathcal{B}}$ are constant) and π_{ji} will not change with time: $\bar{\mathcal{A}}, \bar{\mathcal{B}}$ and π_{ji} will be assumed known.

Constraints. Mode-dependent state ($\Omega^{[\theta_k]}$), input ($\mathbb{U}^{[\theta_k]}$) and disturbance ($\mathbb{W}^{[\theta_k]}$) constraint sets will be considered. The origin $x = 0$ is assumed to belong to all $\Omega^{[i]}$, and likewise $u = 0, w = 0$ will be assumed to belong to all $\mathbb{U}^{[i]}, \mathbb{W}^{[i]}$, respectively, for all $i \in \mathcal{M}$. Constraint sets are assumed compact and polytopic. Notation Ω^* will denote the set in the augmented space $\mathbb{R}^n \times \mathcal{M}$ given

by $\Omega^* := \{(x, \theta) \in \mathbb{R}^n \times \mathcal{M} : x \in \Omega^{[\theta]}\}$. Therefore, the mode-dependent constraints can be stated, for all $k \geq 0$, as $(x_k, \theta_k) \in \Omega^*$, $(u_k, \theta_k) \in \mathbb{U}^*$, $(w_k, \theta_k) \in \mathbb{W}^*$ being \mathbb{U}^* , \mathbb{W}^* likewise defined. Mode, state, input and disturbance sequences fulfilling the constraints will be denoted as *admissible*. Augmented-space *notation will be omitted if all $\Omega^{[i]}$, (or $\mathbb{U}^{[i]}$, $\mathbb{W}^{[i]}$) are equal. Notation $\mathcal{C} \subset \mathbb{C}(\mathbb{R}^n)$ will refer to \mathcal{C} being a polytopic set in \mathbb{R}^n , where convexity and polytopic structure has been assumed just for computational reasons, in order to use the multiparametric toolbox (Herceg *et al.* 2013).

Admissible sequences. \mathcal{S}_i will denote the set of all modes $j \in \mathcal{M}$ accessible from a mode $i \in \mathcal{M}$ in one time step, i.e., $\mathcal{S}_i := \{j \in \mathcal{M} | \pi_{ji} > 0\}$. An admissible switching sequence of length N , $\theta = \{\theta_0, \dots, \theta_{N-1}\}$ for (8.1) is a switching path for which $\theta_{k+1} \in \mathcal{S}_{\theta_k}$, for all $0 \leq k \leq N-2$. Equivalently, denoting as \mathcal{P}_i the predecessors of mode i , i.e., $\mathcal{P}_i := \{j \in \mathcal{M} | \pi_{ij} > 0\}$, admissible sequences will be those for which $\theta_k \in \mathcal{P}_{\theta_{k+1}}$ for $0 \leq k \leq N-2$. Notation $\mathcal{AS}(\theta_0, N)$ will denote the set of admissible sequences of length N starting with θ_0 .

One-step sets. When there is no mode information available to the controller, system (8.1) can be considered to be an uncertain linear system with polytopic uncertainty. In that context, the well-known *robust* 1-step set is defined (see Section 7.2.1). When the mode information is known by the controller, a more flexible definition of 1-step set is proposed by Equation (4.14), quoted below:

Definition 8.1 (Mode-dependent 1-step controllable set). *Given a set $\mathcal{C} \subset \mathbb{R}^n$, the mode-dependent one-step controllable set for mode $i \in \mathcal{M}$ and for systems (8.1) is defined as:*

$$\mathcal{Q}_i(\mathcal{C}) := \{x \in \Omega^{[i]} : \exists u \in \mathbb{U}^{[i]} \text{ such that } A_i x + B_i u + E_i w \in \mathcal{C} \forall w \in \mathbb{W}^{[i]}\} \quad (8.2)$$

and, also, $\mathcal{Q}_i(\emptyset) := \emptyset$. The **1-step set** in the augmented space Ω^* given by (8.3) is redefined, for $\mathcal{C}^* \subset \Omega^*$:

$$\mathcal{Q}^*(\mathcal{C}^*) := \{(x_0, i) \in \Omega^* : \exists u \in \mathbb{U}^{[i]} \text{ s. t.} \\ (A_i x_0 + B_i u + E_i w, j) \in \mathcal{C}^* \forall j \in \mathcal{S}_i, \forall w \in \mathbb{W}^{[i]}\} \quad (8.3)$$

Remark 8.1. *Knowledge of the mode i by the controller is implicitly integrated in (8.3). Analogously to well-known literature, (Kerrigan 2000), removing the existential quantifier (and plugging a predefined controller in) yields the 1-step sets needed*

for analysis-only set-ups, such as the implicitly considered one in Definitions 8.3 or 8.4, in later sections.

Definition 8.2. A set $\mathcal{C}^* \subset \Omega^*$ is *mode-dependent controllable* if $\mathcal{C}^* \subset \mathcal{Q}^*(\mathcal{C}^*)$. The largest of such sets is the *maximal mode-dependent controllable set*, \mathcal{K}_∞^* .

If \mathcal{C}^* is mode-dependent controllable, there exists a controller $u(x_k, \theta_k)$ such that trajectories starting in $(x_0, \theta_0) \in \mathcal{C}^*$ can indefinitely remain there, robustly for any admissible mode/disturbance sequence of arbitrary length.

Adapting algorithms in, for instance, (Kerrigan 2000), \mathcal{K}_∞^* can be computed iterating $\mathcal{K}_l^* = \mathcal{Q}^*(\mathcal{K}_{l-1}^*)$ until convergence, from initial $\mathcal{K}_0^* = \Omega^*$; see Section 7.3.1. Once the converged \mathcal{K}_∞^* is available, for any $(x, i) \in \mathcal{K}_\infty^*$, any controller which steers a state in $\mathcal{K}_\infty^{[i]}$ to the successor set:

$$\mathbb{S}^{[i]} := \bigcap_{j \in \mathcal{S}_i} \mathcal{K}_\infty^{[j]} \quad (8.4)$$

achieves the above-mentioned invariance. \mathcal{K}_∞^* will be denoted as *terminal set* and the associated controller as *terminal controller*²⁵.

If there exists a finite l such that $\mathcal{K}_l^* = \mathcal{K}_{l+1}^* = \dots = \mathcal{K}_\infty^*$, we will say that \mathcal{K}_∞^* has *l-step convergence* or, plainly, that the set is *finitely generated* when the actual value of l is not relevant. From robust invariant-set concepts, given $l \geq 0$, if $(x_0, \theta_0) \notin \mathcal{K}_l^*$ there exists a mode sequence of length $l + 1$ and a *worst-case* disturbance such that no admissible control action can avoid exiting Ω^* .

8.2.1. Problem statement

By definition, for initial mode-state conditions outside the terminal set \mathcal{K}_∞^* there does not exist a controller with probability of success equal to 1, i.e., trajectories starting from such initial states will violate constraints under some admissible mode and disturbance sequences. The objective of this paper is to optimise the ‘*reliability*’ understood as the likelihood²⁶ of not violating constraints in future time; it will be proved to be equal (in some cases) to the likelihood of reaching the terminal set from outside (and indefinitely remaining in it thereafter with a terminal controller).

²⁵In this work, what we call ‘terminal’ set is the ‘maximal controllable’ one in prior literature, and bears no direct relation to the terminal set in, say, predictive control (invariant, non-saturating LQR set (Costa *et al.* 2005)). However, conceptually, it is actually a ‘terminal’ one in later proposed horizon-based algorithms, because an optimal (reliability 1) admissible terminal controller is known inside such set.

²⁶No probability distribution will be assumed on the disturbances, so the results will be robust/worst-case regarding to them, and probabilities will be only associated to the Markov chain governing mode transitions. Some probabilistic set-ups in disturbances can be embedded in our framework by suitably defining the allowed mode-dependent disturbance sets $\mathbb{W}^{[i]}$, as in (Bernardini and Bemporad 2012).

This work will propose iterative algorithms, inspired on dynamic programming, to compute reliability bounds for initial $(x_0, \theta_0) \notin \mathcal{K}_\infty^*$ and an associated controller guaranteeing such bounds.

As later developments will show, there are initial states with zero reliability, i.e., those for which no control law can avoid constraint violation at future time (under some admissible worst-case disturbance). In fact, there are well-known cases in which *some* (or even *all*) of the terminal sets $\mathcal{K}_\infty^{[j]}$ are empty (large disturbances, non-stabilisable modes, etc.). In the latter case, reliability bound will be zero in all Ω^* because no initial state can be robustly kept indefinitely in Ω^* with a causal mode-dependent controller.

In states with zero reliability, a modified performance measure will be of interest: the ‘*robust mean time to failure*’, understood as the average number of steps in which the system starting at a particular state will violate constraints under a worst-case disturbance (precise definition later on).

Next section will define reliability and propose an algorithm for guaranteed-reliability controllers, and Section 8.4 will do the same for the mean-time-to-failure case, albeit briefly as developments will be parallel to those in Section 8.3.

8.3. Reliability Bound Computation

Definition 8.3. Given x_0 , a control law $u(x, \theta)$, an integer horizon k , and a mode sequence $\theta \in \mathcal{AS}(\theta_0, N + 1)$ of length $(N + 1) > k \geq 1$, the controller-sequence pair $\{u(\cdot, \cdot), \theta\}$ is **k -step successful** for $(x_0, \theta_0) \in \Omega^*$ if $(u(x_j, \theta_j), \theta_j) \in \mathbb{U}^*$ and $(x_{j+1}, \theta_{j+1}) \in \Omega^*$ for all w_j such that $(w_j, \theta_j) \in \mathbb{W}^*$, for all $j = 0, \dots, k - 1$. For a fixed controller $u(\cdot, \cdot)$, we will denote as $\mathcal{SS}(x_0, \theta_0, k)$ the set of admissible sequences θ with length $k + 1$ such that $\{u(\cdot, \cdot), \theta\}$ is k -step successful.

Basically, the above definition states that a k -step successful controller-sequence pair can avoid constraint violation for k steps robustly for all admissible disturbances (for a particular initial state). When there is no confusion on the controller under consideration, we will just say that a sequence θ is k -step successful for (x_0, θ_0) .

Note that, as an equivalent recursive characterisation, θ is k -step successful for $(x_0, \theta_0) \in \Omega^*$ if $(u(x_0, \theta_0), \theta_0) \in \mathbb{U}^*$, $(x_1(x_0, \theta_0, w), \theta_1) \in \Omega^*$ –being $x_1(x_0, \theta_0, w) := A_{\theta_0}x_0 + B_{\theta_0}u(x_0, \theta_0) + E_{\theta_0}w$ –, and $\{\theta_1, \dots, \theta_N\}$ is $(k - 1)$ -step successful for $(x_1(x_0, \theta_0, w), \theta_1)$ for all $w \in \mathbb{W}^{[\theta_0]}$.

Definition 8.4. Given a control law $u(x, \theta)$:

- For given $k \geq 1$, the controller's **k -step reliability** at initial conditions (x_0, θ_0) , denoted by $RL_k(x_0, \theta_0)$ is defined as

$$RL_k(x_0, \theta_0) := \sum_{\theta \in \mathcal{SS}(x_0, \theta_0, k)} Pr(\theta) \quad (8.5)$$

If $\mathcal{SS}(x_0, \theta_0, k)$ is empty or $(x_0, \theta_0) \notin \Omega^*$, reliability will be defined $RL_k(x_0, \theta_0) := 0$. Also, for any controller, $RL_0(x_0, \theta_0) := 1$ if $(x_0, \theta_0) \in \Omega^*$, and $RL_0(x_0, \theta_0) := 0$ otherwise.

- the controller's **reliability** at initial conditions (x_0, θ_0) , denoted as $RL(x_0, \theta_0)$, is defined as $RL(x_0, \theta_0) := \lim_{k \rightarrow \infty} RL_k(x_0, \theta_0)$.

If a sequence is k -step successful, trivially it must be $(k-1)$ -step successful; then, the set of length- $(k+1)$ sequences which are k -step successful is smaller than or equal to the set of length- $(k+1)$ sequences which are $(k-1)$ -step successful. So, inequality $0 \leq RL_k(x_0, \theta_0) \leq RL_{k-1}(x_0, \theta_0)$ is straightforward; subsequently, by monotonic-convergence argumentations, existence of the required limit in the definition of RL can be proved.

Informally, Definition 8.4 means that reliability is the probability of not violating constraints in future time from (x_0, θ_0) under *worst-case* disturbances. With \mathcal{K}_i^* in previous section (for the particular controller under scrutiny, see Remark 8.1), it can be proved that:

$$(x, \theta) \in \mathcal{K}_i^* \Leftrightarrow RL_l(x, \theta) = 1 \quad (8.6)$$

Proposition 8.1. For a given controller $u(x, i)$, reliability fulfils the following recursive equation:

$$RL_l(x, i) = \min_{w \in \mathbb{W}^{[i]}} \sum_{j \in \mathcal{S}_i} \pi_{ji} RL_{l-1}(A_i x + B_i u(x, i) + E_i w, j) \quad \forall l \geq 1 \quad (8.7)$$

Proof: Starting with RL_0 , it is straightforward to see that the assertion is true for RL_1 . For larger l , given $\theta = \{i, j, \theta_2, \dots\}$ and $u(x, \theta)$, the controller-sequence pair is l -step successful for $(x_0, i) \in \Omega^*$ if and only if $(u(x_0, i), i) \in \mathbb{U}^*$, and $(x_1, j) \in \Omega^*$, with $x_1 = A_i x + B_i u(x, i) + E_i w_0$ for all $w_0 \in \mathbb{W}^{[i]}$, and $\{u(x, \theta), \{j, \theta_2, \dots\}\}$ are $l-1$ -step successful for initial conditions (x_1, j) . The sum of $Pr(\cdot)$ of all $l-1$ -step successful sequences (noting that RL is zero if $(x_1, j) \notin \Omega^*$) multiplied by their conditional probability (conditioned to $\theta_0 = i$, i.e., π_{ji}) yields (8.7). So, the reasoning can be applied by induction to any $l \geq 1$. \square

As a corollary, if there exists l such that $RL_l(x, i) = RL_{l-1}(x, i)$ for all x and all i , then $RL_l(x, i) = RL(x, i)$.

8.3.1. Iterative $RL_l(\cdot, \cdot)$ Algorithm

Although reliability definition above considers a pre-existing control law, the aim of this section is to obtain an approximation to the maximum-reliability controller. Of course, optimal l -step reliability, denoted as RL_l^{opt} , must verify the Bellman condition:

$$RL_l^{opt}(x, i) = \max_{u \in \mathbb{U}^{[i]}} \min_{w \in \mathbb{W}^{[i]}} \sum_{j \in \mathcal{S}_i} \pi_{ji} RL_{l-1}^{opt}(A_i x + B_i u + E_i w, j) \quad (8.8)$$

However, obtaining such optimal-reliability controller might require a large computational effort. As discussed in the introduction, Chapter 7 introduces a greedy approach, which this paper tries to improve in an algorithm which exploits the dynamic-programming structure of the problem.

Note that each state x will have a different reliability, depending on the mode the system is at when x is reached, requiring the control law to drive the next state to a different target state depending on such mode. So, a structure containing sets of states, mode-dependent value functions (reliability estimates) and mode-dependent target successor sets is needed. This motivates the following notation:

Auxiliary Notation. Let \mathcal{F} denote the mappings $\mathcal{M} \mapsto \mathbb{R}_+ \times \mathbb{C}(\mathbb{R}^n)$, being \mathbb{R}_+ the set of non-negative real numbers. Given $f \in \mathcal{F}$, $j \in \mathcal{M}$, we will denote the components of $f(j)$ as $f(j) := (f^{val}(j), f^{suc}(j))$ with $f(j)^{val} \in \mathbb{R}_+$, $f(j)^{suc} \in \mathbb{C}(\mathbb{R}^n)$.

Let us denote as $\mathcal{O} := \mathbb{C}(\mathbb{R}^n) \times \mathcal{F}$. The component fields of object $o \in \mathcal{O}$ will be denoted as $o_{set} \in \mathbb{C}(\mathbb{R}^n)$, $o_f \in \mathcal{F}$, i.e., $o = \langle o_{set}, o_f \rangle$. Thus, an element $o \in \mathcal{O}$ will have the following data structure:

$$o = \langle o_{set}, o_f \rangle = \left\langle o_{set}, \left\{ \begin{array}{cc} o_f^{val}(1), & o_f^{suc}(1) \\ \vdots & \vdots \\ o_f^{val}(M), & o_f^{suc}(M) \end{array} \right\} \right\rangle \quad (8.9)$$

Given $o, q \in \mathcal{O}$, notation $o \succ q$ (equivalently, q **subset-dominates** o) will be used when $o_{set} \supset q_{set}$ and $o_f^{val}(i) \leq q_f^{val}(i)$ for all $i \in \mathcal{M}$.

Notation $q \blacktriangleleft o$ (o **dominates** q) will mean $o_{set} \supset q_{set}$, $o_f^{val}(i) \geq q_f^{val}(i)$, for all i .

Two operators on elements of \mathcal{O} will be defined next.

Intersection operations over \mathcal{O} . First, the obvious action when a point belongs to two sets, with different possible controllers achieving different reliability bounds, would be to take the maximum-reliability option. This motivates the intersection operator below:

Definition 8.5 (*I-operator*). Let $o, q \in \mathcal{O}$. The operator $I(o, q) : \mathcal{O} \times \mathcal{O} \mapsto \mathcal{O}$ is defined as: $I(o, q) := (o_{set} \cap q_{set}, \max(o_f, q_f))$, where, for given $f, g \in \mathcal{F}$, the maximum $l = \max(f, g)$ is the element $l \in \mathcal{F}$ whose image, for $j \in \mathcal{M}$ is:

$$h(j) := \begin{cases} (f^{val}(j), f^{suc}(j)) & f^{val}(j) \geq g^{val}(j) \\ (g^{val}(j), g^{suc}(j)) & \text{otherwise} \end{cases} \quad (8.10)$$

The *I-operator* will be informally denoted as ‘intersection’ because it behaves similarly to such set operator. Indeed, it is trivial to prove that:

- (a) the *I-operator* is associative, i.e., $I(o, q, r) := I(o, I(q, r)) = I(I(o, q), r)$ for all $o, q, r \in \mathcal{O}$,
- (b) for any $o, q \in \mathcal{O}$: $q \succ I(o, q)$, $o \succ I(o, q)$,
- (c) if $o \succ q$ then $I(o, q) = q$ and, for any $r \in \mathcal{O}$, $o \succ I(q, r)$, and $I(o, r) \succ I(q, r)$.

One-step operations over \mathcal{O} . Abusing the notation, the one-step set $\mathcal{Q}_i(\mathcal{C})$ in Definition 8.1 will be naturally extended to \mathcal{O} , formalising the intuitive idea that, if a target set can be reached from another one, the target’s reliability bounds can be propagated to the latter set as follows:

Definition 8.6 (*1-step operator over \mathcal{O}*). The 1-step operator $\mathcal{Q}_i : \mathcal{O} \mapsto \mathcal{O}$, will be defined, for $o \in \mathcal{O}$, as: $\mathcal{Q}_i(o) := (\mathcal{Q}_i(o_{set}), \mathcal{Q}_i^f(o_f, o_{set}))$, where the operator $\mathcal{Q}_i^f(r, s)$ for $i \in \mathcal{M}$, $r \in \mathcal{F}$, $s \in \mathbb{C}(\mathbb{R}^n)$ is defined as the element $g \in \mathcal{F}$ below:

$$g(j) := \begin{cases} (0, \mathbb{R}^n) & j \neq i \\ (\sum_{\kappa \in \mathcal{S}_i} \pi_{\kappa i} r^{val}(\kappa), s) & j = i \end{cases} \quad (8.11)$$

The justification of the structure (8.9) for \mathcal{O} is as follows. Given $o \in \mathcal{O}$ we have a ‘set’ o_{set} , and a function $o_f \in \mathcal{F}$, which is itself composed of $o_f^{val} : \mathcal{M} \mapsto \mathbb{R}_+$ and $o_f^{suc} : \mathcal{M} \mapsto \mathbb{C}(\mathbb{R}^n)$. The first component will take the role of the ‘value function’ (reliability bound if landing at o_{set} for each mode), whilst the second component will be the current policy’s ‘(mode-dependent) successor set’, which will generalise to states outside \mathbb{K}_∞^* the successor proposed in (8.4) for the terminal controller.

Guaranteed-reliability controller algorithm. From the above considerations, Algorithm 8 is proposed in order to obtain a progressively more accurate bound for the reliability. Basically, it will start with Ω^* and RL_0 at ‘iteration 0’, and it will apply the 1-step operator to each of its elements, as well as the I -operator to the resulting 1-step operations (all combinations²⁷). This will yield additional elements of \mathcal{O} , and all of them will be grouped in a set $O^{[1]} \subset \mathcal{O}$, and, repeating the procedure, we will produce $O^{[2]}, O^{[3]}$ and so on.

Algoritmo 8 l -step reliability bound.

1. Let $l = 1$. Let $O^{[0]}$ be a M -element set $\{o_1, \dots, o_M\}$ such that:

$$(o_j)_{set} = \Omega^{[j]}; (o_j)_f^{val}(j) = 1, \text{ and } (o_j)_f^{val}(i) = 0 \text{ (for } i \neq j);$$

$$\forall i, (o_j)_f^{suc}(i) = \mathbb{R}^n.$$
2. Compute the relevant 1-step set for all elements of $O^{[l-1]}$, conforming a set $D^{[l]} \subset \mathcal{O}$ as:

$$D^{[l]} := \bigcup_{o \in O^{[l-1]}, j \in \mathcal{M}} \mathcal{Q}_j(o), \quad (8.12)$$

3. Compute the intersections for all obtained elements of $D^{[l]}$, initialising with $Y_0^{[l]} = \emptyset$ and repeating until $Y_l^{[l]} = Y_{l-1}^{[l]}$, increasing l :

$$Y_l^{[l]} := \bigcup_{o_1, o_2 \in D^{[l]} \cup Y_{l-1}^{[l]}} I(o_1, o_2) \quad (8.13)$$

4. Update:

$$O^{[l]} := D^{[l]} \cup Y_l^{[l]} \quad (8.14)$$

5. Clean up: if there exist $o, q \in O^{[l]}$, such that $q \blacktriangleleft o$, or $q_{set} = \emptyset$, or $\max_j q^{val}(j) = 0$, remove q from $O^{[l]}$.
 6. If $O^{[l]} = O^{[l-1]}$, then success=TRUE, and let $l = l - 1$, $O^{[\infty]} = O^{[l]}$. END.
 7. Let $l = l + 1$. If $l < l_{MAX}$ go to Step 2. Else, let success=FALSE. END.
-

Taking the algorithm’s output, given x , let us define a collection of objects $O_x^{[l]} \subset \mathcal{O}$ as $O_x^{[l]} := \{o \in O^{[l]} : x \in o_{set}\}$. Let us, too, denote by $I(O_x^{[l]}) \in \mathcal{O}$ the single element of \mathcal{O} defined to be the I -operator of all elements of $O_x^{[l]}$, where the order of intersection is irrelevant, by associativity. Note that recursion (8.13) ensures $I(O_x^{[l]}) \in O^{[l]}$.

Theorem 8.1. *Given any $o \in O^{[l]}$ from Algorithm 8, and initial condition (x_0, θ_0) ,*

²⁷This is a costly step, requiring a number of operations which grows exponentially with the number of elements of $D^{[l]}$; the Appendix will discuss some alternatives to alleviate the load. The algorithm as it stands in this subsection must be thought of in a ‘formal’ sense, and not as an implementation proposal.

1. if $o_f^{val}(\theta_0) > 0$, there exists a control law which sends x_0 to $o_f^{suc}(\theta_0)$ and achieves l -step reliability RL_l of at least $o_f^{val}(\theta_0)$, for $l \geq 0$.

2. Define:

$$\widehat{RL}(x_0, \theta_0, l) := \max_{o \in O_{x_0}^{[l]}} o_f^{val}(\theta_0), \quad \hat{o}(x_0, \theta_0, l) := \arg \max_{o \in O_{x_0}^{[l]}} o_f^{val}(\theta_0) \quad (8.15)$$

Then, if $\widehat{RL}(x_0, \theta_0, l) > 0$, there exists a control law which achieves l -step reliability of at least $\widehat{RL}(x_0, \theta_0, l)$, for $l \geq 0$. Note that, if the maximum above is reached at several elements of $O_{x_0}^{[l]}$, \hat{o} must be understood as any arbitrary choice of one of them.

Proof: First, note that $\widehat{RL}(x_0, \theta_0, 0) = RL_0(x_0, \theta_0)$ by the algorithm initialisation. So, there exists a controller (actually any arbitrary controller) which achieves such reliability bound for RL_0 . Now, consider the theorem to be true for $l - 1, l \geq 1$.

Considering an arbitrary element o of $O^{[l-1]}$, let $i = \theta_0, q = \mathcal{Q}_i(o) \in D^{[l]}$. Then, if $x_0 \in q_{set}$, there exists a control law u which sends the next state $x_1 = A_i x_0 + B_i u + E_i w$ to o_{set} for all admissible disturbances if initial mode is i . Consider now that such u is applied at the present instant, and that the controller arising from o will be used at next sample. Therefore, we can assert that likelihood of success in l steps if such control law were applied, $RL_l(x, i)$, is, at least, the sum (over all admissible successors $j \in \mathcal{S}_i$) of the conditional probability of being in mode j (i.e., π_{ji}) multiplied by the $(l-1)$ -step reliability estimate stored in o , i.e.,

$$RL_l(x, i) \geq \sum_{j \in \mathcal{S}_i} \pi_{ji} o_f^{val}(j) \quad (8.16)$$

Note that the right-hand side of the above inequality is actually stored in q_f (and the corresponding successor set), see (8.11). This proves the first assertion in the theorem statement.

Of course, if the state x_0 belongs to several 1-step sets \mathcal{Q}_i from different elements of $O^{[l-1]}$, the above reasoning can be made for each of them. The definitions in (8.15) just take the element $o \in O_{x_0}^{[l]}$ yielding the largest reliability estimate. Last, an induction argumentation ensures that the theorem holds for any positive l . \square

Note that, by definition of $I(O_{x_0}^{[l]})$, and the fact that $I(O_{x_0}^{[l]}) \in O^{[l]}$, the choice of \hat{o} in (8.15) could be always made to be $\hat{o}(x_0, \theta_0, l) = I(O_{x_0}^{[l]})$ for any θ_0 because $I(O_{x_0}^{[l]})$ stores the best option for all modes, i.e., $o \succ I(O_{x_0}^{[l]})$ for all $o \in O_{x_0}^{[l]}$. This idea will be used in the proof below.

Theorem 8.2. *In the disturbance-free case, the algorithm yields the optimal reliability and associated optimal controller, i.e., $\widehat{RL}(x_0, \theta_0, l) = RL_l^{opt}(x_0, \theta_0)$.*

Proof: First, note that $\widehat{RL}(x_0, \theta_0, 0) = RL_0^{opt}(x_0, \theta_0) = RL_0(x_0, \theta_0)$ by the algorithm initialisation, and RL_0 is identical for any controller. Now, assume the theorem to be true for $l-1, l \geq 1$. Let $i = \theta_0$ and denote by $q(x, l) := I(O_x^{[l]})$, $x_1 := A_i x_0 + B_i u$. Hence, without disturbances, (8.8) gets converted to:

$$RL_l^{opt}(i, x_0) = \max_{u \in \mathbb{U}^{[i]}} \sum_{j \in \mathcal{S}_i} \pi_{ji} RL_{l-1}^{opt}(A_i x_0 + B_i u, j) = \max_{u \in \mathbb{U}^{[i]}} \sum_{j \in \mathcal{S}_i} \pi_{ji} \hat{o}_f^{val}(A_i x_0 + B_i u, j, l-1) \quad (8.17)$$

where the last equality is stated by assumption. Now,

$$\begin{aligned} \max_{u \in \mathbb{U}^{[i]}} \sum_{j \in \mathcal{S}_i} \pi_{ji} \hat{o}_f^{val}(A_i x_0 + B_i u, j, l-1) &= \max_{u \in \mathbb{U}^{[i]}} \sum_{j \in \mathcal{S}_i} \pi_{ji} \max_{o \in O_{x_1}^{[l-1]}} o_f^{val}(j) \\ &= \max_{u \in \mathbb{U}^{[i]}} \sum_{j \in \mathcal{S}_i} \pi_{ji} \cdot [q(x_1, l-1)]_f^{val}(j) = \max_{u \in \mathbb{U}^{[i]}} \max_{o \in O_{x_1}^{[l-1]}} \sum_{j \in \mathcal{S}_i} \pi_{ji} \cdot o_f^{val}(j) \end{aligned} \quad (8.18)$$

because the elements \hat{o} (which might be different, depending on j) have a value function for mode j identical to a *single* element $q(x_1, l-1) \in O_{x_1}^{[l-1]}$ due to the I -operator computations in the Algorithm. As $o \succ q(x_1, l-1)$ for all $o \in O_{x_1}^{[l-1]}$, so we can search over all $O_{x_1}^{[l-1]}$, yielding the last equality.

Note now that, given any $o \in O_{x_1}^{[l-1]}$, there exists u such that x_0 can be steered to some $x_1 \in o_{set}$ if and only if $x_0 \in \mathcal{Q}_i(o_{set})$. Hence,

$$\begin{aligned} \max_{u \in \mathbb{U}^{[i]}} \max_{o \in O_{x_1}^{[l-1]}} \sum_{j \in \mathcal{S}_i} \pi_{ji} \cdot o_f^{val}(j) &\stackrel{*}{=} \max_{o \in O^{[l-1]} \text{ s.t. } x_0 \in \mathcal{Q}_i(o_{set})} \sum_{j \in \mathcal{S}_i} \pi_{ji} \cdot o_f^{val}(j) = \\ &\max_{o \in O_{x_0}^{[l]}} o_f^{val}(i) = \widehat{RL}(i, x_0, l) \end{aligned} \quad (8.19)$$

An induction argument ends the proof. \square

The equality in (8.19) marked with $'\stackrel{*}{=}'$ does not hold in disturbed cases.

Corollary 8.1. *In the undisturbed case, if Algorithm 8 converges in a finite number of steps, considering the minimum l such that $O^{[l]} = O^{[l-1]}$, the resulting $\widehat{RL}(x_0, \theta_0, l)$ is equal to $RL^{opt}(x_0, \theta_0)$.*

Proof: Indeed, in such a case $RL^{opt}(x_0, \theta_0) = RL_l^{opt}(x_0, \theta_0) = RL_{l-1}^{opt}(x_0, \theta_0)$, so the last iteration has obtained the optimal reliability. \square

It might be the case that the algorithm does not converge in a finite number

of steps. In such a case, a controller guaranteeing the state to stay only for a finite number of samples inside Ω^* will be obtained. Although it is faithful to the RL_l definition, in most cases l will be just a handful of samples, which will not be meaningful for applications, because as $RL \leq RL_l$ the optimal (non-converged) RL_l controllers might have *zero* long-term reliability (and, moreover, in the disturbed case the obtained controllers are themselves suboptimal). These issues motivate the next subsection.

8.3.2. Terminal Set Based Algorithm

In case of not having enough computational resources for convergence, if a subset of the terminal set $\mathbb{S}^* \subset \mathcal{K}_\infty^*$ is available, the algorithm could be seeded replacing initialisation in step 1 by a initial value function ('reward') equal to one for being inside \mathbb{S}^* , and zero elsewhere. This is the idea exploited in Chapter 7: forcing the terminal set to be reached for at least one sequence. \mathbb{S}^* may be generated by a finite λ -contractive search (Blanchini 1994, Theorem 3.2), or via LMIs (Kothare *et al.* 1996). In this case, as $\widehat{RL}(x, \theta, 0) \leq RL_0(x, \theta)$, dynamic programming propagation would, trivially, obtain an estimate $\widehat{RL}(x, \theta, l)$ lower than the original Algorithm 8. Let us discuss such issues in more detail.

For a given controller $u(x, \theta)$, let us denote the likelihood of reaching \mathcal{K}_∞^* , under a worst-case disturbance, in *at most* l steps as $RT_l(x_0, \theta_0)$, and

$$RT(x_0, \theta_0) := \lim_{l \rightarrow \infty} RT_l(x_0, \theta_0)$$

Evidently, as all finite sequences reaching the terminal set will be l -step successful for any arbitrary large l , we can trivially assert that

$$RT_l(x_0, \theta_0) \leq RT_{l+1}(x_0, \theta_0) \leq RT(x_0, \theta_0) \leq RL(x_0, \theta_0)$$

for all $l \geq 0$. So, RT_l provides lower bounds on the reliability whereas RL_l provide upper bounds. However, the following result states that such bounds are equal in the finitely generated case.

Theorem 8.3. *Given (x_0, θ_0) , and a control law $u(x, \theta)$, if the terminal set associated to the controller, \mathcal{K}_∞^* is finitely generated, then $RT(x_0, \theta_0) = RL(x_0, \theta_0)$.*

Proof: In order to prove the above assertion, some notation will be introduced. First, given (x_0, θ_0) , and a control law $u(x, \theta)$, we will consider an infinite-length mode sequence to be *successful* if it is k -step successful for (x_0, θ_0) for any finite k . So, $RL(x_0, \theta_0)$ is the probability of the set of successful sequences. Also, let us denote by $\widehat{\Theta}_l(x_0, \theta_0)$ the set of length- $(h + 1)$ sequences which:

(1) are h -step successful for (x_0, θ_0) , and (2) there exists a disturbance sequence $w_j(\boldsymbol{\theta})$, $j = 0, \dots, h - 1$, for every sequence $\boldsymbol{\theta} \in \widehat{\Theta}_l(x_0, \theta_0)$, such that the controller in consideration keeps (x_j, θ_j) in $(\Omega^* \sim \mathcal{K}_l^*)$, for $j = 0, \dots, h$.

Consider l to be the smallest integer such that $\mathcal{K}_\infty^* = \mathcal{K}_l^*$. Successful mode sequences, i.e., which do *not* lead to failure in any finite time (robustly keeping the state inside Ω^*), can either:

1. make state reach the terminal set in finite time for some (or all) disturbance realisations (and remain there ever after because the controller makes \mathcal{K}_l^* robustly invariant), or
2. remain for infinite time in $\Omega^* \sim \mathcal{K}_l^*$ (for the rest of disturbance realisations, which do not reach the terminal set in finite time).

Let us prove that the second case has zero probability: if such assertion is proved, RL will be equal to the probability of reaching the terminal set, as asserted in the theorem.

Indeed, as discussed in Section 8.2, if (x_0, θ_0) is not in \mathcal{K}_l^* , there exists at least one mode sequence of at most length $l + 1$ with non-zero probability, and a worst-case disturbance such that constraint violation is unavoidable. Evidently, the probability of such mode sequence is greater than

$$\rho_l := \min_{\boldsymbol{\theta} \in \mathcal{M}} \min_{\boldsymbol{\theta} \in AS(\boldsymbol{\theta}, l+1)} Pr(\boldsymbol{\theta}),$$

and, hence, the cumulative probability (sum) of all non l -step successful sequences is, too, greater than ρ_l so:

$$(x_0, \theta_0) \in (\Omega^* \sim \mathbb{K}_l^*) \Rightarrow RL_l(x_0, \theta_0) \leq 1 - \rho_l < 1 \quad (8.20)$$

From this argumentation, $Pr(\widehat{\Theta}_l(x_0, \theta_0)) \leq RL_l(x_0, \theta_0) \leq 1 - \rho_l$, where, abusing the notation, the probability of a set of sequences is understood as the sum of individual sequence probabilities, conditional to their first element.

Let us now show that $Pr(\widehat{\Theta}_h)$ tends to zero as h tends to infinity, for $h = l, 2l, 3l, \dots$

Note that, considering length- $(2l + 1)$ sequences, for such sequences to be $(2l)$ -step successful for (x_0, θ_0) , forcedly they must be the concatenation of a l -step successful sequence for (x_0, θ_0) and another l -step successful one for

(x_l, θ_l) . Then, we have:

$$\begin{aligned}
 Pr(\widehat{\Theta}_{2l}(x_0, \theta_0)) &= \sum_{\theta \in \widehat{\Theta}_{2l}(x_0, \theta_0)} Pr(\theta) = \\
 &= \sum_{\theta \in \widehat{\Theta}_l(x_0, \theta_0)} \left(Pr(\theta) \sum_{\theta' \in \widehat{\Theta}_l(x_l, \theta_l)} Pr(\theta') \right) \\
 &\leq \sum_{\theta \in \widehat{\Theta}_l(x_0, \theta_0)} (Pr(\theta)(1 - \rho_l)) \leq (1 - \rho_l)^2 \quad (8.21)
 \end{aligned}$$

Indeed, in the above expression, the following fact has been used: the probability of a length- $(2l + 1)$ sequence $\theta := \{\theta_0, \theta_1, \dots, \theta_{2l}\}$ being $2l$ -step successful for (x_0, θ_0) requires:

1. $\{\theta_0, \dots, \theta_l\}$ being l -step successful for (x_0, θ_0) , and
2. $\{\theta_l, \dots, \theta_{2l}\}$ being l -step successful for (x_l, θ_l) .

and the bound for the sum of probabilities for (x_l, θ_l) is, too, $(1 - \rho_l)$ because $\mathcal{K}_l^* = \mathcal{K}_{2l}^* = \mathcal{K}_\infty^*$ and, by definition of $\widehat{\Theta}_{2l}$, (x_j, θ_j) lies in $\Omega^* \sim \mathcal{K}_\infty^*$ for $j = 0, \dots, 2l$.

Now, if $l = \nu \cdot l$, being ν any arbitrary natural number, a similar argumentation can prove that:

$$Pr(\widehat{\Theta}_{\nu \cdot l}(x_0, \theta_0)) \leq (1 - \rho_l)^\nu$$

Hence, letting ν tending to infinity, we can say that the cumulative probability of ‘all successful sequences for which there exists a disturbance indefinitely keeping the state in a set with reliability not equal to one’, is zero. Hence, the probability of violating constraints under worst-case disturbance (i.e., $1 - RL$) and the probability of reaching the terminal set (RT) add 1, so $RT(x_0, \theta_0) = RL(x_0, \theta_0)$. \square

Basically, the above theorem states that a controller with non-zero reliability cannot keep state wandering outside the terminal set forever because, eventually, the worst-case sequence (which is finite-length and, hence, has non-zero probability) *will* appear, with probability one.

In conclusion, Algorithm 8 had been presented due to the direct relationship with RL_l and the MTTF bound to be defined in next section, as well as because of the optimality under finite-step convergence in an undisturbed case. However, the referred algorithm, if unmodified, only provides an upper

bound of the reliability if not converged, which has little use. On the other hand, terminal-set-based initialisation provides a guaranteed lower bound on the reliability even if *not* converged, which is, of course, more useful in practice. Section 8.5 discusses in more detail such modified algorithm, and some options to alleviate the computational load by prioritising promising operations, used in the examples in Section 8.6.

8.4. Mean time to failure bound computation

Definition 8.7. Given a control law $u(x_k, \theta_k)$, initial conditions (x_0, θ_0) and $\theta \in \mathcal{AS}(\theta_0, N+1)$, the sequence's **guaranteed time to failure** $k^*(x_0, \theta)$ is defined as either the minimum natural number k such that the pair $\{u, \theta\}$ is not k -step successful for (x_0, θ_0) , or $(N+1)$ if it is N -step successful, or zero if $(x_0, \theta_0) \notin \Omega^*$.

Definition 8.8. Given a control law $u(x_k, \theta_k)$, the **robust mean time to failure** $MTTF(x_0, \theta_0)$ is defined as:

$$MTTF(x_0, \theta_0) := \lim_{l \rightarrow \infty} M_l(x_0, \theta_0)$$

where

$$M_l(x_0, \theta_0) := \sum_{\theta \in \mathcal{AS}(\theta_0, l+1)} k^*(x_0, \theta) \cdot Pr(\theta) \quad (8.22)$$

As in Definition 8.4, MTTF involves worst-case (non-random) disturbances. Simple modifications to the set-up in Section 8.3 will be needed to handle the new performance measure. Actually, the needed changes are twofold: first, replacing $\mathcal{Q}_i^f(r, s)$ in Definition 8.6 by the one in Definition 8.9 below and, second, modifying algorithm initialisation. Let us discuss such issues in detail.

Definition 8.9 (1-step MTTF operator on \mathcal{F}). Let $r \in \mathcal{F}$, $s \in \mathbb{C}(\mathbb{R}^n)$. The operator $\mathcal{Q}_i^f(r, s)$ for $i \in \mathcal{M}$ is defined as the element $g \in \mathcal{F}$ below:

$$g(j) := \begin{cases} (0, \mathbb{R}^n) & j \neq i \\ \left(1 + \sum_{k=1}^M \pi_{ki} r^{val}(k), s\right) & j = i \end{cases} \quad (8.23)$$

The modified 1-step operator is motivated by the fact that the recursion (8.7), for a given controller, in the MTTF case changes to:

$$M_l(x, i) \geq 1 + \min_{w \in \mathbb{W}^{[i]}} \sum_{j \in \mathcal{S}_i} \pi_{ji} M_{l-1}(A_i x + B_i u + E_i w, j)$$

because if x_1 is reached from x_0 , then the guaranteed time to failure of x_0 is at least 1 step longer. So, if the state can be robustly driven to a set where $MTTF$

can be lower bounded by some $r^{val}(j)$, propagating the bounds backward will yield improved bounds, parallel to the reliability case considered in prior sections (details left to the reader).

Initialization. Algorithms can run unmodified, but initialised at

$$O = \{o_1, \dots, o_M, q_1, \dots, q_M\},$$

with o_i being identical to the one in Algorithm 8, and q_j defined as:

$$(q_j)_{set} \subset \mathcal{K}_\infty^{[j]}; (q_j)_f^{val}(j) = \infty, (q_j)_f^{suc}(j) \subset \mathbb{S}_i, \text{ and for } j \neq i, (q_j)_f^{val}(i) = 0, (q_j)_f^{suc}(i) = \mathbb{R}^n.$$

Relationship between MTTF and RL. The two performance measures are related by the following proposition:

Proposition 8.2. *For a given controller $u(x_k, \theta_k)$, $MTTF(x_k, \theta_k)$ is finite if and only if $RL(x_k, \theta_k) = 0$.*

Proof: For any l , we can assert, from (8.22), that

$$M_l(x_0, \theta_0) = RL_l(x_0, \theta_0) * (l + 1) + \sum_{\theta \in \mathcal{AS}(\theta_0, l+1): k^*(x_0, \theta) < (l+1)} k^*(x_0, \theta) \cdot Pr(\theta) \quad (8.24)$$

because RL_l is computed only with those sequences which are l -step successful so their guaranteed time to failure is $l + 1$. Hence, $M_l \geq RL_l * (l + 1)$.

If $RL > 0$, as $RL_l > RL$ for any l , $RL > 0$ implies $MTTF = \lim_{l \rightarrow \infty} M_l = \infty$. Conversely, we can write $RL_l \leq M_l / (l + 1)$, so, taking the limit $l \rightarrow \infty$, if $\lim_{l \rightarrow \infty} M_l$ is finite (by assumption), then forcedly $RL = 0$. \square

8.5. Terminal-set algorithm and implementation guidelines

Consider that a subset of the terminal set $S^* \subset \mathcal{K}_\infty^*$ is available. Then, as discussed in the main text, a modified algorithm can be crafted by replacing initialisation in step 1 of Algorithm 8 by new sets and value functions, (see initialisation of Algorithm 9 on page 184; actually, related to the terminal initialisation in the MTTF version). Also, as the definition of RT_l before Theorem 8.3 involve RT_l being the likelihood of reaching the terminal set in *at most*

l steps, this means that expression (8.14) must be replaced by:

$$O^{[l]} := D^{[l]} \cup Y_l^{[l]} \cup O^{[l-1]} \quad (8.25)$$

i.e., previous estimates of the likelihood of reaching the terminal set in a shorter time *should be kept*. With these modifications, the 1-step and I -operations will store a value function which will be a lower bound of the likelihood of reaching the terminal set RT (details left to the reader).

Importantly, all found control laws driving states in a certain set to a mode-dependent successor set will ensure, if successful, infinite lifetime once the terminal set is reached.

The main issue with the proposal in this paper is the computational cost, regarding the exponentially increasing number of set operations which is needed. Some implementation-oriented modifications, aiming to reduce the number of operations or prioritising them, will be incorporated to the new algorithm. Note that these computational issues are not exclusive to our proposal, for instance many contributions on predictive control of MJLS have the same problem (Patrinos *et al.* 2014), needing to resort to the so-called “scenario approach” (Bernardini and Bemporad 2012). Also, in high-dimensional systems, the projection step in computing the controllable sets is very costly. Again, this issue is common to widely-cited papers such as the control-invariant set computation in above-referred (Patrinos *et al.* 2014).

Improved implementation.

The proposition below discusses how to update the value function when a invariant set has been obtained (in the geometric sense).

Proposition 8.3. *Consider $o \in \mathcal{O}$, $q = \mathcal{Q}_i(o)$ for any arbitrary $i \in \mathcal{M}$. If $o_{set} \subset q_{set}$, then the component o_f can be set to $\psi_\infty := \lim_{k \rightarrow \infty} \psi_k$, obtained from recursion $\psi_{k+1} := \max(\psi_k, \mathcal{Q}_i^f(\psi_k, o_{set}))$, starting with $\psi_0 := o_f$. The element \tilde{o} , with $\tilde{o}_{set} := o_{set}$, $\tilde{o}_f := \psi_\infty$ verifies $o \triangleleft \tilde{o}$. Also, defining $\tilde{q} := \mathcal{Q}_i(\tilde{o})$, we have $q \triangleleft \tilde{q}$.*

Proof: Given ψ_0 , then ψ_1 is the value/successor function of $r_1 = I(o, \mathcal{Q}_i(o))$, where $r_{1,set} = o_{set}$. Then, ψ_2 is the one for $r_2 = I(r_1, \mathcal{Q}_i(r_1))$, and so on again, it can be repeated until no improvement occurs. \square

Note that $r_{k,set} = o_{set}$ and, evidently, if $o_{set} \subset q_{set}$, there exists a control action such that o_{set} is invariant under mode i , so we can set itself as its successor if that were the result in ψ_∞ . Thus, in case the above proposition were used in an algorithm modification, the optimal successors in $O^{[l]}$

might not lie in $O^{[l-1]}$ but in $O^{[l]}$ itself, opening the possibility of reaching the terminal set in an arbitrarily long time (although such event has probability tending to zero as time increases, Theorem 8.3). Note, too, that ψ_∞ has an straightforward explicit expression, without the need of actually iterating. For instance, in a system with 3 operating modes, if there exists a set S with $RT(1, x) \geq 0$, $RT(2, x) \geq 0,98$, $RT(3, x) \geq 0,8$, which happens to be invariant with mode 1, i.e., $\mathcal{Q}_1(S) \supset S$, then the above proposition allows to assert $RT(1, x) \geq (0,98\pi_{21} + 0,8\pi_{31})/(1 - \pi_{11})$ too, for $x \in S$. The controller arising from the application of Proposition 8.3 would remain in S under mode 1 (for as long as needed) until mode 2 or 3 occurred.

Avoiding repeated operations.

As $O^{[l-1]} \subset O^{[l]}$, in the modified update (8.25), in order to avoid recomputing 1-step or I operations, the unordered horizon-based set $O^{[l]}$ will be replaced with an *ordered list*, denoted plainly as O , whose element at position α will be denoted as $O(\alpha)$. New objects will be appended at the end of O . Some labels will be introduced (Algorithms 9 and 10): matrices will be defined whose elements will be a label taken from: {'pending', 'dom', 'same', 'done'}. Obviously, the label 'pending' will be used to denote pending operations, then:

- Element $V(j, \alpha)$ of a matrix V of dimensions $\text{card}(O) \times M$ will be changed from 'pending' to 'done' once $\mathcal{Q}_j(O(\alpha))$ has been computed.
- Regarding I -operator, an upper triangular matrix T of dimensions $\text{card}(O) \times \text{card}(O)$ will be used to store the progress information. A labelling algorithm will be later discussed.

The changes to the original algorithm in the initialisation and list management appear as Algorithm 9, which carries out the 1-step and intersection operations one by one, and analyses the resulting object in order to determine whether some of the 'pending' operations are actually needed.

The construction of T in the above algorithms allows to state:

1. If $T(k, j) = \text{'dom'}$, $k < j$, then $O(k) \succ O(j)$.
2. If $T(k, j) = \text{'same'}$, then there exists a collection of elements on the ordered list O , prior to item j , such that $I(O(k), O(j))$ can be obtained via a finite set of operations on them (i.e., there is another way to obtain the 'same' result).

Algoritmo 9 Terminal-set based reliability bound

1. Let $O^{[0]}$ be a M -element set $\{o_1, \dots, o_M\}$ such that:
 $(o_j)_{set} \subset \mathcal{K}_{\infty}^{[j]}$, $(o_j)_f^{val}(j) = 1$, $(o_j)_f^{suc}(j) \subset \mathbb{S}_j$ and, for $j \neq i$, $(o_j)_f^{val}(i) = 0$,
 $(o_j)_f^{suc}(i) = \mathbb{R}^n$.
 2. Initialise V and T to 'pending', initialise N to zero. Let $l = card(O)$.
 3. Choose whether to carry out either
 (a) a pending 'intersection' between two elements $q = I(O(\alpha), O(\beta))$, or
 (b) a 1-step operation $q = \mathcal{Q}_j(O(\alpha))$,
 according to any criteria (either randomly or with probability-based choices to be later discussed).
 4. Label $V(j, \alpha) = \text{'done'}$ in case (b), or $T(\alpha, \beta) = \text{'done'}$ in case (a).
 5. If $q_{set} = \emptyset$ or there exists $r \in O$ such that $r_{set} \supseteq q_{set}$ and $max(r_f, q_f) = r_f$ then go to step 3.
 6. Let $l = h + 1$, let $O(h) = q$.
 7. Add to T and V a column of zeroes at the right-hand side, corresponding to the new element.
 8. If choice is 'intersection', i.e., (a), handle the last column of matrix T with Algorithm 10 below.
 9. If there are no pending operations, then STOP.
 10. Go to step 3.
-

Algoritmo 10 Updating matrix T .

| **Inputs:** α, β, T . | **Outputs:** modified T .

1. For each $r = 1, \dots, \alpha$: if $T(r, \alpha) = \text{'dom'}$, set $T(r, h) = \text{'dom'}$.
 2. For each $r = 1, \dots, \beta$: if $T(r, \beta) = \text{'dom'}$, set $T(r, h) = \text{'dom'}$.
 3. For each r , $1 \leq r \leq h$: if $T(r, \alpha)$ or $T(r, \beta)$ are 'same' or 'done' and $T(r, h + 1) \neq \text{'dom'}$, then set $T(r, h) = \text{'same'}$.
 4. Set $T(\alpha, h) = \text{'dom'}$ and $T(\beta, h) = \text{'dom'}$.
-

Proof: Consider $O(h) = q = I(O(\alpha), O(\beta))$ from step 6 of Alg. 9. The first statement comes from the fact that $O(\alpha) \succ q, O(\beta) \succ q$ is encoded in the Step 4 (Alg. 10), and transitivity of the I -operator justifies carrying out steps 1 and 2 (Alg. 10), as the new element inherits subset-dominated sets from its parents.

Regarding the second statement, note that Step 4 (Alg. 9) will set $T(\alpha, \beta) = \text{'done'}$.

Consider now for arbitrary $r < h$, the operation

$$I(O(r), O(h)) = I(O(r), O(\alpha), O(\beta))$$

Step 3 (Alg. 10) labels with 'same' the element $T(r, h)$ if $T(r, \alpha)$ or $T(r, \beta)$ are 'same' or 'done' because:

- If $T(r, \alpha) = \text{'same'}$, it means that $I(O(r), O(\alpha))$ can be obtained as the intersection of pre-existing elements²⁸. Then operations between such pre-existing elements and β , which was also on the list, would obtain the 'same' result: there is no need of evaluating:

$$I(O(r), O(h)) = I(I(O(r), O(\alpha)), O(\beta))$$

, so Algorithm 10 set $T(r, h) = \text{'same'}$.

- If $T(r, \alpha) = \text{'done'}$, there exists s such that $I(O(r), O(\alpha)) = O(s)$; hence, $I(O(r), O(h)) = I(O(r), O(\alpha), O(\beta)) = I(O(s), O(\beta))$ so the intersection (r, h) can be skipped, obtaining the same results with s and β ; so, Alg. 10 sets $T(r, h) = \text{'same'}$.

□

As a consequence of the above, I -operations associated to elements of T with a label different to 'pending' can be skipped. The algorithm ends and produces finitely-generated sets when no pending operations remain.

Probabilistic ranking.

Another improvement regarding step 3 of Algorithm 9, can be conceived: in case of limited computing resources it would be desirable to prioritise the evaluation of high-likelihood options. Note that computing the value function of all pending operations (Definitions 8.5 and 8.6) before actually computing the 'set' is very fast. With that data, pending operations can be sorted according to, for instance, the stationary probability. This idea is reminiscent

²⁸We are implicitly using an induction argumentation: we are proving that if the proposition holds for prior list elements, so it will for subsequent additions.

of the probability-based scenario generation in, for instance, (Bernardini and Bemporad 2012). In this way, in non-converged cases the operations with better expected value function would be carried out first.

8.6. Examples

Example 1.

Consider the unstable first-order system $x_{k+1} = 1,2x_k + u_k$ in mode 1, and (actuator failure) $x_{k+1} = 1,2x_k$ in mode 2, consider $\mathbb{U} = \{|u| \leq 1\}$, $\Omega^{[1]} = \Omega^{[2]} = \{|x| \leq 20\}$. It will be assumed that there is a 10 % chance of being in mode 2 at any time. The terminal set ($RL = 1$) is given by $\mathcal{K}_\infty^{[1]} = [-0,833, +0,833]$, $\mathcal{K}_\infty^{[2]} = \{0\}$. As there is a non-stabilisable mode, the algorithms will not converge.

In order to illustrate the steps of the algorithms, to compute reliability bounds, we would start by:

- $o_1 = \langle 0,8333, \{o_1^{val}(1) = 1, o_1^{suc}(1) = 0, o_1^{val}(2) = 0, o_1^{suc}(2) = \infty\} \rangle$,
- $o_2 = \langle 0, \{o_2^{val}(1) = 0, o_2^{suc}(1) = \infty, o_2^{val}(2) = 1, o_2^{suc}(2) = 0\} \rangle$.

where, abusing the notation, a single number γ in place of a set should be understood as $\{|x| \leq \gamma\}$. Then, we can compute:

- $o_3 = \mathcal{Q}_1(o_1) = \langle 1,527, \{o_3^{val}(1) = 0,9, o_3^{suc}(1) = 0,833, o_3^{val}(2) = 0, o_3^{suc}(2) = \infty\} \rangle$,
- $o_4 = \mathcal{Q}_2(o_1) = \langle 0,694, \{o_4^{val}(1) = 0, o_4^{suc}(1) = \infty, o_4^{val}(2) = 0,9, o_4^{suc}(2) = 0,833\} \rangle$.

As o_4 has an smaller associated set than o_1 , from the properties of $I(o_1, o_4)$, we can now update o_4 to: $o_4 = \langle 0,694, \{o_4^{val}(1) = 1, o_4^{suc}(1) = 0, o_4^{val}(2) = 0,9, o_4^{suc}(2) = 0,833\} \rangle$. Now, we can set $o_5 = \mathcal{Q}_2(o_4) = \langle 0,694/1,2, \{o_5^{val}(1) = 0, o_5^{suc}(1) = \infty, o_5^{val}(2) = 0,9 \cdot 1 + 0,1 \cdot 0,9, o_5^{suc}(2) = 0,694\} \rangle$, and so on: suitable steps in the algorithms would similarly proceed until desired.

Figure 8-1 depicts the obtained reliability and MTTF bounds (in both cases, algorithms have been stopped when the final O had 1000 or more elements). The ordinate axis depicts:

- log-likelihood $y = -\log_{10}(1 - RL(x, \theta))$ (i.e., the inverse of the probability of failure), and

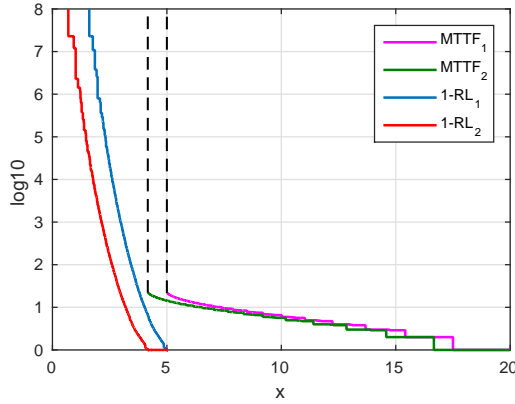


Figura 8-1: Example 1: $1 - RL(x, \theta)$ vs. MTTF for each mode, without disturbances (\log_{10}).

- estimated MTTF bound (logarithmic scale, too; magenta for mode 1, green for mode 2).

For instance, when starting in mode 1 for any $|x| > 5$, constraint violation is unavoidable due to control saturation ($1,2|x| - 1 > |x|$), so reliability is zero. Also in $|x| = 5$ we have $RL = 0$ because in mode 1, successor state is 5 but sooner or later mode 2 will occur, so it will depart to the unrecoverable region. This is in accordance with the results from the algorithm, which asserts, too, that $MTTF_1 \geq 21,92$ for $|x| = 5$.

Also, if starting in mode 2, if $|x| \geq 5/1,2 = 4,1667$ then the system will be steered out of $|x| < 5$ so, again, it will be unrecoverable (reliability zero). For $|x| = 5/1,2$, the algorithm proves that $MTTF_2 \geq 21,62$). If starting in mode 2, if $|x| < 4,166$ the system is steered to $|x| < 5$ so, as mode 1 is quite likely, in a ‘lucky’ trial, the controller will drive it to the origin: reliability is non-zero, MTTF is infinite. For instance, for $x = 2,60$ the likelihood of ending up in the origin is at least 0,99 if starting in mode 2, and 0,9999 if starting in mode 1; for $x = 1,47$, $RL(1,47, 2) \geq 1 - 10^{-5}$ and $RL(1,47, 1) \geq 1 - 10^{-8}$.

Example 2.

Consider $x_{k+1} = 1,2x_k + u_k + w_k$ in mode 1, and $x_{k+1} = 1,2x_k + w_k$ in mode 2, with $\Omega^{[1]} = \Omega^{[2]} = \{|x| \leq 5\}$, $\mathbb{W} = \{|w| < 0,1\}$ and same \mathbb{U} , and mode probabilities as above. In this case, RL is zero everywhere (possibly unstable disturbed system). MTTF bound changes sharply at $v^1 = \{|x| < 4,5\}$ for mode 1 and $v^2 = \{|x| < 3,67\}$ for mode 2 (note that v^1 is the maximal robust invariant set for ‘only mode 1’ and 3.67 is that of $Q_2(v^1)$). Outside such

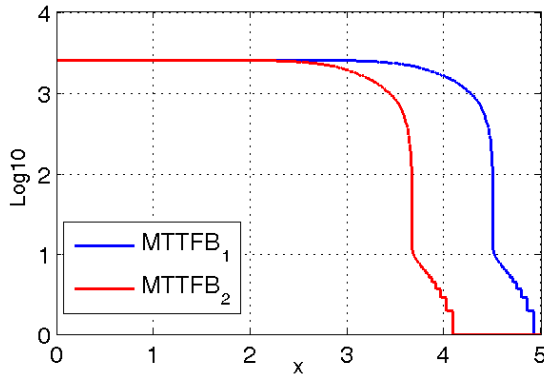


Figura 8-2: Example 2: Mean-time to failure bound (\log_{10}).

sets, failure is very quick to come (estimated $\text{MTTF} < 10$). For $|x| < 2$ the MTTF is larger than 2510 samples. The estimated log-MTTF bound appears in Figure 8-2; O had 950 elements.

Example 3.

Consider the undisturbed second-order MJLS (8.1) with three operating modes, $i = \{1, 2, 3\}$ considered in Section 7.5. The transition probabilities, model matrices A_i , B_i and constraint regions are given in Table 7.1.

Terminal sets $\mathcal{K}_\infty^{[i]}$ were computed using the mode-dependent one-step controllable set (8.2) and the Algorithm 9 was executed. The total number of sets when the algorithm converged was 34. Computation time (i7-4790K, Matlab® R2015a) was 19.94 seconds.

Figure 8-3 depicts the ‘stationary’ reliability bound, considering that the initial mode probability distribution is the stationary one, i.e.,

$$RL_s(x) := \sum_{i=1}^3 p_i RL(x, i),$$

where, from the eigenvectors of \mathbb{P} , the prior probability of each mode has been set to $p_1 = 0,7353$, $p_2 = 0,0294$ and $p_3 = 0,2353$.

As it is a disturbance-free case with finite-step algorithm convergence, results are optimal (Theorem 8.2). Obviously, the results improve over those in Section 7.5 Figure 7-2, for instance, given an initial state $x_0 = [3,748, 6,961]$, Figure 7-2 provides a guaranteed reliability $RL(x_0, \theta_0)$ of $\{0,9604, 0, 1\}$ when

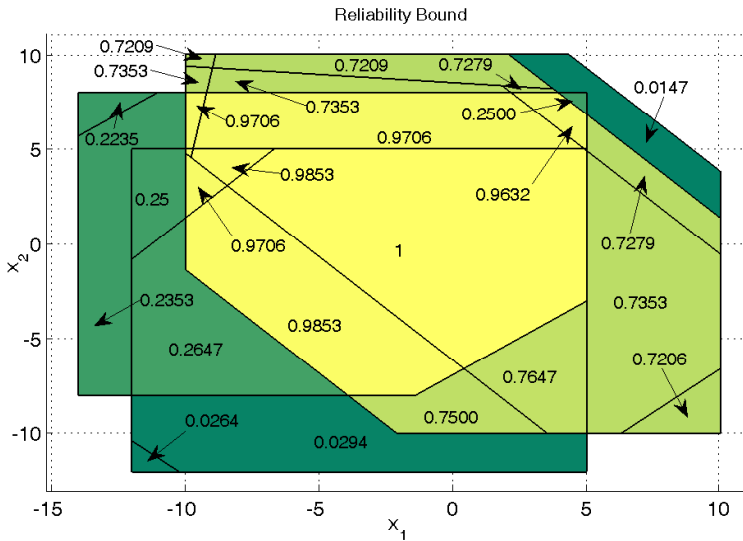


Figura 8-3: Example 3: Optimal reliability (stationary mode assumption).

$\theta_0 = \{1, 2, 3\}$, respectively; the here presented modifications provide an optimal reliability of $\{0.99, 0, 1\}$.

8.7. Conclusion

This work has extended prior results in Chapter 7 dealing with reliability bounds and associated mode-dependent control laws for Markov-Jump linear systems. Reliability is understood as the probability of avoiding constraint violations in future time; obviously, inside the maximal robust controllable set reliability is 1, and the concept is closely related to the likelihood of reaching the terminal set.

The extensions account for disturbances and non-stabilisable modes. Iterative algorithms are presented, improving the reliability bound achieved in the referred prior results. Optimality is achieved in the disturbance-free case if algorithms converge in a finite number of iterations. Later, a further extension computes bounds on robust mean time to failure (time to constraint violation under worst-case disturbance) in zero-reliability regions. Algorithms are based on polyhedral 1-step sets and intersections of such sets.

Conclusiones

Conclusiones y Trabajo Futuro

En la presente tesis doctoral se ha propuesto el diseño de controladores predictivos basado en escenarios para sistemas lineales con saltos markovianos sujetos a restricciones duras, considerando dos escenarios diferentes: el primero de ellos no arriesgados y el otro arriesgados. Se entiende como escenario no arriesgados aquellos en donde las trayectorias del sistema con estados iniciales pertenecientes a ciertos conjuntos invariantes, convergen –en media– al origen o a una vecindad invariante de este con un 100 % de probabilidad.

Las contribuciones de la tesis abarcan escenarios no arriesgados y, posteriormente, algunos casos de escenarios arriesgados.

En el primero de los casos, se ha formulado un MPC para secuencias de modos de funcionamientos con diferentes horizontes, tanto basado en un modelo de modo conocido en el momento de control como en un modelo más general basado en cierta información sobre probabilidades de modo. Se han propuesto condiciones e ingredientes terminales y propiedades de las secuencias o árboles de escenarios para asegurar la factibilidad recursiva y estabilidad en media cuadrática del sistema en lazo cerrado, sin la necesidad de imponer funciones de Lyapunov adicionales como restricciones, que es cómo solventaban el problema en literatura previa.

En caso de escenarios arriesgados, en el capítulo 6 se presenta un primer enfoque en el diseño de MPC para MJLS sujeto a restricciones. En este capítulo se trata con sistemas que pueden tener uno o varios modos inestables en el sistema, y el enfoque se basa en evaluar la confiabilidad de un escenario por simulación exhaustiva. Se propuso un algoritmo que, por un lado, determina la probabilidad de violar restricciones en instante de tiempo futuro y por otro, la probabilidad de alcanzar un cierto conjunto terminal. Luego, El pro-

blema de optimización predictivo fue formulado tomando en cuenta solo las realizaciones más probables de la variable θ_k .

En el siguiente capítulo, partiendo del máximo conjunto contractivo en donde existe una probabilidad del 100 % de llevar el estado del sistema al origen, se obtuvo una secuencia de conjuntos anidados y acciones de control asociados a ellos. Dichos conjuntos que son dependiente de una secuencia en particular, pueden garantizar una cota de confiabilidad mínima, entendida como la posibilidad de conducir el estado del sistema al origen sin violar restricciones. Entonces, para estados iniciales pertenecientes a estos conjuntos existe al menos una secuencia de longitud finita con un controlador asociado, tal que, el sistema puede ser llevado al origen con una probabilidad menor al 100 %.

Por último, en el capítulo 8 se presentó un algoritmo iterativo que mejora la cota de confiabilidad obtenida en el capítulo anterior. Si el algoritmo convergen en un número finito de paso, se obtiene una ley de control óptima para el caso de sistemas libres de perturbaciones. Para sistemas inestables en lazo cerrado, se propuso una modificación del algoritmo con el fin de calcular el tiempo medio para fallar. Ambos algoritmos están basados en operaciones poliédricas tales como, conjunto a un paso e intersecciones entre estos.

Los resultados de la tesis permiten, en los casos de cambio abrupto de modo estudiados, calcular la probabilidad de éxito de una ley de control en función de las condiciones iniciales, y diseñar leyes de control que intenten garantizar una cota mínima de esta confiabilidad o, al menos, de tiempo medio para fallar.

Obviamente, el mayor inconveniente a horizontes largos es la carga computacional necesaria dada la explosión del número de combinaciones. Por eso, la generación de escenarios más probables de horizonte largo, dejando las opciones menos probables a horizonte corto (o fuera de consideración) es crucial en la aplicabilidad del control predictivo para sistemas con saltos markovianos y esta tesis ha contribuido a precisar y desarrollar cuestiones en esta materia.

Trabajo Futuro

Una línea de interés sería intentar diseñar controladores en un contexto multicriterio, generando un frente de Pareto entre la optimalidad de prestaciones (índice de coste cuadrático) y confiabilidad del controlador para determinado conjunto de condiciones iniciales. Obviamente, fuera del conjunto máximo invariante bajo control es obligado hacerlo; pero, incluso dentro de

él, el descartar casos de muy baja probabilidad podría conducir a obtener una gran mejora en el índice de coste. El problema genérico sería considerar una distribución de probabilidad de coste. El gran problema es que el número de escenarios posibles es muy grande, y alguna simplificación debería ser abordada.

Por último, se tratará de aplicar los resultados a distintos sistemas reales. La aplicaciones de estas ideas a configuraciones de control en red, fallos esporádicos, etc. parecen ser idóneas; en estos momentos se está estudiando posibles contextos de aplicación en la rama de ingeniería eléctrica.

Bibliografía

- Aberkane, Samir (2011). Stochastic stabilization of a class of nonhomogeneous markovian jump linear systems. *Systems & Control Letters* **60**(3), 156–160.
- Aberkane, Samir, Jean Ponsart, Mickael Rodrigues and Dominique Sauter (2008). Output feedback control of a class of stochastic hybrid systems. *Automatica* **44**(5), 1325–1332.
- Alamo, Teodoro, José Bravo and Eduardo Camacho (2005). Guaranteed state estimation by zonotopes. *Automatica* **41**(6), 1035–1043.
- Alessio, Alessandro and Alberto Bemporad (2009). A survey on explicit model predictive control. In: *Nonlinear model predictive control*. pp. 345–369. Springer.
- Ariño, Carlos, Emilio Pérez, Antonio Sala and Fernando Bedate (2014a). Polytopic invariant and contractive sets for closed-loop discrete fuzzy systems. *Journal of the Franklin Institute* **351**(7), 3559–3576.
- Arino, Carlos, Ernesto Perez, Andres Querol and Antonio Sala (2014b). Model predictive control for discrete fuzzy systems via iterative quadratic programming. In: *Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on*. IEEE. pp. 2288–2293.
- Ariño, Carlos, Ernesto Perez, Fernando Bedate and Antonio Sala (2013). Robust polytopic invariant sets for discrete fuzzy control systems. In: *Fuzzy Systems, 2013 IEEE International Conference on*. IEEE. pp. 1–7.
- Åström, Karl and Björn Wittenmark (2013). *Computer-controlled systems: theory and design*. Courier Corporation.
- Bank, Bernd, Jürgen Guddat, Diethard Klatte, Bernd Kummer and Klaus Tammer (1982). Non-linear parametric optimization. *Akademie-Verlag, Berlin*.

- Baotic, MATO (2005). Optimal Control of Piecewise Affine Systems—a Multi-parametric Approach—. PhD thesis. University of Zagreb, Croatia.
- Bemporad, Alberto and Manfred Morari (1999a). Control of systems integrating logic, dynamics, and constraints. *Automatica* **35**(3), 407–427.
- Bemporad, Alberto and Manfred Morari (1999b). Robust model predictive control: A survey. In: *Robustness in identification and control*. pp. 207–226. Springer.
- Bemporad, Alberto, Francesco Borrelli and Manfred Morari (2003). Min-max control of constrained uncertain discrete-time linear systems. *Automatic Control, IEEE Transactions on* **48**(9), 1600–1606.
- Bemporad, Alberto, Francesco Borrelli, Manfred Morari et al. (2002a). Model predictive control based on linear programming~ the explicit solution. *IEEE Transactions on Automatic Control* **47**(12), 1974–1985.
- Bemporad, Alberto, Manfred Morari, Vivek Dua and Efstratios N Pistikopoulos (2002b). The explicit linear quadratic regulator for constrained systems. *Automatica* **38**(1), 3–20.
- Berenguel, Manuel, F. Rodríguez, F. Acién and J. L. Garcia (2004). Model predictive control of ph in tubular photobioreactors. *Journal of Process Control* **14**(4), 377–387.
- Bernardini, Daniele and Alberto Bemporad (2009). Scenario-based model predictive control of stochastic constrained linear systems. In: *Decision and Control, 2009. Proceedings of the 48th IEEE Conference on*. IEEE. pp. 6333–6338.
- Bernardini, Daniele and Alberto Bemporad (2012). Stabilizing model predictive control of stochastic constrained linear systems. *Automatic Control, IEEE Transactions on* **57**(6), 1468–1480.
- Bertsekas, Dimitri (1999). Nonlinear programming.
- Besselmann, Thomas (2010). *Constrained optimal control*. Diss., Eidgenössische Technische Hochschule ETH Zürich, Nr. 18950, 2010.
- Blair Jr, W. and D. Sworder (1975). Feedback control of a class of linear discrete systems with jump parameters and quadratic cost criteria. *International Journal of Control* **21**(5), 833–841.

- Blanchini, F (1994). Ultimate boundedness control for uncertain discrete-time systems via set-induced lyapunov functions. *Automatic Control, IEEE Transactions on* **39**(2), 428–433.
- Blanchini, Franco (1999). Set invariance in control. *Automatica* **35**(11), 1747–1767.
- Blanchini, Franco and Stefano Miani (2000). Any domain of attraction for a linear constrained system is a tracking domain of attraction. *SIAM Journal on Control and Optimization* **38**(3), 971–994.
- Blanke, Mogens and Jochen Schröder (2003). *Diagnosis and fault-tolerant control*. Springer.
- Blanke, Mogens, Michel Kinnaert, Jan Lunze, Marcel Staroswiecki and J Schröder (2010). *Diagnosis and fault-tolerant control*. Springer Publishing Company, Incorporated.
- Blasco, Xavier, Miguel Martínez, Juan M. Herrero, Cesar Ramos and Javier Sanchis (2007). Model-based predictive control of greenhouse climate for reducing energy and water consumption. *Computers and Electronics in Agriculture* **55**(1), 49–70.
- Borrelli, F, A Bemporad and M Morari (2014). Predictive control for linear and hybrid systems. *preparation, available online at <http://www.mpc.berkeley.edu/mpc-course-material>*.
- Borrelli, Francesco (2003). *Constrained optimal control of linear and hybrid systems*. Vol. 290. Springer.
- Boukas, El Kebir and ZK Liu (2001). Manufacturing systems with random breakdowns and deteriorating items. *Automatica* **37**(3), 401–408.
- Boyd, Stephen and Lieven Vandenberghe (2004). *Convex optimization*. Cambridge university press.
- Boyd, Stephen P, Laurent El Ghaoui, Eric Feron and Venkataramanan Balakrishnan (1994). *Linear matrix inequalities in system and control theory*. Vol. 15. SIAM.
- Bravo, José, Daniel Limón, Teodoro Alamo and Eduardo Camacho (2005). On the computation of invariant sets for constrained nonlinear systems: An interval arithmetic approach. *Automatica* **41**(9), 1583–1589.

- Bumroongsri, Pornchai and Soorathep Kheawhom (2012). An off-line robust MPC algorithm for uncertain polytopic discrete-time systems using polyhedral invariant sets. *Journal of Process Control* **22**(6), 975–983.
- Calafiore, Giuseppe, Marco Campi et al. (2006). The scenario approach to robust control design. *IEEE Transactions on Automatic Control* **51**(5), 742–753.
- Calle-Prado, Alejandro, Salvador Alepuz, Josep Bordonau, Joan Nicolas-Apruzzese, Patricio Cortés and Jose Rodriguez (2015). Model predictive current control of grid-connected neutral-point-clamped converters to meet low-voltage ride-through requirements. *IEEE Transactions on Industrial Electronics* **62**(3), 1503–1514.
- Camacho, Eduardo and Carlos Bordons (2013). *Model predictive control*. Springer Science & Business Media.
- Camacho, Eduardo F and Carlos Bordons (2010). Control predictivo: Pasado, presente y futuro. *RIAI* **1**(3), 5–28.
- Camacho, Eduardo F, Daniel R. Ramírez, Daniel Limón, D Muñoz and Teodoro Alamo (2010). Model predictive control techniques for hybrid systems. *Annual reviews in control* **34**(1), 21–31.
- Christophersen, Frank J (2006). Optimal control and analysis for constrained piecewise affine systems. PhD thesis. Diss., Eidgenössische Technische Hochschule ETH Zürich, Nr. 16807, 2006.
- Costa, Osvaldo, E. Assumpção, E. Boukas and R. Marques (1999). Constrained quadratic state feedback control of discrete-time markovian jump linear systems. *Automatica* **35**(4), 617–626.
- Costa, Osvaldo, M. Fragoso and R. Marques (2005). *Discrete time Markov jump linear systems*. Springer.
- Cutler, Charles R and Brian L Ramaker (1980). Dynamic matrix control. A computer control algorithm. In: *joint automatic control conference*. number 17. p. 72.
- Dantzig, George Bernard (1998). *Linear programming and extensions*. Princeton university press.
- de Souza, Carlos E (2006). Robust stability and stabilization of uncertain discrete-time markovian jump linear systems. *Automatic Control, IEEE Transactions on* **51**(5), 836–841.

- Dehghan, Masood and Chong-Jin Ong (2012). Discrete-time switching linear system with constraints: Characterization and computation of invariant sets under dwell-time consideration. *Automatica* **48**(5), 964–969.
- Diehl, Moritz, Rishi Amrit and James Rawlings (2011). A Lyapunov function for economic optimizing model predictive control. *Automatic Control, IEEE Transactions on* **56**(3), 703–707.
- Ding, Baocang (2011). Dynamic output feedback predictive control for nonlinear systems represented by a Takagi–Sugeno model. *Fuzzy Systems, IEEE Transactions on* **19**(5), 831–843.
- Dórea, C. and J. Hennes (1999). (a, b)-invariant polyhedral sets of linear discrete-time systems. *Journal of optimization theory and applications* **103**(3), 521–542.
- Fang, Yuguang and K. Loparo (2002). Stochastic stability of jump linear systems. *IEEE Transactions on Automatic Control* **47**(7), 1204–1208.
- Farina, Marcello, Luca Giulioni and Riccardo Scattolini (2016). Stochastic linear model predictive control with chance constraints—a review. *Journal of Process Control* **44**, 53–67.
- Feng, Xiangbo, Kenneth A. Loparo, Yuandong Ji and Howard Jay Chizeck (1992). Stochastic stability properties of jump linear systems. *Automatic Control, IEEE Transactions on* **37**(1), 38–53.
- Frank, Paul M (2002). Handling modelling uncertainty in fault detection and isolation systems. *Journal of Control Engineering and Applied Informatics* **4**(4), 29–46.
- Gilbert, Elmer and Kok Tan (1991). Linear systems with state and control constraints: The theory and application of maximal output admissible sets. *Automatic Control, IEEE Transactions on* **36**(9), 1008–1020.
- Goodwin, Graham C, María M Seron and José A De Doná (2006). *Constrained control and estimation: an optimisation approach*. Springer Science & Business Media.
- Grancharova, Alexandra and Tor Arne Johansen (2012). Multi-parametric programming. In: *Explicit Nonlinear Model Predictive Control*. pp. 1–37. Springer.

- Grant, Michael and Stephen Boyd (2008). Graph implementations for nonsmooth convex programs. In: *Recent Advances in Learning and Control* (V. Blondel, S. Boyd and H. Kimura, Eds.). pp. 95–110. Lecture Notes in Control and Information Sciences. Springer-Verlag Limited. http://stanford.edu/~boyd/graph_dcp.html.
- Grieder, P, S. Rakovic, Manfred Morari and David Mayne (2005). Invariant sets for switched discrete time systems subject to bounded disturbances. In: *IFAC World Congress*.
- Grüne, Lars and Jürgen Pannek (2011). *Nonlinear model predictive control*. Springer.
- Hartley, Edward N and Jan M Maciejowski (2014). Reconfigurable predictive control for redundantly actuated systems with parameterised input constraints. *Systems & Control Letters* **66**, 8–15.
- Helwa, Mohamed K, Zhiyun Lin and Mireille E Broucke (2016). On the necessity of the invariance conditions for reach control on polytopes. *Systems & Control Letters* **90**, 16–19.
- Herceg, Martin, Michal Kvasnica, Colin N Jones and Manfred Morari (2013). Multi-parametric Toolbox 3.0. In: *Control Conference (ECC), 2013 European*. IEEE. pp. 502–510.
- Horst, Reiner and Panos Pardalos (2013). *Handbook of global optimization*. Vol. 2. Springer Science & Business Media.
- Hu, Tingshu, Zongli Lin and Ben Chen (2002). An analysis and design method for linear systems subject to actuator saturation and disturbance. *Automatica* **38**(2), 351–359.
- Isermann, Rolf (2011). *Fault-diagnosis applications: model-based condition monitoring: actuators, drives, machinery, plants, sensors, and fault-tolerant systems*. Springer.
- Kalman, Rudolf et al. (1960). Contributions to the theory of optimal control. *Bol. Soc. Mat. Mexicana* **5**(2), 102–119.
- Karmarkar, Narendra (1984). A new polynomial-time algorithm for linear programming. In: *Proceedings of the sixteenth annual ACM symposium on Theory of computing*. ACM. pp. 302–311.
- Kerrigan, Eric (2000). Robust constraint satisfaction: Invariant sets and predictive control. PhD thesis. PhD thesis, Cambridge.

- Kerrigan, Eric and Jan Maciejowski (2000). Invariant sets for constrained nonlinear discrete-time systems with application to feasibility in model predictive control. In: *Decision and Control, 2000. Proceedings of the 39th IEEE Conference on*. Vol. 5. IEEE. pp. 4951–4956.
- Kothare, Mayuresh, Venkataramanan Balakrishnan and Manfred Morari (1996). Robust constrained model predictive control using linear matrix inequalities. *Automatica* **32**(10), 1361–1379.
- Kouro, Samir, Marcelo A. Perez, Jose Rodriguez, Ana M. Llor and Hector A. Young (2015). Model predictive control: Mpc’s role in the evolution of power electronics. *IEEE Industrial Electronics Magazine* **9**(4), 8–21.
- Kouvaritakis, Basil and Mark Cannon (2015). *Model Predictive Control: Classical, Robust and Stochastic*. Springer.
- Krtolica, R., Ü Özgüner, H. Chan, H. Göktas, J. Winkelman and M. Liubakka (1994). Stability of linear feedback systems with random communication delays. *International Journal of Control* **59**(4), 925–953.
- Kvasnica, Michal, Johan Löfberg and Miroslav Fikar (2011). Stabilizing polynomial approximation of explicit mpc. *Automatica* **47**(10), 2292–2297.
- Kvasnica, Michal, Pascal Grieder, Mato Baotić and Manfred Morari (2004). Multi-parametric toolbox (MPT). In: *Hybrid systems: computation and control*. pp. 448–462. Springer.
- Lazar, Mircea (2006). Model predictive control of hybrid systems: Stability and robustness.
- Lazar, Mircea, W. Heemels, Siep Weiland and Alberto Bemporad (2006). Stabilizing model predictive control of hybrid systems. *Automatic Control, IEEE Transactions on* **51**(11), 1813–1818.
- Lee, J. and Zhenghong Yu (1997). Worst-case formulations of model predictive control for systems with bounded parameters. *Automatica* **33**(5), 763–781.
- Löfberg, J. (2004). Yalmip : A toolbox for modeling and optimization in MATLAB. In: *Proceedings of the CACSD Conference*. Taipei, Taiwan.
- Li, Hongyi, Huijun Gao, Peng Shi and Xudong Zhao (2014). Fault-tolerant control of markovian jump stochastic systems via the augmented sliding mode observer approach. *Automatica* **50**(7), 1825–1834.

- Ling, Qiang and Haojiang Deng (2012). A new proof to the necessity of a second moment stability condition of discrete-time markov jump linear systems with real states. *Journal of Applied Mathematics*.
- Liu, Wei, Huaguang Zhang, Zhanshan Wang and Qiuye Sun (2012). State estimation for discrete-time Markov jump linear systems based on orthogonal projective theorem. *International Journal of Control, Automation and Systems* **10**(5), 1049–1054.
- Löfberg, Johan (2003). *Minimax approaches to robust model predictive control*. Vol. 812. Linköping University Electronic Press.
- Lu, Jianbo, Dewei Li and Yugeng Xi (2013). Constrained model predictive control synthesis for uncertain discrete-time markovian jump linear systems. *Control Theory & Applications, IET*.
- Lu, Yaohui and Yaman Arkun (2000). Quasi-min-max MPC algorithms for LPV systems. *Automatica* **36**(4), 527–540.
- Ma, Shuping and El-Kébir Boukas (2011). Guaranteed cost control of uncertain discrete-time singular Markov jump systems with indefinite quadratic cost. *International journal of robust and nonlinear control* **21**(9), 1031–1045.
- Mayne, D. Q., J. B. Rawlings, C. V. Rao and P. O. M. Scokaert (2000). Constrained model predictive control: Stability and Optimality. *Automatica* **36**(6), 789–814.
- Mayne, David Q (2014). Model predictive control: Recent developments and future promise. *Automatica* **50**(12), 2967–2986.
- Mayne, David Q and S Raković (2003). Model predictive control of constrained piecewise affine discrete-time systems. *International Journal of Robust and Nonlinear Control* **13**(3-4), 261–279.
- Müller, Matthias and Frank Allgöwer (2012). Improving performance in model predictive control: switching cost functionals under average dwell-time. *Automatica* **48**(2), 402–409.
- Muñoz, David, Alberto Bemporad and Carlo Filippi (2004). Robust explicit mpc based on approximate multi-parametric convex programming. In: *Decision and Control, 2004. CDC. 43rd IEEE Conference on*. Vol. 3. IEEE. pp. 2491–2496.
- Murty, Katta G (1983). Linear programming.

- Nesterov, Yurii, Arkadii Nemirovskii and Yinyu Ye (1994). *Interior-point polynomial algorithms in convex programming*. Vol. 13. SIAM.
- Neumaier, Arnold (2004). Complete search in continuous global optimization and constraint satisfaction. *Acta numerica* **13**, 271–369.
- Park, B. and W. Kwon (2002). Robust one-step receding horizon control of discrete-time markovian jump uncertain systems. *Automatica* **38**(7), 1229–1235.
- Patrinos, Panagiotis, Pantelis Sopasakis, Haralambos Sarimveis and Alberto Bemporad (2014). Stochastic model predictive control for constrained discrete-time markovian switching systems. *Automatica* **50**(10), 2504–2514.
- Pérez, Emilio, Carlos Ariño, Xavier Blasco and Miguel A. Martínez (2011). Solución explícita al control predictivo de sistemas lineales sujetos a restricciones no convexas. *Revista Iberoamericana de Automática e Informática Industrial RIAI* **8**(3), 167–181.
- Pitarch, José L., Antonio Sala and Carlos Ariño (2015). Estabilidad de sistemas takagi-sugeno bajo perturbaciones persistentes: estimación de conjuntos inescapables. *Revista Iberoamericana de Automática e Informática Industrial RIAI* **12**(4), 457–466.
- Propoi, A. (1963). Use of linear programming methods for synthesizing sampled-data automatic systems. *Automation and remote control* **24**(7), 837–844.
- Qin, Joe and Thomas Badgwell (2003). A survey of industrial model predictive control technology. *Control engineering practice* **11**(7), 733–764.
- Raković, Sara, P. Grieder, Michal Kvasnica, David Mayne and Manfred Morari (2004). Computation of invariant sets for piecewise affine discrete time systems subject to bounded disturbances. In: *Decision and Control, 2004. CDC. 43rd IEEE Conference on*. Vol. 2. IEEE. pp. 1418–1423.
- Ramirez, D. R., Jorn K. Gruber, Teodoro Alamo, Carlos Bordons and Eduardo F. Camacho (2008). Control predictivo min-máx de una planta piloto. *Revista Iberoamericana de Automática e Informática Industrial RIAI* **5**(3), 37–47.
- Rawlings, James and David Mayne (2009). *Model predictive control: Theory and design*. Nob Hill Pub.

- Rawlings, James and Rishi Amrit (2009). Optimizing process economic performance using model predictive control. In: *Nonlinear model predictive control*. pp. 119–138. Springer.
- Rawlings, James, David Angeli and Cuyler N Bates (2012). Fundamentals of economic model predictive control. In: *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*. IEEE. pp. 3851–3861.
- Richalet, J, A Rault, JL Testud and J Papon (1976). Algorithmic control of industrial processes. In: *Proceedings of the 4th IFAC symposium on identification and system parameter estimation*. pp. 1119–1167.
- Richalet, Jacques, A. Rault, JL Testud and J. Papon (1978). Model predictive heuristic control: Applications to industrial processes. *Automatica* **14**(5), 413–428.
- Rossiter, J Anthony, Basil Kouvaritakis and MJ Rice (1998). A numerically robust state-space approach to stable-predictive control strategies. *Automatica* **34**(1), 65–73.
- Rubagotti, Matteo, Sergio Trimboli and Alberto Bemporad (2013). Stability and invariance analysis of uncertain discrete-time piecewise affine systems. *Automatic Control, IEEE Transactions on* **58**(9), 2359–2365.
- Sala, Antonio (2009). On the conservativeness of fuzzy and fuzzy-polynomial control of nonlinear systems. *Annual Reviews in Control* **33**(1), 48–58.
- Sala, Antonio and José L. Pitarch (2016). Optimisation of transient and ultimate inescapable set with polynomial boundaries for nonlinear systems. *Automatica* (<http://dx.doi.org/10.1016/j.automatica.2016.06.017>), In press.
- Schildbach, Georg, Lorenzo Fagiano, Christoph Frei and Manfred Morari (2014). The scenario approach for stochastic model predictive control with bounds on closed-loop constraint violations. *Automatica* **50**(12), 3009–3018.
- Schildbach, Georg, Paul Goulart and Manfred Morari (2015). Linear controller design for chance constrained systems. *Automatica* **51**, 278–284.
- Schwarm, Alexander T and Michael Nikolaou (1999). Chance-constrained model predictive control. *AIChE Journal* **45**(8), 1743–1752.
- Scokaert, P. and David Mayne (1998). Min-max feedback model predictive control for constrained linear systems. *Automatic Control, IEEE Transactions on* **43**(8), 1136–1142.

- Skogestad, Sigurd and Ian Postlethwaite (2007). *Multivariable feedback control: analysis and design*. Vol. 2. Wiley New York.
- Sturm, J.F. (1999). Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software* **11–12**, 625–653. Version 1.05 available from <http://fewcal.kub.nl/sturm>.
- Tanaka, Kazuo and Hua O Wang (2004). *Fuzzy control systems design and analysis: a linear matrix inequality approach*. John Wiley & Sons.
- Tijms, Henk C. (2003). *A first course in stochastic models*. John Wiley and Sons.
- Todorov, Marcos G and Marcelo D Fragoso (2014). New methods for mode-independent robust control of Markov jump linear systems. In: *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*. IEEE. pp. 4222–4227.
- Toh, Kim-Chuan, Michael J Todd and Reha H Tütüncü (1999). Sdpt3—a matlab software package for semidefinite programming, version 1.3. *Optimization methods and software* **11(1-4)**, 545–581.
- Wan, Zhaoyang and Mayuresh V Kothare (2003). An efficient off-line formulation of robust model predictive control using linear matrix inequalities. *Automatica* **39(5)**, 837–846.
- Wang, Ye, Carlos Ocampo-Martinez and Vicenç Puig (2016). Stochastic model predictive control based on gaussian processes applied to drinking water networks. *IET Control Theory & Applications* **10(8)**, 947–955.
- Wu, Jing and Tongwen Chen (2007). Design of networked control systems with packet dropouts. *Automatic Control, IEEE Transactions on* **52(7)**, 1314–1319.
- Zhang, Hui, Junmin Wang and Yang Shi (2013). Robust hinf sliding-mode control for markovian jump systems subject to intermittent observations and partially known transition probabilities. *Systems & Control Letters* **62(12)**, 1114–1124.
- Zhang, Liqian, Yang Shi, Tongwen Chen and Biao Huang (2005). A new method for stabilization of networked control systems with random delays. *Automatic Control, IEEE Transactions on* **50(8)**, 1177–1181.
- Zhang, Lixian and El-Kébir Boukas (2009). Stability and stabilization of markovian jump linear systems with partly unknown transition probabilities. *Automatica* **45(2)**, 463–468.

- Zhang, Lixian and James Lam (2010). Necessary and sufficient conditions for analysis and synthesis of markov jump linear systems with incomplete transition descriptions. *Automatic Control, IEEE Transactions on* **55**(7), 1695–1701.
- Zhang, Tiejun, Gang Feng and Jianhong Lu (2007). Fuzzy constrained min-max model predictive control based on piecewise lyapunov functions. *Fuzzy Systems, IEEE Transactions on* **15**(4), 686–698.

Apéndice

Apéndice A

Optimización con Restricciones

Programación Lineal (LP)

Para este tipo de problema de optimización en específico, la función de coste y las restricciones son lineales:

$$\begin{aligned} \min_x \quad & c^T x \\ \text{s.t.} \quad & Gx \leq h, \\ & Ax = b, \end{aligned} \tag{A.1}$$

donde $c \in \mathbb{R}^{n_x}$, $G \in \mathbb{R}^{n_G \times n_x}$, $h \in \mathbb{R}^h$, $A \in \mathbb{R}^{n_b \times n_x}$ y $b \in \mathbb{R}^{n_b}$. Aunque no existe una formula analítica para obtener la solución de un LP, existen numeroso optimizadores eficientes que resuelven dicho problema, ejemplo, *método Simplex* (Murty 1983, Dantzig 1998), *método de punto interior* (Karmarkar 1984, Nesterov *et al.* 1994).

Programación Cuadrática (QP)

La programación cuadrática se refiere a un problema de optimización convexa en donde la función objetivo es *cuadrática* mientras que las restricciones

son *lineales*. Puede ser establecido de la siguiente forma:

$$\begin{aligned} \min_x \quad & \frac{1}{2}x^T Qx + c^T x & (\text{A.2}) \\ \text{s.t.} \quad & Gx \leq h, \\ & Ax = b, \end{aligned}$$

donde, la matriz Q debe ser semi-definida positiva para que la función de coste sea convexa, i.e. $Q \succeq 0$. Además, $Q \in \mathbb{R}^{n_x \times n_x}$, $c \in \mathbb{R}^{n_x}$, $G \in \mathbb{R}^{n_G \times n_x}$, $h \in \mathbb{R}^{n_h}$, $A \in \mathbb{R}^{n_b \times n_x}$ y $b \in \mathbb{R}^{n_b}$. Los problemas QP pueden ser resueltos usando *métodos de conjuntos activos*, o *método de punto interior*. Detalles para encontrar soluciones de estos métodos pueden encontrarse por ejemplo en (Boyd and Vandenberghe 2004). El tiempo necesario para la solución de esta clase de optimización es mayor que el necesitado para solucionar un problema LP (Neumaier 2004, página. 37).

Programación Semi-Definida (SDP)

Un problema SDP es un problema de optimización convexa que puede ser expresado de la forma:

$$\min_x \quad c^T x \quad (\text{A.3})$$

$$\text{s.t.} \quad F_0 + \sum_{i=1}^m x_i F_i > 0, \quad (\text{A.4})$$

donde x_1, x_2, \dots, x_m son las variables y las matrices simétricas $F_i = F_i^T \in \mathbb{R}^{n \times n}$ vienen dadas, $c \in \mathbb{R}^{n_x}$. La desigualdad (A.4) usualmente es denotada como *desigualdades matriciales lineales* (LMI) (Boyd et al. 1994). $F(x) > 0$ significa que la función $F(x)$ es definida positiva. Múltiples LMIs $F_1(x), \dots, F_h(x) > 0$, $h > 0$ puede ser expresada como una sola LMI de la forma:

$$\text{diag}(F_1(x), \dots, F_h(x)) > 0$$

Nótese que, la LMI (A.4) es una restricción convexa en la variable x , i.e., $\{x \mid F(x) > 0\}$ es un conjunto convexo. Aunque la LMI (A.4) parece tener una forma especial, se puede representar por una gran variedad de restricciones convexas en x , en particular, desigualdades lineales, cuadráticas (siempre que sea convexas) y restricciones que aparecen en teoría de control como dle tipo

Lyapunov. *etc.*

Un caso en concreto son las restricciones de Lyapunov, donde las variables aparecen en forma matricial, por ejemplo:

$$A^T P + P A < 0, \quad (\text{A.5})$$

donde $A \in \mathbb{R}^{n_x \times n_x}$ es una matriz conocida y $P = P^T$ es la variable de la LMI.

Desigualdades cuadráticas convexas son convertidas a una forma LMI haciendo uso del complemento de Schur.

Complemento de Schur: Dado un conjunto de desigualdades no lineales:

$$R(x) > 0, \quad Q(x) - S(x)R(x)^{-1}S(x)^T > 0, \quad (\text{A.6})$$

donde $Q(x) = Q(x)^T$, $R(x) = R(x)^T$, $S(x)$ depende de x de forma afín. Por tanto, (A.6) puede ser expresado como una LMI de la siguiente forma:

$$\begin{bmatrix} Q(x) & S(x) \\ S(x)^T & R(x) \end{bmatrix} > 0 \quad (\text{A.7})$$

Existen diferentes Software que tratan con problema de optimización convexa como pueden ser YALMIP (Löfberg 2004) o CVX (Grant and Boyd 2008). También existen varios resolvedores de minimización convexa como SeDuMi (Self-Dual-Minimization) (Sturm 1999) o SDPT3 (Semidefinite Programming Tool) (Toh *et al.* 1999).

Ejemplo: Dado un sistema dinámico lineal $\dot{x} = Ax$, con:

$$A = \begin{bmatrix} -1 & 2 & 0 \\ -3 & -4 & 1 \\ 0 & 0 & -2 \end{bmatrix}$$

se desea probar la estabilidad de dicho sistema encontrando una matriz simétrica P que cumpla (A.5), *i.e.*:

$$\begin{aligned} A^T P + P A &\leq 0 \\ P &\geq 0 \end{aligned}$$

Esto puede ser fácilmente implementado en Matlab® usando YALMIP/SeDuMi con el siguiente código:

```

1 A = [-1 2 0;-3 -4 1;0 0 -2]; % Definici\on del sistema
2 P = sdpvar(3,3); % Se define la matriz sim\etrica P
3 F = [P >= eye(3), A'*P+P*A <= 0]; % Se define las restricciones
4 optimize(F); % Se optimiza para encontrar una soluci\on factible
5 Pfactible = value(P); % Soluci\on

```

Programación Multi-paramétrica

Gran parte del análisis y diseño de controladores para sistemas con restricciones descrito en este trabajo usa el concepto básico subyacente de la programación multi-paramétrica ²⁹ (Bank *et al.* 1982, Borrelli 2003) y las referencias citadas en dichos trabajos. Un problema de programación multiparamétrico tiene la siguiente forma:

$$J^*(x) = \min_{\theta} J(\theta, x) \quad (\text{A.8})$$

$$s.t \quad g(\theta, x) \leq 0, \quad (\text{A.9})$$

donde $\theta \in \mathbb{R}^{n_{\theta}}$ representa la variable a ser optimizada, $x \in \mathbb{R}^{n_x}$ es un vector de parámetros, $J : \mathbb{R}^{n_{\theta}} \times \mathbb{R}^{n_x}$ es la función de coste y $g : \mathbb{R}^{n_{\theta}} \times \mathbb{R}^{n_x}$ las restricciones. El objetivo principal en programación *multi-paramétrica* es obtener la solución θ para un rango entero de los parámetros x .

Dependiendo de si la función de coste (A.8) es lineal o cuadrática, existe dos grupo bien definidos: a) *programación lineal multi-paramétrica* or b) *programación cuadrática multi-paramétrica* . Para un detalle exhaustivo y en profundidad de programación multi-paramétrica se recomiendan los siguientes trabajos (Borrelli 2003, Baotic 2005, Grancharova and Johansen 2012).

²⁹La programación multi-paramétrica se refiere a la solución de un problema de optimización el cual depende de un parámetro x . En la literatura se usa el termino *multi-paramétrico* para hacer énfasis en el hecho de que el parámetro considerado no es un escalar, si no más bien un vector.

Apéndice B

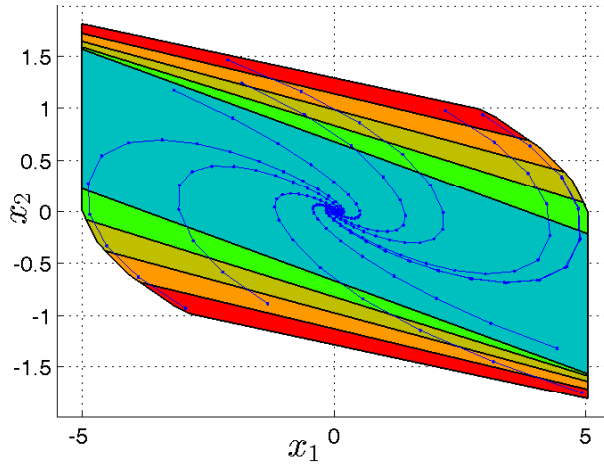
Ejemplos

B.1. Ejemplo: MPC para Sistemas LTI

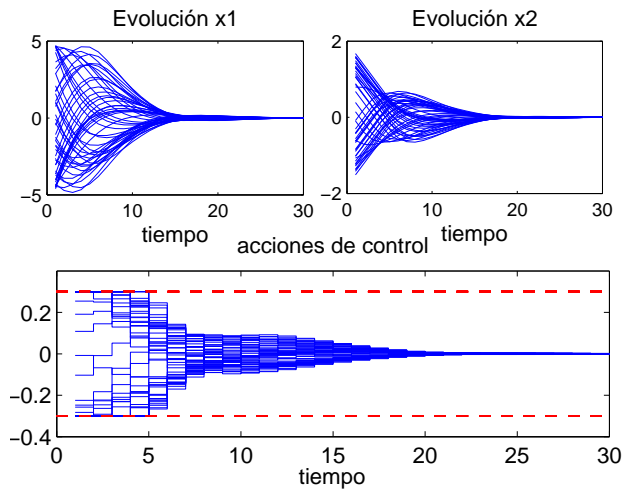
A continuación se presenta un código de implementación de un MPC para un sistema LTI utilizando *multi-parametric toolbox 3.0* (Herceg *et al.* 2013) en Matlab®.

```
1 % Control predictivo para sistemas LTI usando MPT 3.0
2 A = [1 1; 0 1]; B = [0; 1]; % Definición del sistema.
3 model=LTISystem('A',A,'B',B);
4 % Restricciones en los estados
5 model.x.min = [-5; -5];
6 model.x.max = [5; 5];
7 % Restricciones en la entrada
8 model.u.min = -0.3;
9 model.u.max = 0.3;
10 %Restricciones poliédricas
11 X = Polyhedron('lb',model.x.min,'ub',model.x.max);
12 U = Polyhedron('lb',model.u.min,'ub',model.u.max);
13 % Penalizaciones cuadráticas
14 model.u.penalty = QuadFunction(1);
15 model.x.penalty = QuadFunction(0.01*eye(2));
16 %Cálculo del conjunto terminal
17 Xf=model.LQRSet();
18 % Se calculan los conjuntos de estados Factibles a partir de Xf
19 Pn = [];
20 X0 = Xf;
21 for i = 1:N
22 Kset = model.reachableSet('X', X0, 'U', U,'direction', 'backward');
23 Kset = Kset.intersect(X).minHRep();
```

```
24 Pn = [Pn, Kset];
25 if Kset==X0
26 break
27 else
28 X0 = Kset;
29 end
30 end
31 %Obtenci\on del MPC
32 model.x.with('terminalSet');
33 model.x.with('terminalPenalty');
34 model.x.terminalPenalty=model.LQRPenalty;
35 model.x.terminalSet=Xf;
36 model.x.with('setConstraint')
37 model.x.setConstraint=Kset;
38 ctrlMPC=MPCController(model,N);
39 Simu=ClosedLoop(ctrlMPC,model); %Simular en lazo cerrado.
40 x0=[-4.0933;1.6723];
41 data=Simu.simulate(x0,30);
```



(a) Ejemplo de un sistema LTI controlado con un MPC. El conjunto más pequeño corresponde al terminal \mathbb{X}_f , mientras que el más grande (rojo) al conjunto de estados factibles \mathcal{K}_4 .



(b) Evoluciones de los estados del sistemas y acciones de control para 50 simulaciones diferentes.

Figura B-1: Ejemplo de un MPC para sistemas LTI.

B.2. Ejemplo: MPC Robusto para Sistemas LTIs Inciertos

Considere el siguiente modelo lineal discretizado con un periodo de muestreo de 0.15 minutos, el cual corresponde al sistema no isotérmico CSTR (Wan and Kothare 2003):

$$x_{k+1} = \begin{bmatrix} C_A \\ T \end{bmatrix} = \begin{bmatrix} 0,85 - 0,0986\alpha(k) & -0,0014\alpha(k) \\ 0,9864\alpha(k)\beta(k) & 0,047 + 0,01403\alpha(k)\beta(k) \end{bmatrix} \cdot x_k + \begin{bmatrix} 0,15 & 0 \\ 0 & -0,912 \end{bmatrix} \begin{bmatrix} C_{A,F} \\ F_C \end{bmatrix}, \quad y_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} C_A \\ T \end{bmatrix} \quad (\text{B.1})$$

donde x representa el vector de concentración y temperatura del reactor, *i.e.*, $x = [C_A, T]'$ y u viene formado por la concentración de alimentación del reactor $C_{A,F}$ y el flujo de refrigeración F_C , $u = [C_{A,F}, F_C]'$.

Por otro lado, $1 \leq \alpha(k) = k_0/10^9 \leq 10$ y $1 \leq \beta = -\Delta H_{rxn}/10^7 \leq 10$. Debido a que los dos parámetros inciertos α y β son independiente uno del otro, es necesario considerar un modelo politópico incierto con su cuatros vértices, los cuales representan todas las posibles combinaciones de dichos parámetros. Por tanto, el modelo incierto viene dado por:

$$\Omega = \text{conv} \left\{ \begin{bmatrix} 0,751 & -0,0014 \\ 0,986 & 0,063 \end{bmatrix}, \begin{bmatrix} 0,751 & -0,0014 \\ 9,864 & 0,189 \end{bmatrix}, \begin{bmatrix} -0,136 & -0,0014 \\ 9,864 & 0,189 \end{bmatrix}, \begin{bmatrix} -0,136 & -0,0014 \\ 98,644 & 1,451 \end{bmatrix} \right\} \quad (\text{B.2})$$

El objetivo entonces es regular las salidas C_A y T a través de las variables manipuladas $C_{A,F}$ y F_C . El problema de optimización con rendimiento robusto está definido por (1.42) sujeto a las restricciones: $|C_{A,F}| \leq 0,5 \text{ kmol/m}^3$ y $|F_C| \leq 1,5 \text{ m}^3/\text{min}$, donde $J_\infty(k)$ viene dado por (1.43) con $Q = I$ y $R = 0,2I$.

En primer lugar se ejecuta el algoritmo presentado en la Sección 1.5.5. Para ello se selecciona la siguiente secuencia de estados:

$$x_i = \left\{ \begin{bmatrix} 0,0525 \\ 0,0525 \end{bmatrix}, \begin{bmatrix} 0,0475 \\ 0,0475 \end{bmatrix}, \begin{bmatrix} 0,0425 \\ 0,0425 \end{bmatrix}, \begin{bmatrix} 0,0375 \\ 0,0375 \end{bmatrix}, \begin{bmatrix} 0,0325 \\ 0,0325 \end{bmatrix}, \begin{bmatrix} 0,0275 \\ 0,0275 \end{bmatrix} \right\}$$

Dicha secuencia de estado cumple con la condición $x_{k+1}^T \bar{Q}_i^{-1} x_{k+1} \leq 1$, obteniéndose el siguiente resultado:

$$K_i = \left\{ \begin{bmatrix} -1,5109 & -0,0127 \\ 24,7486 & 0,1359 \end{bmatrix}, \begin{bmatrix} -1,5139 & -0,0125 \\ 27,6870 & 0,1658 \end{bmatrix}, \right. \\ \left. \begin{bmatrix} -1,4978 & -0,0118 \\ 31,3196 & 0,2050 \end{bmatrix}, i = \{1, 2, 3\} \right\}$$

$$K_j = \left\{ \begin{bmatrix} -1,4418 & -0,0101 \\ 35,8726 & 0,2561 \end{bmatrix}, \begin{bmatrix} -1,3027 & -0,0066 \\ 41,6196 & 0,3202 \end{bmatrix}, \right. \\ \left. \begin{bmatrix} -0,3426 & 0,0109 \\ 51,7492 & 0,4801 \end{bmatrix}, j = \{4, 5, 6\} \right\}$$

$$\bar{Q}_i = \left\{ \begin{bmatrix} 354,4893 & 4,1154 \\ 4,1154 & 0,0917 \end{bmatrix}, \begin{bmatrix} 432,7154 & 5,1890 \\ 5,1890 & 0,1199 \end{bmatrix}, \right. \\ \left. \begin{bmatrix} 540,0147 & 6,7278 \\ 6,7278 & 0,1629 \end{bmatrix}, j = \{1, 2, 3\} \right\}$$

$$\bar{Q}_j = \left\{ \begin{bmatrix} 692,8083 & 9,0355 \\ 9,0355 & 0,2321 \end{bmatrix}, \begin{bmatrix} 920,9687 & 12,7131 \\ 12,7131 & 0,3507 \end{bmatrix}, \right. \\ \left. \begin{bmatrix} 1286,5 & 17,6 \\ 17,6 & 6 \end{bmatrix}, j = \{4, 5, 6\} \right\}$$

En la figura B-2 se muestran los conjuntos invariante elipsoidales definidos por $\mathcal{E}_i = \{x \in \mathbb{R}^{n_x} | x^T \bar{Q}_i^{-1} x \leq 1\}$, para la anterior secuencia de estados.

Partiendo de la discusión 1.5 se procede a calcular los conjuntos poliédricos invariante para poder compararlos con los elipsoidales. Para ello se hace uso del Algoritmo 3.1 presentado en (Bumroongsri and Kheawhom 2012). En la Figura B-3 muestra una comparación entre los conjuntos invariantes poliédricos los conjuntos elipsoidales. Puede observarse claramente que los conjuntos invariantes elipsoidales son más pequeños que los poliédricos, debido al conservadurismo presente en la minimización del problema (1.42). También puede observarse en la figura que para un estado dado, por ejemplo, el punto A el conjunto \mathcal{E}_1 no puede conducir dicho estado al origen, debido a

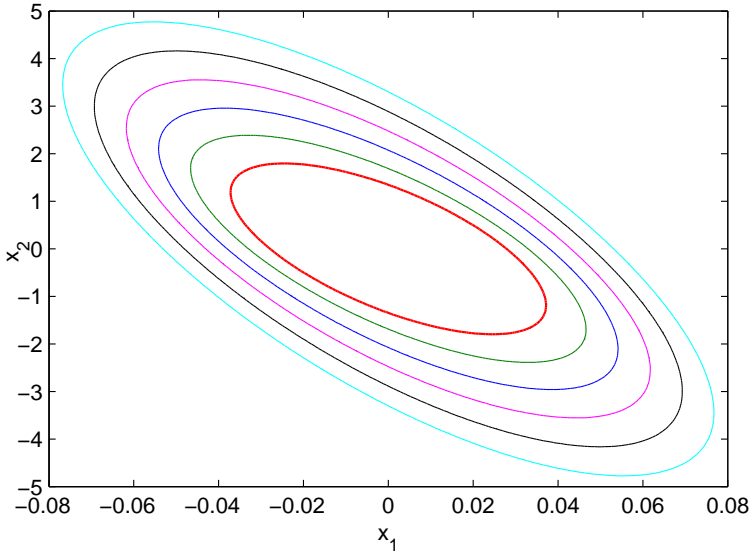
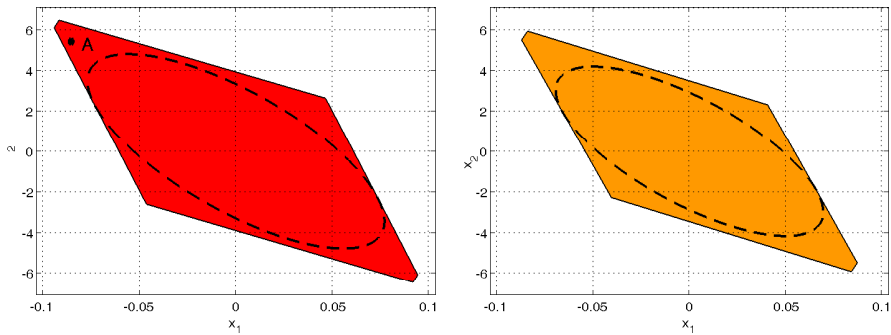
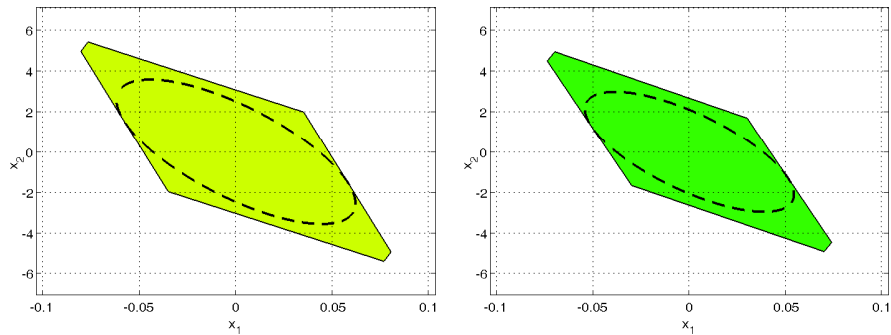


Figura B-2: Conjuntos invariante elipsoidales \mathcal{E}_i . La elipse más grande (color azul claro) representa al conjunto $\mathcal{E}_1 = \{x \in \mathbb{R}^{n_x} | x^T \bar{Q}_1^{-1} x \leq 1\}$, mientras que la siguiente elipse (color negra) representa a $\mathcal{E}_2 = \{x \in \mathbb{R}^{n_x} | x^T \bar{Q}_2^{-1} x \leq 1\}$, así sucesivamente, la elipse más pequeña (color rojo) es $\mathcal{E}_6 = \{x \in \mathbb{R}^{n_x} | x^T \bar{Q}_6^{-1} x \leq 1\}$.

que $x_{punto A} \notin \mathcal{E}_1$, sin embargo, dado que $x_{punto A} \in S_i$ es posible aplicar la ganancia K_1 al sistema.



(a) Polıtopo invariante (Rojo) S_1 . Elipse invariante \mathcal{E}_1 (líneas punteadas) (b) Polıtopo invariante (Naranja) S_2 . Elipse invariante \mathcal{E}_2 (líneas punteadas)



(c) Polıtopo invariante (Amarillo) S_3 . Elipse invariante \mathcal{E}_3 (líneas punteadas) (d) Polıtopo invariante (Verde) S_4 . Elipse invariante \mathcal{E}_4 (líneas punteadas)

Figura B-3: Comparación de los conjuntos invariante elipsoidales obtenidos con el algoritmo de la Sección 1.5.5 con los polıtopos invariante calculado con el Algoritmo 3.1 en.

Implementación en Matlab: A continuación se presenta un código en Matlab® que sirve para resolver el problema de minimización 1.48 presente en el Teorema 1.1.

```

1  %Modelo incierto de la planta
2  A1=[0.751 -0.0014;0.986 0.063];
3  A2=[0.751 -0.0014;9.864 0.189];
4  A3=[-0.136 -0.014;9.864 0.189];
5  A4=[-0.136 -0.014;98.644 1.451];
6  B1=[0.15 0;0 -0.912]; B2=B1; B3=B2; B4=B3;
7  Ai={A1,A2,A3,A4};
8  Bi={B1,B2,B3,B4};
9  Q=eye(2); %Ponderaci\on del estado
10 R=0.2*eye(2); % Ponderaci\on de las acciones de control
11 % Restrcciones
12 umax=[0.5;1.5];
13 ymax=[0.5;5];
14 %secuencia de estados x_i
15 xi={[0.0525;0.0525],[0.0475;0.0475],[0.0425;0.0425],[0.0375;0.0375],
16 [0.0325;0.0325],[0.0275;0.0275]};
17 %% Obtenci\on de los conjuntos elipsoidales invariante
18 syms x1 x2
19 for i=1:length(xi)
20     xk = xi{i};
21     [K{i},P_opt{i},gamma_opt]=mpc_lmi(Ai,Bi,C,Q,R,xk,umax,ymax);
22     ezplot([x1 x2]*P_opt{i}*[x1;x2]-1,[-0.1 0.1 -6 6])
23     hold on
24 end
25 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
26 function [K_opt,P_opt,gamma_opt]= ...
27     mpc_lmi(A,B,C,Qest,R,xk,umax,ymax)
28 %% Entradas: A,B: Lista de matrices A={A1,A2,...} B={B1,B2,...}.
29 %%          Qest,R: Matrices de ponderaciones
30 %%          xk, estado medido;
31
32 vertices = length(A);
33 [n,nu]=size(B{1});
34 [n,ny]=size(C);
35
36 %% VARIABLES LMI'S
37 Q= sdpvar(n,n);
38 Y= sdpvar(nu,n);
39 gamma= sdpvar(1);
40 X= sdpvar(nu,n);
41 Z= sdpvar(n,n);
42
43 F=[];
44 for j=1:vertices

```

```
44 Lmi_1=[Q (Q*A{j}'+Y'*B{j}') Q*sqrt(Qest) Y'*sqrt(R);...
45      (A{j}+Q+B{j})*Y) Q zeros(2,2) zeros(2,2);...
46      sqrt(Qest)*Q zeros(2,2) gamma*eye(2) zeros(2,2);...
47      sqrt(R)*Y zeros(2,2) zeros(2,2) gamma*eye(2)];
48 Lmistate=[Z C*(A{j}*Q+B{j}*Y)];...
49 (A{j}*Q+B{j}*Y)'*C' Q];
50 F=[F,Lmi_1>=0,Lmistate>=0];
51 end
52 %%%restriccionesLMI : Entrada y estados
53 for r=1:ny
54 F=[F,Z(r,r)<=ymax(r)^2];
55 end
56 lmiu=[X Y;Y' Q];
57 lmirest=[1 xk';xk Q];
58 F=[F,lmiu>=0,lmirest>=0];
59 for r=1:nu
60 F=[F,X(r,r)<=umax(r)^2];
61 end
62 optimize(F,gamma);
63 gamma_opt=value(gamma);
64 P_opt=inv(value(Q));
65 K_opt=value(Y*inv(value(Q)));
66 end
```

Cuadro B.1: Parámetros del sistema económico.

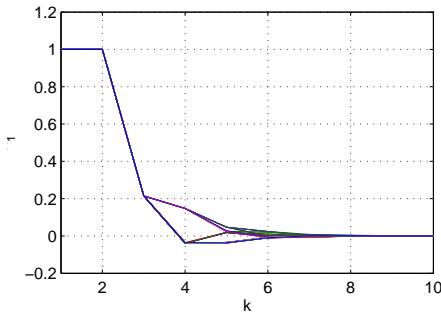
Vértices	Parámetros					
	A_i^l			B_i^l		
Vértices de $i = 1$	$\begin{bmatrix} 0 & 1 \\ -2,6 & 3,3 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \\ -2,4 & 3,1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$
Vértices de $i = 2$	$\begin{bmatrix} 0 & 1 \\ -4,4 & 4,6 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \\ -4,2 & 4,4 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$
Vértices de $i = 3$	$\begin{bmatrix} 0 & 1 \\ 5,4 & -5,3 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \\ 5,2 & -5,1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$
vértice de \mathbb{P}	$\begin{bmatrix} 0,55 & 0,23 & 0,22 \\ 0,36 & 0,35 & 0,29 \\ 0,32 & 0,16 & 0,52 \end{bmatrix}$			$\begin{bmatrix} 0,79 & 0,11 & 0,10 \\ 0,27 & 0,53 & 0,20 \\ 0,23 & 0,07 & 0,70 \end{bmatrix}$		
$Q_{(1)} = \begin{bmatrix} 0,01 & 0 \\ 0 & 0,01 \end{bmatrix}$	$Q_{(2)} = \begin{bmatrix} 0,01 & 0 \\ 0 & 0,01 \end{bmatrix}$	$Q_{(3)} = \begin{bmatrix} 0,01 & 0 \\ 0 & 0,01 \end{bmatrix}$				
$R_{(1)} = 0,01$	$R_{(2)} = 0,01$	$R_{(3)} = 0,01$				

B.3. Ejemplo: MPC Robusto para MJLS Inciertos

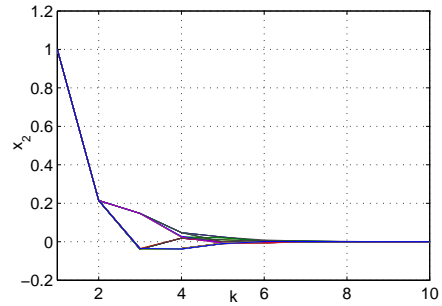
A continuación se presenta un ejemplo para verificar los resultados propuesto en la Sección 2.5. Para ellos, considérese un MJLS (2.3) el cual consiste en un sistema económico. El sistema tiene tres modos de operación, *i.e.*, $i \in \mathcal{M}$, $\mathcal{M} = \{1, 2, 3\}$. Los parámetros del sistema se presentan en la Tabla B.1. Se selecciona como estado inicial $x_0 = [1, 1]^T$.

Para este sistema se han diseñado dos controladores diferentes. El primero, será el MPC sin restricciones de (2.28), mientras que el segundo corresponde a un MPC con restricciones dado por (2.29a), asumiendo restricciones en la acción de control $|u| \leq 1$. En ambos diseño se considera horizonte $N = 3$.

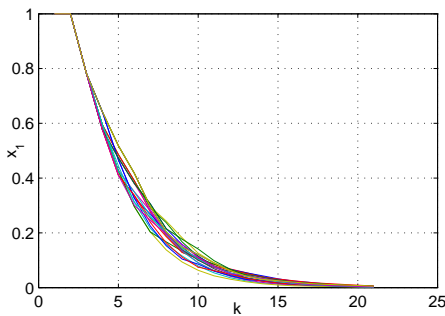
Se procede a realizar un total de 50 simulaciones con secuencias aleatorias de la variable θ_k , pero con $\theta_0 = 1$ para cada una de ellas. En la Figura B-4 se presenta las variables de estados x_1, x_2 para dichas simulaciones, para ambos casos en consideración. Se aprecia que los estados alcanzan el origen en ambas simulaciones. Puede observarse también que los estados para el caso sin restricciones llegan al punto de equilibrio en aproximadamente 8 instante de tiempo, mientras que en el caso con restricciones tarda 20 instante en alcanzarlo. Esto es debido a que la acción de control sin restricciones son más enérgica, en algunos casos, siendo el triple que las secuencias de control con restricciones.



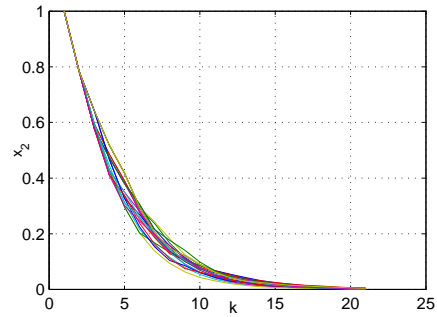
(a) Evolución del estado x_1 (sin restricciones)



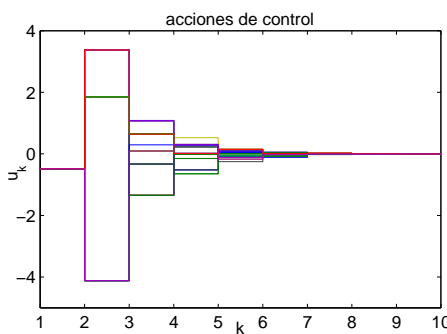
(b) Evolución del estado x_2 (sin restricciones)



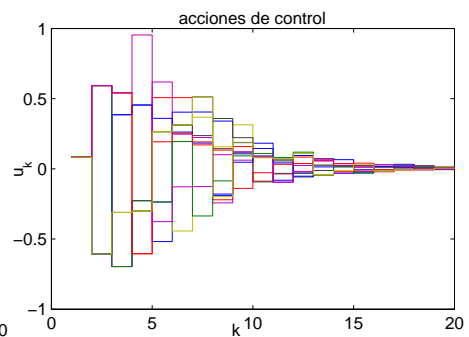
(c) Evolución del estado x_1 (con restricciones)



(d) Evolución del estado x_2 (con restricciones)



(e) acciones de sin restricciones



(f) acciones de con restricciones

Figura B-4: Comparativa de evolución del estado controlado con MPC sin restricciones ((a)–(b)) y con restricciones ((c)–(d))