Hernandez-Mejias, MA.; Sala, A. (2017). Reliability and time-to-failure bounds for discrete-time constrained Markov jump linear systems. International Journal of Robust and Nonlinear Control. 27:1773-1791. doi:10.1002/rnc.3635

INTERNATIONAL JOURNAL OF ROBUST AND NONLINEAR CONTROL
Int. J. Robust. Nonlinear Control 0000; **00**:1–22
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/rnc

Reliability and Time-to-Failure Bounds for Discrete-Time Constrained Markov Jump Linear Systems

Manuel A. Hernández-Mejías, Antonio Sala*†

Dept. Ing. Sistemas y Automática. Universitat Politècnica de València. Cno Vera s/n, E-46022 Valencia, Spain.

SUMMARY

This paper 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* heuristic approach was proposed in an earlier work [1] 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. Copyright © 0000 John Wiley & Sons, Ltd.

Received ...

KEY WORDS: Reliability analysis, constrained Markov-Jump linear systems, invariant sets, fault-tolerant control.

1. INTRODUCTION

Avoiding constraint violation in closed-loop operation of a control system is a relevant problem; it is, for instance, one of the motivations behind the success of model predictive control [2].

In a worst-case setting, the concept of invariant sets [3, 4] helps solving the above problem: by definition, any initial state in the referred invariant sets can be guaranteed to never violate constraints for all future time ('*safe*' sets). If state and input constraint sets are polyhedral, (robust) invariant and control-invariant sets can be computed with well known algorithms [4] using multi-parametric toolbox (MPT) [5]. The framework allows extending set invariance results to switching [6, 7, 8] and piecewise affine systems [9]; in such systems, collections of polytopes [8] should be handled in some cases.

*Correspondence to: Antonio Sala, Dept. Ing. Sistemas y Automática. Universitat Politècnica de València. Cno Vera s/n, E-46022 Valencia, Spain.
†Email: asala@isa.upv.es

The above worst-case analysis may be conservative when there is a probabilistic description of the disturbances or the switching between different possible system models (operation *modes*): basically, in such settings the 'probability of constraint violation' should be managed somehow. Of course, the robust invariant sets in the cited literature will be sets where probability of not violating constraints (to be denoted as *reliability* [1]) is one, but there might be larger sets where such reliability is close to one; determining them can be useful for practical purposes. There might be situations in which constraint violation is ultimately unavoidable (zero reliability), say, with faults in unstable systems; in such a case, an alternative *mean time to failure* (MTTF) measure [10], understood as the average time trajectories take to hit constraints, should be considered. Obviously, reliability and MTTF will depend on how close initial state is from constraint boundaries.

The above issues arise in practice in, for instance, networked control systems with random packet losses [11], control of processes subject to randomly-occurring faults (fault-tolerant control [12]), predictive control under scenarios discarding some possible outcomes [13], etc.

Specifically, this paper extends the classical set-invariance ideas 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) [14, 15]). 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 [16, Lemma 16], [7, 17]. Of course, linear matrix inequalities may be used to handle ellipsoidal invariant sets for MJLS [18], but this approach will not be pursued in this work.

As discussed above, the work [1] defined *reliability* as the probability of avoiding future constraint violations, but only 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.

Results in the above-cited paper apply only to suboptimal reliability computations for *stabilisable*, *undisturbed* MJLS. The specific objective of this work is overcoming such issues by:

1. Devising better controller options than the greedy one, exploiting the possibility of 'betting' on more than one sequence (via an intersection operator). In fact, an *optimal reliability* controller will be proven under a finitely-generated assumption.
2. Generalising the results to disturbed and non-stabilisable systems where, in fact, robust invariant sets might be non-existent. In such a case, both reliability and MTTF bounds will be considered, albeit the results cannot be guranteed to be optimal.

The basic tool will be based on dynamic-programming argumentations, whose 'steps' will be identified with '1-step sets' in the above-cited invariance-based control.

This paper is organized as follows: preliminary, definitions and notation are given in next section, as well as the problem statement. Section 3 presents the main result of this paper, proposing generalised 1-step and intersection operators, used in an algorithm which obtains a progressively more accurate reliability bound; two initialisation options are proposed and compared. The MTTF

Prepared using rncauth.cls
DOI: 10.1002/rnc

bound is discussed in Section 4, for zero-reliability set-ups (large disturbances, unstable modes, ...). Numerical examples appear in Section 5. Finally, a conclusion section closes the paper. An Appendix discusses algorithm implementation details and computational improvements.

2. PRELIMINARIES

Consider a non-autonomous discrete time-varying linear system:

$$x_{k+1} = A_{\theta_k} x_k + B_{\theta_k} u_k + E_{\theta_k} w_k, \quad \theta_k \in \mathcal{M},\ k \geq 0 \tag{1}$$

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 (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 $\mathcal{C} \in \mathbb{C}(\mathbb{R}^n)$ 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 [5].

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, $\boldsymbol{\theta} = \{\theta_0, \dots, \theta_{N-1}\}$ for (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.

$Pr(\boldsymbol{\theta})$ will denote the probability of a particular sequence $\boldsymbol{\theta}$ conditioned to its first element, i.e., $Pr(\boldsymbol{\theta}) := \Pi_{j=1}^{N-1} \pi_{\theta_j \theta_{j-1}}$. All finite-length admissible sequences have $Pr(\boldsymbol{\theta}) \neq 0$.

Prepared using rncauth.cls
DOI: 10.1002/rnc

One-step sets. When there is no mode information available to the controller, system (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 in [3, 4]. When the mode information is known by the controller, a more flexible definition of 1-step set is proposed in [16, 1], quoted below:

Definition 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 $i \in \mathcal{M}$ is defined as:

$$Q_i(\mathcal{C}) := \{x \in \Omega^{[i]} : \exists u \in \mathbb{U}^{[i]} \; such \; that \; A_i x + B_i u + E_i w \in \mathcal{C} \; \forall w \in \mathbb{W}^{[i]}\} \tag{2}$$

and, also, $Q_i(\emptyset) := \emptyset$. The 1-step set in the augmented space Ω^* is redefined, for $\mathcal{C}^* \subset \Omega^*$, as [1]:

$$Q^*(\mathcal{C}^*) := \{(x_0, i) \in \Omega^* : \exists u \in \mathbb{U}^{[i]} \; s.\,t. \; (A_i x_0 + B_i u + E_i w, j) \in \mathcal{C}^* \; \forall j \in \mathcal{S}_i, \forall w \in \mathbb{W}^{[i]}\} \tag{3}$$

Remark 1
Knowledge of the mode i by the controller is implicitly integrated in (3). Analogously to well-known literature, [4], 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 3 or 4, in later sections.

Definition 2
A set $\mathcal{C}^* \subset \Omega^*$ is mode-dependent controllable if $\mathcal{C}^* \subset Q^*(\mathcal{C}^*)$. The largest of such sets is the maximal mode-dependent controllable set, $\mathbb{K}^*_\infty$.

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, [4], $\mathbb{K}^*_\infty$ can be computed iterating $\mathbb{K}^*_l = Q^*(\mathbb{K}^*_{l-1})$ until convergence, from initial $\mathbb{K}^*_0 = \Omega^*$; see [16, 1]. Once the converged $\mathbb{K}^*_\infty$ is available, for any $(x, i) \in \mathbb{K}^*_\infty$, any controller which steers a state in $\mathbb{K}^{[i]}_\infty$ to the successor set:

$$\mathbb{S}^{[i]} := \cap_{j \in \mathcal{S}_i} \mathbb{K}^{[j]}_\infty \tag{4}$$

achieves the above-mentioned invariance. $\mathbb{K}^*_\infty$ will be denoted as *terminal set* and the associated controller as *terminal controller*[†].

If there exists a finite l such that $\mathbb{K}^*_l = \mathbb{K}^*_{l+1} = \cdots = \mathbb{K}^*_\infty$, we will say that $\mathbb{K}^*_\infty$ 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 ideas, given $l \geq 0$, if $(x_0, \theta_0) \notin \mathbb{K}^*_l$ there exists a mode sequence of length $l + 1$ and a *worst-case* disturbance such that no admissible control action can avoid exiting Ω^*.

2.1. Problem statement

By definition, for initial mode-state conditions outside the terminal set $\mathbb{K}^*_\infty$ there does not exist a controller with probability of success equal to 1, i.e., trajectories starting from such initial states

[†]In 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 [14]). 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.

Prepared using rncauth.cls
DOI: 10.1002/rnc

will violate constraints under some admissible mode and disturbance sequences. The objective of this paper is to optimise the '*reliability*' understood as the likelihood[‡] 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).

This work will propose iterative algorithms, inspired on dynamic programming, to compute reliability bounds for initial $(x_0, \theta_0) \notin \mathbb{K}_\infty^*$ 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 $\mathbb{K}_\infty^{[j]}$ 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 4 will do the same for the mean-time-to-failure case, albeit briefly as developments will be parallel to those in Section 3.

3. RELIABILITY BOUND COMPUTATION

Definition 3
Given x_0, a control law $u(x, \theta)$, an integer horizon k, and a mode sequence $\boldsymbol{\theta} \in \mathcal{AS}(\theta_0, N+1)$ of length $(N+1) > k \geq 1$, the controller-sequence pair $\{u(\cdot,\cdot), \boldsymbol{\theta}\}$ 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 $j = 0, \ldots, k-1$. For a fixed controller $u(\cdot,\cdot)$, we will denote as $\mathcal{SS}(x_0, \theta_0, k)$ the set of admissible sequences $\boldsymbol{\theta}$ with length $k+1$ such that $\{u(\cdot,\cdot), \boldsymbol{\theta}\}$ 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 $\boldsymbol{\theta}$ is k-step successful for (x_0, θ_0).

Note that, as an equivalent recursive characterisation, $\boldsymbol{\theta}$ 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 $\{\theta_1, \ldots, \theta_N\}$ is $(k-1)$-step successful for $(x_1(x_0, \theta_0, w), \theta_1)$ for all $w \in \mathbb{W}^{[\theta_0]}$.

Definition 4
Given a control law $u(x, \theta)$:

[‡]As discussed in the introduction, no 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 [13] (details left to the reader).

Prepared using rncauth.cls
DOI: 10.1002/rnc

- 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

$$RL_k(x_0, \theta_0) := \sum_{\boldsymbol{\theta} \in \mathcal{SS}(x_0, \theta_0, k)} Pr(\boldsymbol{\theta}) \tag{5}$$

  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 $RL(x_0, \theta_0) := \lim_{k \to \infty} RL_k(x_0, \theta_0)$.

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 4 means that reliability is the probability of not violating constraints in future time from (x_0, θ_0) under *worst-case* disturbances. With $\mathbb{K}_l^*$ in previous section (for the particular controller under scrutiny, see Remark 1), it can be proved that:

$$(x, \theta) \in \mathbb{K}_l^* \quad \Leftrightarrow \quad RL_l(x, \theta) = 1 \tag{6}$$

Proposition 1

For a given controller $u(x, i)$, reliability fulfils the following recursive equation:

$$RL_l(x, i) = \min_{w \in \mathbb{W}^{[i]}} \sum_{j \in \mathcal{S}_i} \pi_{ji} RL_{l-1}(A_i x + B_i u(x, i) + E_i w, j) \quad \forall l \geq 1 \tag{7}$$

Proof

Starting with RL_0, it is straightforward to see that the assertion is true for RL_1. For larger l, given $\boldsymbol{\theta} = \{i, j, \theta_2, \dots\}$ 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 (7). So, the reasoning can be applied by induction to any $l \geq 1$. □

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 $RL_l(x, i) = RL(x, i)$.

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:

$$RL_l^{opt}(x, i) = \max_{u \in \mathbb{U}^{[i]}} \min_{w \in \mathbb{W}^{[i]}} \sum_{j \in \mathcal{S}_i} \pi_{ji} RL_{l-1}^{opt}(A_i x + B_i u + E_i w, j) \tag{8}$$

Prepared using rncauth.cls
DOI: 10.1002/rnc

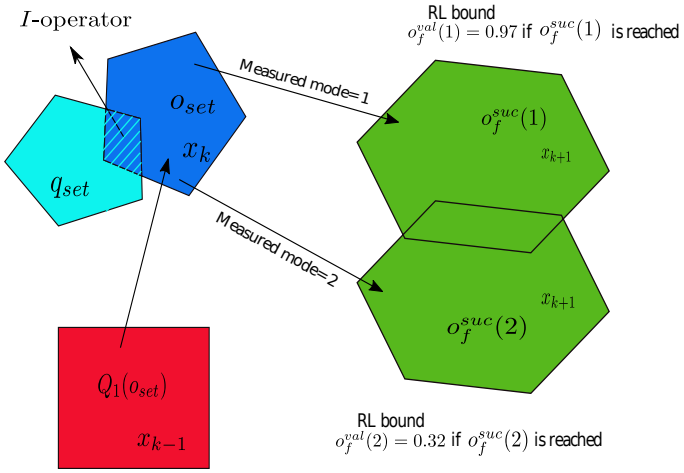


Figure 1. Illustration of the data structure $\mathcal{O}$ and operators Q_i, I.

However, obtaining such optimal-reliability controller might require a large computational effort. As discussed in the introduction, [1] 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:

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:

$$o = \langle o_{set}, o_f \rangle = \left\langle o_{set}, \left\{ \begin{array}{cc} o_f^{val}(1), & o_f^{suc}(1) \\ \vdots & \vdots \\ o_f^{val}(M), & o_f^{suc}(M) \end{array} \right\} \right\rangle \tag{9}$$

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.

The justification of the structure (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 $\mathbb{K}_\infty^*$ the successor proposed in (4) for the terminal controller. Figure 1 illustrates these concepts for a 2-mode setting: for instance, the blue pentagon depicts an hypothetical o_{set}; for x_k in such set, there exists a current control policy which drives the next

DOI: 10.1002/rnc

state x_{k+1} to one of the green hexagons, depending on mode θ_k being either 1 or 2. The reliability estimate under such controller would be, say, 0.97 if $\theta_k = 1$, and 0.32 if $\theta_k = 2$.

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 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 $h = \max(f,g)$ is the element $h \in \mathcal{F}$ whose image, for $j \in \mathcal{M}$ is:

$$h(j) := \begin{cases} (f^{val}(j), f^{suc}(j)) & f^{val}(j) \geq g^{val}(j) \\ (g^{val}(j), g^{suc}(j)) & \text{otherwise} \end{cases} \tag{10}$$

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 $I(o,r) \succ I(q,r)$.

Resorting to Figure 1, the I-operator would evaluate the reliability for states in the shaded blue-cyan region considering them to be in both o_{set}, discussed before, and in q_{set}, whose successors and value function have not been represented to avoid cluttering the figure. $I(o,q)$ creates a new element of $\mathcal{O}$ associating to the small intersection set the most reliable successor/value option.

One-step operations over $\mathcal{O}$. Abusing the notation, the one-step set $Q_i(\mathcal{C})$ in Definition 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 6 (1-step operator over $\mathcal{O}$)
The 1-step operator $Q_i : \mathcal{O} \mapsto \mathcal{O}$, will be defined, for $o \in \mathcal{O}$, as: $Q_i(o) := (Q_i(o_{set}), Q_i^f(o_f, o_{set}))$, where the operator $Q_i^f(r,s)$ 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:

$$g(j) := \begin{cases} (0, \mathbb{R}^n) & j \neq i \\ (\sum_{\kappa \in \mathcal{S}_i} \pi_{\kappa i} r^{val}(\kappa), s) & j = i \end{cases} \tag{11}$$

For instance, Figure 1 depicts a red square s_{set} being $s = Q_1(o)$: If $x_{k-1} \in s_{set}$ were reached with mode $\theta_{k-1} = 1$, every point in the red square could be steered to the blue o_{set} for any admissible disturbance; reliability if such action were taken, i.e., setting $s_f^{suc}(1) = o_{set}$, would be $s_f^{val}(1) = 0.97 * \pi_{11} + 0.32 * \pi_{21}$; of course, if the red square were reached with $\theta_{k-1} = 2$ no bound can be asserted apart from the trivial reliability bound $s_f^{val}(2) = 0$, with $s_f^{suc}(2) = \mathbb{R}^n$. As a further example, note also that, in order to make all the discussed concepts meaningful,

DOI: 10.1002/rnc

$o_{set} \subset Q_1(o_f^{suc}(1))$ and $o_{set} \subset Q_2(o_f^{suc}(2))$; this will be guaranteed by the algorithm discussed next. Numerical illustration of the above operators in a first-order case appear later on in Example 1.

Guaranteed-reliability controller algorithm. From the above considerations, Algorithm 1 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 combinations[§]). 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.

Algorithm 1 l-step reliability bound.

1. Let $l = 1$. Let $O^{[0]}$ be a M-element set $\{o_1, \ldots, o_M\}$ such that:
$$(o_j)_{set} = \Omega^{[j]};\ (o_j)_f^{val}(j) = 1, \text{ and } (o_j)_f^{val}(i) = 0 \text{ (for } i \neq j) \text{ ; } \forall i,\ (o_j)_f^{suc}(i) = \mathbb{R}^n.$$

2. Compute the relevant 1-step set for all elements of $O^{[l-1]}$, conforming a set $D^{[l]} \subset \mathcal{O}$ as:
$$D^{[l]} := \bigcup_{o \in O^{[l-1]},\, j \in \mathcal{M}} Q_j(o), \tag{12}$$

3. Compute the intersections for all obtained elements of $D^{[l]}$, initialising with $Y_0^{[l]} = \emptyset$ and repeating until $Y_h^{[l]} = Y_{h-1}^{[l]}$, increasing h:
$$Y_h^{[l]} := \bigcup_{o_1, o_2 \,\in\, D^{[l]} \cup Y_{h-1}^{[l]}} I(o_1, o_2) \tag{13}$$

4. Update:
$$O^{[l]} := D^{[l]} \cup Y_h^{[l]} \tag{14}$$

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 (13) ensures $I(O_x^{[l]}) \in O^{[l]}$.

Theorem 1

Given any $o \in O^{[l]}$ from Algorithm 1, and initial condition (x_0, θ_0),

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 $l \geq 0$.

[§]This 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 though of in a 'formal' sense, and not as an implementation proposal.

Prepared using rncauth.cls
DOI: 10.1002/rnc

2. Define:

$$\widehat{RL}(x_0,\theta_0,l) := \max_{o\in O_{x_0}^{[l]}} o_f^{val}(\theta_0), \qquad \hat{o}(x_0,\theta_0,l) := \arg\max_{o\in O_{x_0}^{[l]}} o_f^{val}(\theta_0) \tag{15}$$

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 $i = \theta_0$, $q = Q_i(o) \in D^{[l]}$. 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.,

$$RL_l(x,i) \geq \sum_{j\in\mathcal{S}_i} \pi_{ji} o_f^{val}(j) \tag{16}$$

Note that the right-hand side of the above inequality is actually stored in q_f (and the corresponding successor set), see (11). This proves the first assertion in the theorem statement.

Of course, if the state x_0 belongs to several 1-step sets Q_i from different elements of $O^{[l-1]}$, the above reasoning can be made for each of them. The definitions in (15) just take the element $o \subset O_{x_0}^{[l]}$ yielding the largest reliability estimate. Last, an induction argumentation ensures that the theorem holds for any positive l. □

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 (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 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) gets converted to:

$$RL_l^{opt}(i,x_0) = \max_{u\in\mathbb{U}^{[i]}} \sum_{j\in\mathcal{S}_i} \pi_{ji} RL_{l-1}^{opt}(A_i x_0 + B_i u, j) = \max_{u\in\mathbb{U}^{[i]}} \sum_{j\in\mathcal{S}_i} \pi_{ji} \hat{o}_f^{val}(A_i x_0 + B_i u, j, l-1) \tag{17}$$

DOI: 10.1002/rnc

where the last equality is stated by assumption. Now,

$$\max_{u\in\mathbb{U}^{[i]}} \sum_{j\in\mathcal{S}_i} \pi_{ji}\hat{o}_f^{val}(A_ix_0+B_iu,j,l-1) = \max_{u\in\mathbb{U}^{[i]}} \sum_{j\in\mathcal{S}_i} \pi_{ji} \max_{o\in O_{x_1}^{[l-1]}} o_f^{val}(j)$$
$$= \max_{u\in\mathbb{U}^{[i]}} \sum_{j\in\mathcal{S}_i} \pi_{ji}\cdot[q(x_1,l-1)]_f^{val}(j) = \max_{u\in\mathbb{U}^{[i]}} \max_{o\in O_{x_1}^{[l-1]}} \sum_{j\in\mathcal{S}_i} \pi_{ji}\cdot o_f^{val}(j) \quad (18)$$

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 $x_0\in Q_i(o_{set})$. Hence,

$$\max_{u\in\mathbb{U}^{[i]}} \max_{o\in O_{x_1}^{[l-1]}} \sum_{j\in\mathcal{S}_i} \pi_{ji}\cdot o_f^{val}(j) \overset{*}{=} \max_{o\in O^{[l-1]}s.t.x_0\in Q_i(o_{set})} \sum_{j\in\mathcal{S}_i} \pi_{ji}\cdot o_f^{val}(j) = \max_{o\in O_{x_0}^{[l]}} o_f^{val}(i) = \widehat{RL}(i,x_0,l) \quad (19)$$

An induction argument ends the proof. □

The equality in (19) marked with '$\overset{*}{=}$' does not hold in disturbed cases; details omitted for brevity.

Corollary 1
In the undisturbed case, if Algorithm 1 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. □

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.

3.2. Terminal set based algorithm

In case of not having enough computational resources for convergence, if a subset of the terminal set $S^*\subset\mathbb{K}_\infty^*$ is available, the algorithm could be seeded replacing initialisation in step 1 by a initial value function ('reward') equal to one for being inside S^*, and zero elsewhere. This is the idea exploited in [1]: forcing the terminal set to be reached for at least one sequence. S^* may be generated by a finite λ-contractive search [3, Theorem 3.2], or via LMIs [19]. 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 1. Let us discuss such issues in more detail.

For a given controller $u(x,\theta)$, let us denote the likelihood of reaching $\mathbb{K}_\infty^*$, under a worst-case disturbance, in *at most* l steps as $RT_l(x_0,\theta_0)$, and $RT(x_0,\theta_0):=\lim_{l\to\infty}RT_l(x_0,\theta_0)$. Evidently, as

DOI: 10.1002/rnc

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 3
Given (x_0, θ_0), and a control law $u(x, \theta)$, if the terminal set associated to the controller, $\mathbb{K}^*_\infty$ 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 $\widehat{\mathbf{\Theta}}_h(x_0, \theta_0)$ 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 $w_j(\boldsymbol{\theta})$, $j = 0, \ldots, h-1$, for every sequence $\boldsymbol{\theta} \in \widehat{\mathbf{\Theta}}_h(x_0, \theta_0)$, such that the controller in consideration keeps (x_j, θ_j) in $(\Omega^* \sim \mathbb{K}^*_l)$, for $j = 0, \ldots, h$.

Consider l to be the smallest integer such that $\mathbb{K}^*_\infty = \mathbb{K}^*_l$. 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 $\mathbb{K}^*_l$ robustly invariant), or

2. remain for infinite time in $\Omega^* \sim \mathbb{K}^*_l$ (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 2, if (x_0, θ_0) is not in $\mathbb{K}^*_l$, 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 $\rho_l := \min_{\theta \in \mathcal{M}} \min_{\boldsymbol{\theta} \in \mathcal{AS}(\theta, l+1)} Pr(\boldsymbol{\theta})$, and, hence, the cumulative probability (sum) of all non l-step successful sequences is, too, greater than ρ_l so:

$$(x_0, \theta_0) \in (\Omega^* \sim \mathbb{K}^*_l) \Rightarrow RL_l(x_0, \theta_0) \leq 1 - \rho_l < 1 \tag{20}$$

From this argumentation, $Pr(\widehat{\mathbf{\Theta}}_l(x_0, \theta_0)) \leq RL_l(x_0, \theta_0) \leq 1 - \rho_l$, 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 $Pr(\widehat{\mathbf{\Theta}}_h)$ tends to zero as h tends to infinity, for $h = l, 2l, 3l, \ldots$.

Note that, considering length-$(2l+1)$ sequences, for such sequences to be $(2l)$-step succesful 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:

DOI: 10.1002/rnc

$$Pr(\widehat{\boldsymbol{\Theta}}_{2l}(x_0,\theta_0)) = \sum_{\boldsymbol{\theta}\in\widehat{\boldsymbol{\Theta}}_{2l}(x_0,\theta_0)} Pr(\boldsymbol{\theta}) = \sum_{\boldsymbol{\theta}\,\in\,\widehat{\boldsymbol{\Theta}}_l(x_0,\theta_0)} \left(Pr(\boldsymbol{\theta}) \sum_{\boldsymbol{\theta}'\,\in\,\widehat{\boldsymbol{\Theta}}_l(x_l,\theta_l)} Pr(\boldsymbol{\theta}') \right)$$
$$\leq \sum_{\boldsymbol{\theta}\,\in\,\widehat{\boldsymbol{\Theta}}_l(x_0,\theta_0)} (Pr(\boldsymbol{\theta})(1-\rho_l)) \leq (1-\rho_l)^2 \quad (21)$$

Indeed, in the above expression, the following fact has been used: the probability of a length-$(2l+1)$ sequence $\boldsymbol{\theta} := \{\theta_0, \theta_1, \ldots, \theta_{2l}\}$ being $2l$-step successful for (x_0, θ_0) requires:

1. $\{\theta_0, \ldots, \theta_l\}$ being l-step successful for (x_0, θ_0), and
2. $\{\theta_l, \ldots, \theta_{2l}\}$ 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 $\mathbb{K}_l^* = \mathbb{K}_{2l}^* = \mathbb{K}_\infty^*$ and, by definition of $\widehat{\boldsymbol{\Theta}}_{2l}$, (x_j, θ_j) lies in $\Omega^* \sim \mathbb{K}_\infty^*$ for $j = 0, \ldots, 2l$.

Now, if $h = \nu \cdot l$, being ν any arbitrary natural number, a similar argumentation can prove that:

$$Pr(\widehat{\boldsymbol{\Theta}}_{\nu\cdot l}(x_0,\theta_0)) \leq \ (1-\rho_l)^\nu$$

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 $RT(x_0, \theta_0) = RL(x_0, \theta_0)$. □

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 1 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 hast 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. An Appendix 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 5.

4. MEAN TIME TO FAILURE BOUND COMPUTATION

Definition 7

Given a control law $u(x_k, \theta_k)$, initial conditions (x_0, θ_0) and $\boldsymbol{\theta} \in \mathcal{AS}(\theta_0, N+1)$, the sequence's **guaranteed time to failure** $k^*(x_0, \boldsymbol{\theta})$ is defined as either the minimum natural number k such that the pair $\{u, \boldsymbol{\theta}\}$ is not k-step successful for (x_0, θ_0), or $(N+1)$ if it is N-step successful, or zero if $(x_0, \theta_0) \notin \Omega^*$.

Prepared using rncauth.cls
DOI: 10.1002/rnc

Definition 8
Given a control law $u(x_k, \theta_k)$, the ***robust mean time to failure*** $MTTF(x_0, \theta_0)$ is defined as:

$$MTTF(x_0, \theta_0) := \lim_{l \to \infty} M_l(x_0, \theta_0)$$

where

$$M_l(x_0, \theta_0) := \sum_{\boldsymbol{\theta} \in \mathcal{AS}(\theta_0, l+1)} k^*(x_0, \boldsymbol{\theta}) \cdot Pr(\boldsymbol{\theta}) \tag{22}$$

As in Definition 4, MTTF involves worst-case (non-random) disturbances. Simple modifications to the set-up in Section 3 will be needed to handle the new performance measure. Actually, the needed changes are twofold: first, replacing $Q_i^f(r, s)$ in Definition 6 by the one in Definition 9 below and, second, modifying algorithm initialisation. Let us discuss such issues in detail.

Definition 9 (1-step MTTF operator on $\mathcal{F}$)
Let $r \in \mathcal{F}$, $s \in \mathbb{C}(\mathbb{R}^n)$. The operator $Q_i^f(r, s)$ for $i \in \mathcal{M}$ is defined as the element $g \in \mathcal{F}$ below:

$$g(j) := \begin{cases} (0, \mathbb{R}^n) & j \neq i \\ \left(1 + \sum_{k=1}^{M} \pi_{ki} r^{val}(k), s\right) & j = i \end{cases} \tag{23}$$

The modified 1-step operator is motivated by the fact that the recursion (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 $O = \{o_1, \ldots, o_M, q_1, \ldots q_M\}$, with o_i being identical to the one in Algorithm 1, and q_j defined as:
$(q_j)_{set} \subset \mathbb{K}_\infty^{[j]}$; $(q_j)_f^{val}(j) = \infty$, $(q_j)_f^{suc}(j) \subset \mathbb{S}_i$, 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 2
For a given controller $u(x_k, \theta_k)$, $MTTF(x_k, \theta_k)$ is finite if and only if $\mathrm{RL}(x_k, \theta_k) = 0$.

Proof
For any l, we can assert, from (22), that

$$M_l(x_0, \theta_0) = RL_l(x_0, \theta_0) * (l+1) + \sum_{\boldsymbol{\theta} \in \mathcal{AS}(\theta_0, l+1):\, k^*(x_0, \boldsymbol{\theta}) < (l+1)} k^*(x_0, \boldsymbol{\theta}) \cdot Pr(\boldsymbol{\theta}) \tag{24}$$

because RL_l is computed only with those sequences which are l-step successful so their guaranteed time to failure is $l + 1$. Hence, $M_l \geq RL_l * (l + 1)$.

Prepared using rncauth.cls
DOI: 10.1002/rnc

If $RL > 0$, as $RL_l > RL$ for any l, $RL > 0$ implies $MTTF = lim_{l\to\infty} M_l = \infty$. Conversely, we can write $RL_l \leq M_l/(l+1)$, so, taking the limit $l \to \infty$, if $\lim_{l\to\infty} M_l$ is finite (by assumption), then forcedly $RL = 0$. □

Note that, as a consequence of this proposition, the above-discussed initialisation can be improved by choosing $(q_j)_{set}$ to be any non-zero reliability set (instead of the terminal one), if the results of a previously-run reliability algorithm were available.

5. 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 $\mathbb{K}_\infty^{[1]} = [-0.833, +0.833]$, $\mathbb{K}_\infty^{[2]} = \{0\}$. 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:

- $o_1 =< 0.8333, \{o_1^{val}(1) = 1, o_1^{suc}(1) = 0, o_1^{val}(2) = 0,\ o_1^{suc}(2) = \infty\} >$,
- $o_2 =< 0, \{o_2^{val}(1) = 0, o_2^{suc}(1) = \infty, o_2^{val}(2) = 1, o_2^{suc}(2) = 0\} >$.

where, abusing the notation, a single number γ in place of a set should be understood as $\{|x| \leq \gamma\}$. Then, we can compute:

- $o_3 = Q_1(o_1) =< 1.527, \{o_3^{val}(1) = 0.9, o_3^{suc}(1) = 0.833, o_3^{val}(2) = 0, o_3^{suc}(2) = \infty\} >$,
- $o_4 = Q_2(o_1) =< 0.694, \{o_4^{val}(1) = 0, o_4^{suc}(1) = \infty, o_4^{val}(2) = 0.9, o_4^{suc}(2) = 0.833\} >$.

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: $o_4 =< 0.694, \{o_4^{val}(1) = 1, o_4^{suc}(1) = 0, o_4^{val}(2) = 0.9,\ o_4^{suc}(2) = 0.833\} >$. Now, we can set $o_5 = Q_2(o_4) =< 0.694/1.2, \{o_5^{val}(1) = 0, o_5^{suc}(1) = \infty, o_5^{val}(2) = 0.9 \cdot 1 + 0.1 \cdot 0.9,\ o_5^{suc}(2) = 0.694\} >$, and so on: suitable steps in the algorithms would similarly proceed until desired.

Figure 2 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
- 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,

Prepared using rncauth.cls
DOI: 10.1002/rnc

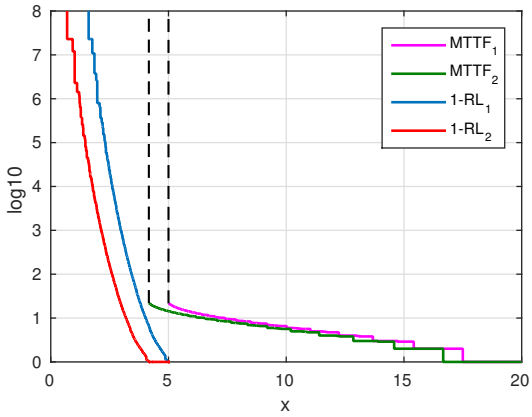


Figure 2. Example 1: $1 - RL(x, \theta)$ *vs.* MTTF for each mode, without disturbances ($\log_{10}$).

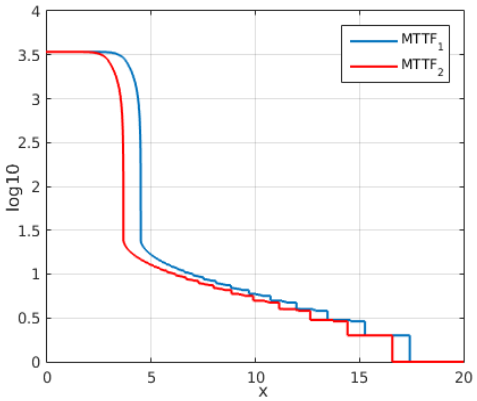


Figure 3. Example 2: Mean-time to failure bound ($\log_{10}$).

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 now the above system subject to disturbance, $x_{k+1} = 1.2x_k + u_k + w_k$ (mode 1), $x_{k+1} = 1.2x_k + w_k$ (mode 2), with $\mathbb{W} = \{|w| < 0.1\}$ and same $\mathbb{U}$, $\Omega^{[i]}$ and mode probabilities as above. In this case, RL is zero everywhere (a possibly unstable disturbed system yields empty robust invariant sets), and MTTF is shown in Figure 3 (logarithmic scale). At the moment of stopping the algorithm, O had 4300 elements.

For instance, MTTF bound changes sharply at $|x| \approx 4.5$ for mode 1 (note that if $|x| = 4.5$, under a worst-case disturbance, we will have $1.2|x| - 1 + 0.1 = 4.5$, so the bound corresponds to a sort of 'invariant under mode 1' set). Also, the MTTF bound is larger than 3414 samples for $|x| < 2.5$; last, in $|x| > 4.5$ for mode 1 and $|x| > 4.5/1.2$ for mode 2, failure is ver quick to come (estimated MTTF cannot be guaranteed to be larger than 24.15).

Example 3. Let us now consider the undisturbed second-order MJLS (1) with three operating modes, $i = \{1, 2, 3\}$ considered in [1]. The model matrices A_i, B_i and constraint regions are directly

DOI: 10.1002/rnc

taken from [1]. The transition probabilities are[¶] set to $\pi_{21} = \pi_{22} = \pi_{33} = 0.4$, and the rest of transition probabilities equal to 0.3.

Terminal sets $\mathbb{K}_\infty^{[i]}$ were first computed using the mode-dependent one-step controllable set (2) and, subsequently, Algorithm 2 in the Appendix was executed. The total number of sets when the algorithm converged was 139. Computation time (i7-4790K, Matlab® R2015a) was 26.84 seconds.

Figure 4a depicts the 'stationary' reliability bound $RL_s(x) := \sum_{i=1}^{3} p_i RL(x,i)$, where, from the eigenvectors of $\mathbb{P}$, the prior probability of each mode has been set to the stationary values $p_1 = 0.30, p_2 = 0.3667, p_3 = 0.3333$. As it is a disturbance-free case with finite-step algorithm convergence, results are optimal (Theorem 2). Figure 4b compares the result obtained in this work and those in [1]. This figure shows the increase in achieved reliability by Algorithm 2 with respect to the reliability obtained in [1], reaching an improvement of 0.25 in some states (for brevity, the equivalent plot of Figure 4a with the algorithm in [1] hasn't been included in this paper, showing only the difference).

As a last example, Figure 4c depicts the stationary reliability bound, considering a disturbed system, with $E_1 = E_2 = \begin{pmatrix} 1 & 0.5 \end{pmatrix}^T$ and $\mathbb{W} = \{|w| < 1.2\}$. Clearly, the resulting reliability bounds are lower, as the robust-invariant and one-step sets are smaller due to the (worst-case) disturbance effect. In fact, it is well known that invariant-set computations in disturbed systems do not in general converge in a finite number of iterations [4]; the same happens here: computation of terminal sets is now carried out seeding an initial $\mathbb{K}_0^* = \mathbb{W}^*$ in the iterations below Definition 2, so that any $\mathbb{K}_l^*$ such that $\mathbb{K}_l^* \subset \mathbb{K}_{l+1}^*$ is mode-dependent controllable and, hence, any of them can be used as terminal set for Algorithm 2. In this case $l = 41$, which, with a tolerance of 10^{-6} achieved the required inclusion, was used. Algorithm 2 was subsequently executed and the data in Figure 4c was obtained when it was stopped with O having 128 elements.

6. CONCLUSION

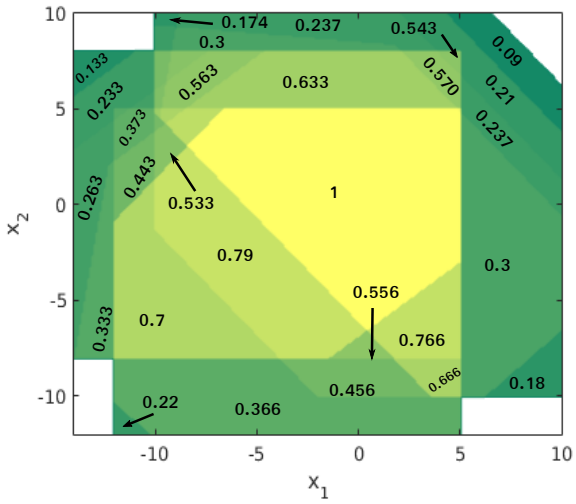
This work has extended prior results in [1] dealing with reliability bounds and associated mode-dependent control laws for Markov-Jump linear systems. Reliability is understood as the probability of avoiding constraint violations in future time; obviously, inside the maximal robust controllable set reliability 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.

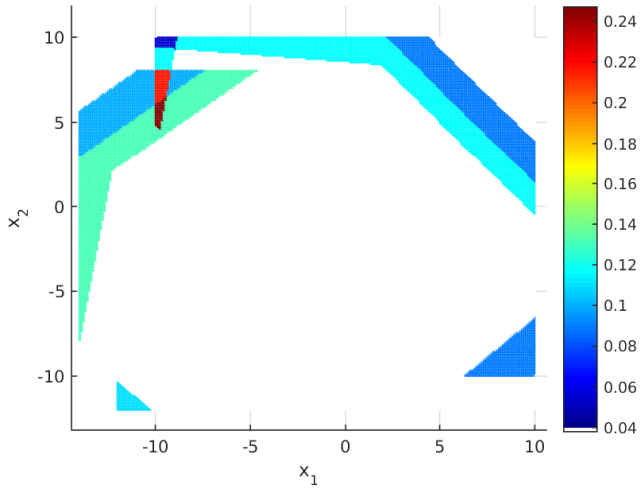
REFERENCES

[¶]This choice of probability matrix is done to emphasise the difference between the prior 'greedy' approach and the optimal controller here proposed; in [1], the probability of the most-probable sequence was, intentionally, set to a large value, in order to obtain reliable enough results by betting to a single outcome.

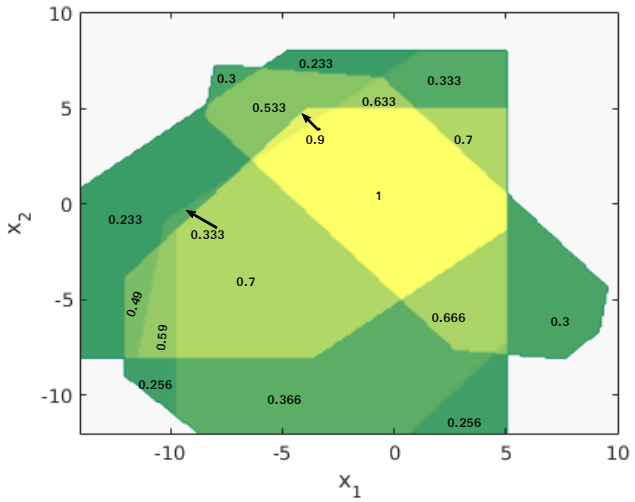
Prepared using rncauth.cls
DOI: 10.1002/rnc



(a) Optimal reliability RL for different initial states, under stationary mode assumption.



(b) Reliability increase with respect to greedy algorithm.



(c) Reliability bound with disturbances.

Figure 4. Reliability estimates for Example 3.

1. Hernández-Mejías M, Sala A, Arino C, Querol A. Reliable controllable sets for constrained Markov-jump linear systems. *International Journal of Robust and Nonlinear Control* 2016; **26**(10):2075–2089.
2. Mayne DQ. Model predictive control: Recent developments and future promise. *Automatica* 2014; **50**(12):2967–2986.
3. Blanchini F. Ultimate boundedness control for uncertain discrete-time systems via set-induced lyapunov functions. *Automatic Control, IEEE Transactions on* 1994; **39**(2):428–433.
4. Kerrigan E. Robust constraint satisfaction: Invariant sets and predictive control. PhD Thesis, PhD thesis, Cambridge 2000.
5. Herceg M, Kvasnica M, Jones CN, Morari M. Multi-parametric toolbox 3.0. *Control Conference (ECC), 2013 European*, IEEE, 2013; 502–510.
6. De Santis E, Di Benedetto M, Berardi L. Computation of maximal safe sets for switching systems. *Automatic Control, IEEE Transactions on* 2004; **49**(2):184–195.

DOI: 10.1002/rnc

7. Grieder P, Rakovic S, Morari M, Mayne D. Invariant sets for switched discrete time systems subject to bounded disturbances. *IFAC World Congress*, 2005.
8. Rakovic S, Grieder P, Kvasnica M, Mayne D, Morari M. Computation of invariant sets for piecewise affine discrete time systems subject to bounded disturbances. *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, vol. 2, IEEE, 2004; 1418–1423.
9. Lazar M, Heemels W, Weiland S, Bemporad A. Stabilizing model predictive control of hybrid systems. *Automatic Control, IEEE Transactions on* 2006; **51**(11):1813–1818.
10. Bukowski JV, Goble WM. Defining mean time-to-failure in a particular failure-state for multi-failure-state systems. *Reliability, IEEE Transactions on* 2001; **50**(2):221–228.
11. Zhang L, Shi Y, Chen T, Huang B. A new method for stabilization of networked control systems with random delays. *Automatic Control, IEEE Transactions on* 2005; **50**(8):1177–1181.
12. Blanke M, Schröder J. *Diagnosis and fault-tolerant control*, vol. 691. Springer, 2006.
13. Bernardini D, Bemporad A. Stabilizing model predictive control of stochastic constrained linear systems. *Automatic Control, IEEE Transactions on* 2012; **57**(6):1468–1480.
14. do V Costa O, Fragoso M, Marques R. *Discrete time Markov jump linear systems*. Springer, 2005.
15. Zhang L, Boukas E. Stability and stabilization of markovian jump linear systems with partly unknown transition probabilities. *Automatica* 2009; **45**(2):463–468.
16. Patrinos P, Sopasakis P, Sarimveis H, Bemporad A. Stochastic model predictive control for constrained discrete-time markovian switching systems. *Automatica* 2014; **50**(10):2504–2514.
17. Dehghan M, Ong C. Discrete-time switching linear system with constraints: Characterization and computation of invariant sets under dwell-time consideration. *Automatica* 2012; **48**(5):964–969.
18. Lu J, Li D, Xi Y. Constrained model predictive control synthesis for uncertain discrete-time markovian jump linear systems. *Control Theory & Applications, IET* 2013; **7**(5):707–719.
19. Kothare MV, Balakrishnan V, Morari M. Robust constrained model predictive control using linear matrix inequalities. *Automatica* 1996; **32**(10):1361–1379.

APPENDIX: TERMINAL-SET ALGORITHM AND IMPLEMENTATION GUIDELINES

Consider that a subset of the terminal set $S^* \subset \mathbb{K}^*_\infty$ is available. Then, as discussed in the main text, a modified algorithm can be crafted by replacing initialisation in step 1 of Algorithm 1 by new sets and value functions, (see initialisation of Algorithm 2 on page 21; actually, related to the terminal initialisation in the MTTF version). Also, as the definition of RT_l before Theorem 3 involve RT_l being the likelihood of reaching the terminal set in *at most* l steps, this means that expression (14) must be replaced by:

$$O^{[l]} := D^{[l]} \cup Y_h^{[l]} \cup O^{[l-1]} \tag{25}$$

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 [16], needing to resort to the so-called "scenario approach" [13]. Also, in high-dimensional systems, the projection

Prepared using rncauth.cls
DOI: 10.1002/rnc

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 [16].

Improved implementation.

The proposition below discusses how to update the value function when a invariant set has been obtained (in the geometric sense).

Proposition 3
Consider $o \in \mathcal{O}$, $q = Q_i(o)$ for any arbitrary $i \in \mathcal{M}$. If $o_{set} \subset q_{set}$, then the component o_f can be set to $\psi_\infty := \lim_{k\to\infty} \psi_k$, obtained from recursion $\psi_{k+1} := \max(\psi_k, Q_i^f(\psi_k, o_{set}))$, starting with $\psi_0 := o_f$. The element $\tilde{o}$, with $\tilde{o}_{set} := o_{set}$, $\tilde{o}_f := \psi_\infty$ verifies $o \blacktriangleleft \tilde{o}$. Also, defining $\tilde{q} := Q_i(\tilde{o})$, we have $q \blacktriangleleft \tilde{q}$.

Proof
Given ψ_0, then ψ_1 is the value/successor function of $r_1 = I(o, Q_i(o))$, where $r_{1,set} = o_{set}$. Then, ψ_2 is the one for $r_2 = I(r_1, Q_i(r_1))$, and so on again, it can be repeated until no improvement occurs. □

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 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., $Q_1(S) \supset S$, 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 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 (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 2 and 3): 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 $card(O) \times M$ will be changed from '`pending`' to '`done`' once $Q_j(O(\alpha))$ has been computed.
- Regarding I-operator, an upper triangular matrix T of dimensions $card(O) \times card(O)$ 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 2, 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:

DOI: 10.1002/rnc

Algorithm 2 Terminal-set based reliability bound

1. Let $O^{[0]}$ be a M-element set $\{o_1, \ldots, o_M\}$ such that:
   $(o_j)_{set} \subset \mathbb{K}_{\infty}^{[j]}$, $(o_j)_f^{val}(j) = 1$, $(o_j)_f^{suc}(j) \subset \mathbb{S}_j$ 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 $h = card(O)$.
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 $q = Q_j(O(\alpha))$,
   according to any criteria (either randomly or with probability-based choices to be later discussed).
4. Label $V(j, \alpha) =$'`done`' in case (b), or $T(\alpha, \beta) =$'`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 $h = h + 1$, 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 3 below.
9. If there are no pending operations, then STOP.
10. Go to step 3.

Algorithm 3 Updating matrix T.

| **Inputs**: α, β, T. | **Outputs**: modified T.

1. For each $r = 1, \ldots, \alpha$: if $T(r, \alpha) =$'`dom`', set $T(r, h) =$'`dom`'.
2. For each $r = 1, \ldots, \beta$: if $T(r, \beta) =$'`dom`', set $T(r, h) =$'`dom`'.
3. For each r, $1 \leq r \leq h$: if $T(r, \alpha)$ or $T(r, \beta)$ are '`same`' or '`done`' and $T(r, h+1) \neq$'`dom`', then set $T(r, h) =$'`same`'.
4. Set $T(\alpha, h) =$'`dom`' and $T(\beta, h) =$'`dom`'.

1. If $T(k, j) =$'`dom`', $k < j$, then $O(k) \succ O(j)$.

2. If $T(k, j) =$'`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).

Proof

Consider $O(h) = q = I(O(\alpha), O(\beta))$ from step 6 of Alg. 2. The first statement comes from the fact that $O(\alpha) \succ q$, $O(\beta) \succ q$ is encoded in the Step 4 (Alg. 3), and transitivity of the I-operator justifies carrying out steps 1 and 2 (Alg. 3), as the new element inherits subset-dominated sets from its parents.

Regarding the second statement, note that Step 4 (Alg. 2) will set $T(\alpha, \beta) =$'`done`'.

Consider now for arbitrary $r < h$, the operation $I(O(r), O(h)) = I(O(r), O(\alpha), O(\beta))$. Step 3 (Alg. 3) labels with '`same`' the element $T(r, h)$ if $T(r, \alpha)$ or $T(r, \beta)$ are '`same`' or '`done`' because:

Prepared using rncauth.cls
DOI: 10.1002/rnc

- If $T(r,\alpha) =$ '`same`', it means that $I(O(r),O(\alpha))$ can be obtained as the intersection of pre-existing elements[‖]. 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 3 set $T(r,h) =$'`same`'.
- If $T(r,\alpha) =$'`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. 3 sets $T(r,h) =$'`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 2, 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 5 and 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 of the probability-based scenario generation in, for instance, [13]. In this way, in non-converged cases the operations with better expected value function would be carried out first.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the financial support of Spanish MINECO (DPI2011-27845-C02-01, FPU12/02107) and Generalitat Valenciana (PrometeoII/2013/004).

[‖]We are implicitly using an induction argumentation: we are proving that if the proposition holds for prior list elements, so it will for subsequent additions.

DOI: 10.1002/rnc