

Document downloaded from:

<http://hdl.handle.net/10251/88030>

This paper must be cited as:

Leopoldo Armesto; Girbés, V.; Sala, A.; Miroslav Zima; Václav mídl (2015). Duality-Based Nonlinear Quadratic Control: Application to Mobile Robot Trajectory-Following. IEEE Transactions on Control Systems Technology. 23(4):1494-1504. doi:10.1109/TCST.2014.2377631.



The final publication is available at

<http://doi.org/10.1109/TCST.2014.2377631>

Copyright Institute of Electrical and Electronics Engineers (IEEE)

Additional Information

(c) 2015 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/ republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works.

# Duality-based Nonlinear Quadratic Control: Application to Mobile Robot Trajectory-following

Leopoldo Armesto, Vicent Girbés, Antonio Sala, Miroslav Zima, and Václav Šmídl

**Abstract**—This paper presents non-iterative linearization-based controllers for nonlinear unconstrained systems, coined as Extended Rauch-Tung-Striebel (ERTS) and Unscented Rauch-Tung-Striebel (URTS) controllers, derived from the duality between optimal control and estimation. The proposed controllers use a Rauch-Tung-Striebel forward-backward smoother as a state estimator in order to compute the original optimal control problem. The new controllers are applied to trajectory-following problems of differential-drive mobile robots and compared with iterative iLQR controller, nonlinear model predictive control and approximate inference approaches. Simulations show that ERTS and URTS controllers produce almost-optimal solutions with a significantly lower computing time, avoiding initialization issues in the other algorithms (in fact, they can be used to initialize them). The paper validates ERTS controller with an experiment of a Pioneer 3DX mobile robot.

## I. INTRODUCTION

Optimal control is widely used in control practice due to its advantages regarding the individual tuning of actuator amplitudes and control goals for each output, with well-known solutions for the linear case, both unconstrained (LQR) and constrained [1], [2]. However, it is limited to a narrow spectrum of applications because many systems in practice are inherently nonlinear. Nonlinear optimal control strategies are computationally more demanding, see [3]–[5] for some model-based approaches to handling it.

The goal of model-based optimal control is designing a stabilizing control while minimizing a given performance criterion, usually in a quadratic form, assuming a deterministic plant model is available. Closed-loop solutions can not be found analytically in a general nonlinear case since it involves obtaining the solution of the corresponding Hamilton Jacobi-Bellman equations [6]. One approach to avoid this problem is the iterative solution of a finite-horizon optimal control problem for a given state with a receding horizon implementation; control approaches using this strategy are referred to as model predictive control (MPC, [1]) and nonlinear model predictive control (NMPC, [5]). These approaches can deal with the unconstrained and constrained problems, where both states and control inputs must satisfy particular conditions. MPC is restricted to quadratic cost functions, linear systems and linear constraints, while NMPC can optimize non-quadratic

cost functions for nonlinear systems restricted to general nonlinear constraints. Another well-accepted solution for the nonlinear case is the iterative linear quadratic regulator (iLQR) [7], which linearizes the cost function at given (non-optimal) state and control input trajectories and then computes optimal increments based on the linearized model. The algorithm converges with appropriate jumps on the direction of control input gradient and Hessian based on Levenberg-Marquardt approach.

The main drawback of existing approaches is the computational cost required to provide optimal solutions. Since all-known methods are iterative, they may take a large number of iterations to converge to the optimal solution depending on the chosen algorithm's initialization parameters (i.e., initial non-optimal trajectory): the choice of linearization points (gradient computations) are key for obtaining an optimal control policy for nonlinear systems; if fact, the optimal choice would be linearising around the (not yet known) optimal trajectory.

Recent contributions propose solving the above deterministic optimal control problems by embedding them into a generic stochastic optimal control (SOC) framework [8]–[10]. Indeed, in [8], the well-known duality in the Riccati equations arising in the LQR and Kalman filter setups is extended to other nonlinear control cases by reformulating the problem in terms of the Kullback-Leibler divergence. In particular, Todorov proposes a forward-backward smoothing approach, different from the classic forward-only prediction in Kalman's observer. A related approach, denoted as Approximate Inference Control (AICO), appears in [9]. The author proposes a probability distribution over a binary variable, based on the exponential of the cost index, and solves the resulting SOC setting. In particular, he proves equivalence between the SOC problem and estimation of the marginal distribution of the state conditioned to the binary variable readings, the latter addressed via message-passing algorithm [11]. The resulting algorithm, when applied to nonlinear control problems, is iterative; it is non-iterative for the linear LQR case.

Based on these ideas, this paper proposes the use of non-iterative controllers based on the duality between optimal control and estimation under certain assumptions [8] for nonlinear systems, extending our preliminary conference version [12]. In this sense, this work proposes an Extended Rauch-Tung-Striebel (ERTS) controller for nonlinear systems, which is derived from the duality between optimal control and estimation as an extension of Todorov's work. The proposed controller uses non-linear extensions of the Rauch-Tung-Striebel (RTS) forward-backward smoother for the dual estimation problem for nonlinear systems. It is a two-pass technique that allows to compute linearization points based on the Extended Kalman Filter (EKF) in the first step and then to smooth such trajectory

---

L. Armesto and V. Girbés are with Instituto de Diseño y Fabricación at Universitat Politècnica de Valencia, Spain

A. Sala is with Instituto Universitario de Automática e Informática Industrial at Universitat Politècnica de Valencia, Spain

M. Zima is with Regional Innovation Centre for Electrical engineering, University of West Bohemia, Plzen, Czech Republic

V. Šmídl is with the Institute of Information Theory and Automation, Prague, Czech Republic

using a backward RTS smoother. The computed estimate of the next state is then used for the computation of the optimal control within a receding horizon policy. This results in an efficient controller with complexity  $\mathcal{O}(N^2)$  in state dimensions. The controller is optimal for LQ systems and the extension to non-linear settings is done by linearization along predicted trajectory where results are slightly suboptimal, as discussed later. An alternative to linearization at a single point as used in EKF is the Unscented Transform [13], which computes linearization at multiple points and thus further reduce linearization errors. Application of this transform to the smoothing problem yields the Unscented Rauch-Tung-Striebel (URTS) controller.

In this paper, motivated by the wide range of applications of the trajectory-following problem [14]–[18], a comparative study of performance and computational cost has been carried out in such specific application: despite of the fact that ERTS (and URTS) use a non-iterative algorithm, their solution is only around 25% (and 24%) worse than the best solution, respectively, with lower execution time in the case of ERTS. They can, too, be used as seed for iterative algorithms, such as iLQR.

The paper is organized as follows: Preliminaries can be found in Section II, section III states the problem and section IV focuses on the duality between optimal control and estimation and particularizes the ideas from previous sections to Gaussian distributions and presents ERTS and URTS controllers. The controllers are analysed in Section V and compared against iLQR, NMPC and AICO algorithms. In addition to this, we show some experimental results for the ERTS trajectory following problem using a Pioneer 3DX mobile robot platform in Section V. Conclusions are drawn in Section VI.

## II. PROBLEM DEFINITION

We are concerned with the problem of optimal control of a stochastic nonlinear dynamic system with state  $\mathbf{x}_t$  and the deterministic control action  $\mathbf{u}_t$  with known model:

$$\mathbf{x}_{t+1} \sim p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t) \quad (1)$$

We seek an optimal control policy  $\mathbf{u}_t = \pi_t(\mathbf{x}_t, \mathbf{s}_{t:N}) \triangleq \pi_t$  which optimizes the expected value of the following additive cost function:

$$L(\mathbf{x}_0, \mathbf{s}_{0:N}, \mathbf{u}_{0:N-1}) = q_N(\mathbf{x}_N, \mathbf{s}_N) + \sum_{t=0}^{N-1} l_t(\mathbf{x}_t, \mathbf{s}_t, \mathbf{u}_t) \quad (2)$$

where sequence  $\mathbf{s}_{0:N} \triangleq \mathbf{s}_0, \dots, \mathbf{s}_N$  stands for the desired quantities related to states (actually, reference trajectories for some outputs),  $q_N(\mathbf{x}_N, \mathbf{s}_N)$  is an arbitrary function for the final cost and the intermediate cost is  $l_t(\mathbf{x}_t, \mathbf{s}_t, \mathbf{u}_t)$ . Design of the control policy is a classical problem of dynamic programming

$$J_t(\mathbf{x}_t, \mathbf{s}_{t:N}) = \min_{\pi_t} \mathbb{E} \{ l_t(\mathbf{x}_t, \mathbf{s}_t, \mathbf{u}_t) + J_{t+1}(\mathbf{x}_{t+1}, \mathbf{s}_{t+1:N}) \} \quad (3)$$

where  $\mathbb{E}$  is the expected value over distribution of  $\mathbf{x}_{t+1}$  and  $J_t$  is the Bellman function. Analytical solution of (3) exists only

for limited cases such as the linear quadratic systems. For non-linear systems, the evaluation has to be approximated. Here, we focus on reformulation of this problem into a dual problem of stochastic smoothing [8] and its relations with alternative approaches.

### A. Duality of estimation and control

As demonstrated in [8], the optimization problem (3) can be translated into the language of probability calculus and solved as an optimal smoothing problem. This is possible if there exist probability distributions  $\bar{p}(\mathbf{x}_{t+1}|\mathbf{x}_t)$ ,  $p(\mathbf{s}_t|\mathbf{x}_t)$  such that the cost index can be written as

$$l_t(\mathbf{x}_t, \mathbf{s}_t, \mathbf{u}_t) = q_t(\mathbf{x}_t, \mathbf{s}_t) + \text{KL}(p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t) || \bar{p}(\mathbf{x}_{t+1}|\mathbf{x}_t)) \quad (4)$$

where KL is the Kullback Leibler divergence [19]

$$\begin{aligned} \text{KL}(p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t) || \bar{p}(\mathbf{x}_{t+1}|\mathbf{x}_t)) &= \\ &= \int_{\mathbb{R}^{n_x}} \log \left[ \frac{p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t)}{\bar{p}(\mathbf{x}_{t+1}|\mathbf{x}_t)} \right] p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t) d\mathbf{x}_{t+1} \end{aligned} \quad (5)$$

and  $\bar{p}(\mathbf{x}_{t+1}|\mathbf{x}_t)$  is a *reference dynamics*, and

$$p(\mathbf{s}_t|\mathbf{x}_t) \propto e^{-q_t(\mathbf{x}_t, \mathbf{s}_t)} \quad (6)$$

Under these assumptions, it can be proved that the optimal control problem (3) is dual to the problem of marginal prediction of the state [8], [12]; with general solution of the Bellman function

$$J_{t+1}(\mathbf{x}_{t+1}, \mathbf{s}_{t+1:N}) = -\log p(\mathbf{s}_{t+1:N}|\mathbf{x}_{t+1}) + \zeta_{t+1}, \quad (7)$$

where  $\zeta_{t+1}$  is a constant (i.e., independent of  $\mathbf{x}_t$ ) and

$$p(\mathbf{s}_{t+1:N}|\mathbf{x}_{t+1}) = \int_{\mathbb{R}^{n_x}} p(\mathbf{x}_{t+2:N}, \mathbf{s}_{t+1:N}|\mathbf{x}_{t+1}) d\mathbf{x}_{t+2:N} \quad (8)$$

is the marginal predictive model, which can be computed recursively using equation (33).

The optimal policy is then defined as a minimizer of :

$$\pi_t^* = \arg \min_{\pi_t} \text{KL}((p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t) || p_{\pi}(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{s}_{t+1:N}))) \quad (9)$$

$$p_{\pi}(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{s}_{t+1:N}) \propto e^{-J_{t+1}(\mathbf{x}_{t+1}, \mathbf{s}_{t+1:N})} \bar{p}(\mathbf{x}_{t+1}|\mathbf{x}_t) \quad (10)$$

under the assumption that such a minimizer exists and the minimum of (9) is independent of  $\mathbf{x}_t$ . Here,  $\propto$  denotes equality up to a multiplicative constant. We note that after substitution of (7) into (10) the result is the Bayes rule and thus a definition of the optimum smoothing problem. This problem is commonly studied and many methods has been developed for its solution.

The duality can be used to establish duality between the classical LQR design and the Rauch-Tung-Striebel (RTS) smoother [20] for linear model (1) with quadratic cost. Appendices A and B detail the main developments in order to establish such duality. Appendix C derives LQR controller form KL cost for linear-time variant (LTV) systems.

## B. Iterative linearisation-based optimal control

As nonlinear optimal control problems are hard problems, linearisation around a trajectory is often used, so the systems appear as:

$$\delta \mathbf{x}_{t+1} = \mathbf{A}_t(\mathbf{x}_t)\delta \mathbf{x}_t + \mathbf{B}_t(\mathbf{x}_t)\delta \mathbf{u}_t \quad (11)$$

where  $\delta \mathbf{x}_t$ ,  $\delta \mathbf{u}_t$  are increments over a chosen linearisation trajectory.

The basic problem of linearisation approaches is, of course, choosing the points in which to linearise the system. Actually, it is clear that the optimal linearisation points would be those of the optimal trajectory; however, as they depend on the to-be-computed control and the control depends on such points, iterative algorithms are needed (linearise around first trajectory estimate, compute control, compute new trajectory, repeat). Current iterative linearisation-based options in literature for finding solutions to nonlinear optimal control problems are:

a) *Iterative LQR (iLQR)*: If the result of the standard LQR algorithm is used to compute a new optimal increment of control action (and the ensuing state trajectory), the system can be re-linearised around the new trajectory and this can be iterated until convergence. This is the basis of the iLQR approach proposed in [7], and of a generalisation involving linearisation of nonlinear (but affine in noise) models coined as iLQG in [21].

b) *Nonlinear Model Predictive control (NMPC)*: Nonlinear model-predictive control minimises a finite-time quadratic cost based on open-loop predictions using standard nonlinear optimization algorithms. Most NMPC implementations compute gradients and Hessians of the quadratic cost function which, ultimately, depend on the model gradient (i.e., linearisation) at a candidate solution to be iteratively improved. It can deal with non-linear cost functions and state and control input constraints. The reader is referred, for instance, to [5] for ample information on the topic.

c) *Approximate Inference Control (AICO)*: The approach in [9] proposes an artificial binary random variable  $\mathbf{z}_t$  such that  $P(\mathbf{z}_t = 1 | \mathbf{x}_t, \mathbf{u}_t) = e^{-t(\mathbf{x}_t, \mathbf{s}_t, \mathbf{u}_t)}$ . In this way, the control problem can be posed as a determining a trajectory which maximizes the probability of  $\mathbf{z}_t = 1$  for all times, transformed to an estimation problem via Bayes' rule. The proposed solution to the problem uses the message-passing algorithm [11], and in the LQR case it needs a single forward-backward pass, actually being a message-passing implementation of the RTS smoother, instead of the original Riccati one. For nonlinear systems, AICO uses an iterative approach to compute linearization points (with iterations of the message-passing algorithm which, themselves, include nested iterations to find suitable linearization points for Gaussian belief propagation).

## C. Problem statement

The objective of this work is to investigate efficiency of the existing tools for non-linear stochastic smoothing to the problem of optimal control via duality. We will focus on smoothing approaches with Gaussian posterior, namely the

Extended Rauch-Tung-Striebel (RTS) smoother and the Unscented RTS [22]. Both smoothers are analogous to the non-linear extensions of the Kalman filter, the Extended Kalman filter [23] and the Unscented Kalman filter [13], respectively.

The distinction from the alternative linearization approaches is that the ERTS linearise the trajectory using not only the system dynamic model, but also the covariance matrices representing penalizations of the cost function. The resulting algorithm use only two passes of the horizon without any further iterations.

A preliminary conference version of this work appears in [12]. In the present paper, more detailed account of the root ideas is provided, and a deeper analysis and discussion with alternative options in literature is carried out. Given the upfront good accuracy of the non-iterative ERTS algorithm, its viability as an initialiser of iLQR setups will be also discussed in next sections.

Implementation details and experimental results with a Pioneer 3DX mobile robot using ROS (Robotic Operating System) [24] are also provided within the context of trajectory-following of a closed trajectory.

## III. NON-LINEAR QUADRATIC CONTROL VIA ESTIMATION

Consider a nonlinear model with affine control input and Gaussian disturbance

$$p(\mathbf{x}_{t+1} | \mathbf{x}_t, \mathbf{u}_t) \triangleq \mathcal{N}(\mathbf{f}(\mathbf{x}_t) + \mathbf{B}_t(\mathbf{x}_t)\mathbf{u}_t, \mathbf{V}_t) \quad (12)$$

for a known vector-valued function  $\mathbf{f}(\mathbf{x}_t)$ , matrix  $\mathbf{B}_t(\mathbf{x}_t) \triangleq \mathbf{B}_t$ . Consider a quadratic cost (2):

$$L_t = \frac{1}{2} \mathbf{e}_N^T \mathbf{Q}_N \mathbf{e}_N + \frac{1}{2} \sum_{t=0}^{N-1} (\mathbf{e}_t^T \mathbf{Q}_t \mathbf{e}_t + \mathbf{u}_t^T \mathbf{R}_t \mathbf{u}_t) \quad (13)$$

for  $\mathbf{e}_t = \mathbf{s}_t - \mathbf{h}(\mathbf{x}_t)$  and known  $\mathbf{Q}_t$ ,  $\mathbf{R}_t$  and known vector-valued function  $\mathbf{h}(\mathbf{x}_t)$ .

Dual formulation of quadratic cost (13) in the sense of Section II.A is established for the following distributions:

$$\bar{p}(\mathbf{x}_{t+1} | \mathbf{x}_t) \triangleq \mathcal{N}(\mathbf{f}(\mathbf{x}_t), \bar{\mathbf{V}}_t), \bar{\mathbf{V}}_t = \mathbf{B}_t \mathbf{R}_t^{-1} \mathbf{B}_t^T \quad (14)$$

$$p(\mathbf{s}_t | \mathbf{x}_t) \triangleq \mathcal{N}(\mathbf{h}(\mathbf{x}_t), \mathbf{Q}_t^{-1}) \quad (15)$$

The result of substitution of these distributions into (4) via (6), is the quadratic cost (13) plus constant terms without any influence on the optimal control policy.

### A. Solution of the dual problem

Distributions (14) and (15) form a definition of the problem of stochastic filtering, where the former is the model of system dynamics, and the latter is known as the observation model. Note that in the dual formulation, the requested values  $\mathbf{s}_t$  act as observations.

The first task is to find solution of the integral (8). Just like the optimal control problem (3), solution of this equation is analytically tractable for linear systems (i.e.  $\mathbf{f}(\mathbf{x}_t) = \mathbf{A}_t \mathbf{x}_t$ )

with Gaussian noise via algorithm known as the Rauch-Tung-Striebel (RTS) smoother, [20]. The resulting distribution is Gaussian

$$p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{s}_{t+1:N}) = \mathcal{N}(\hat{\mathbf{x}}_{t+1|N}, \mathbf{P}_{t+1|N})$$

which corresponds to quadratic Bellman function (7).

The RTS smoother uses the following steps, given a known  $\mathbf{x}_t$  by assumption:

- forward pass of Kalman filter computing  $p(\mathbf{x}_\tau|\mathbf{x}_t, \mathbf{s}_{t:\tau})$  for  $\tau = t, \dots, N$ .
- backward pass computing the  $p(\mathbf{x}_\tau|\mathbf{x}_t, \mathbf{s}_{t:N})$  for  $\tau = N - 1, \dots, t + 1$ .

### B. Rauch-Tung-Striebel Nonlinear Controller

If functions  $\mathbf{f}(\mathbf{x}_t)$ ,  $\mathbf{h}(\mathbf{x}_t)$  are nonlinear, the RTS can be used for the linearized model at each trajectory point; however, optimality of the proposed estimate is no longer guaranteed. Notwithstanding, this is analog to the Extended Kalman filter, successfully used in many control and robotics applications.

Inspired on that success, we propose to approximate the non-linear dual problem by nonlinear extensions of the RTS smoother, such as the Extended RTS (ERTS) [25] which is based on linearization at the point of the expected value. It is well known that Unscented Kalman Filter and smoothers [22] provides more accurate results than EKF, by propagating lower errors, due to linearization at multiple points given by the Unscented Transform. The Unscented RTS version (URTS) also requires a forward-backward pass and thus, assumptions done for ERTS will be also extended to URTS. Simulations and experiments later in this paper will show that, indeed, good performance in practice can be achieved using the duality via ERTS or URTS framework in the algorithms to be discussed next. In next section, a comparative analysis of accuracy and computational time will be discussed in the context of a robotic application example.

A receding horizon implementation of the above ERTS/URTS control laws will be pursued; thus, at any arbitrary instant  $t$ , with known state  $\mathbf{x}_t$ , the needed estimate distribution will be a Gaussian distribution

$$p_\pi = p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{s}_{t+1:t+N}) = \mathcal{N}(\hat{\mathbf{x}}_{t+1|t+N}, \mathbf{P}_{t+1|t+N}) \quad (16)$$

with mean value  $\hat{\mathbf{x}}_{t+1|t+N}$  and covariance matrix  $\mathbf{P}_{t+1|t+N}$ ,

1) *Computation of nonlinear control law:* Once the estimation problem has been solved by either of the two above proposals, duality indicates that the optimal control action should fulfill the implicit equation (9). The Gaussian approximation of the smoother (16) defines the second argument of the KL cost (9) yielding the implicit function to be defined as a minimizer of quadratic loss  $\text{KL}(p||p_\pi) = \mathbf{e}_t^T \mathbf{P}_{t+1|t+N}^{-1} \mathbf{e}_t$  plus constants with:

$$\mathbf{e}_t = \mathbf{f}(\mathbf{x}_t) + \mathbf{B}_t \mathbf{u}_t - \hat{\mathbf{x}}_{t+1|t+N} \quad (17)$$

The minimizer is found to be

$$\mathbf{u}_t = (\mathbf{B}_t^T \mathbf{P}_{t+1|t+N}^{-1} \mathbf{B}_t)^{-1} \mathbf{B}_t^T \mathbf{P}_{t+1|t+N}^{-1} (\hat{\mathbf{x}}_{t+1|t+N} - \mathbf{f}(\mathbf{x}_t)) \quad (18)$$

using standard minimization of quadratic functions.

*Remark 3.1 (Computational simplification):* We note that for invertible matrices  $\mathbf{B}_t$ ,  $\mathbf{e}_t$  can be trivially made zero and hence, the variance of the smoother is irrelevant and does not need to be evaluated.

This is also the case for LTV systems or when using linearisation-based approaches such as ERTS, because the smoother with perfect knowledge of  $\mathbf{x}_t$  provides an estimation such that  $\mathbf{f}(\mathbf{x}_t) - \hat{\mathbf{x}}_{t+1|t+N}$  always lies in the space of  $\mathbf{B}_t \mathbf{u}_t$ , so (17) can always be solved for  $\mathbf{u}_t$ , hence (18) gives the same result than

$$\mathbf{u}_t = (\mathbf{B}_t^T \mathbf{B}_t)^{-1} \mathbf{B}_t^T (\hat{\mathbf{x}}_{t+1|t+N} - \mathbf{f}(\mathbf{x}_t)) \quad (19)$$

2) *ERTS control algorithm:* From the above considerations, the resulting control algorithm (Algorithm 1), denoted as Extended Rauch-Tung-Striebel (ERTS) controller, is composed of two parts: 1) computing  $\hat{\mathbf{x}}_{t+1|t+N}$  via the ERTS smoother, and 2) obtaining the approximation of the optimal control (19). The state  $\mathbf{x}_t$  is assumed to be known, so the proposed controller is a deterministic state feedback one.

---

#### Algorithm 1 ERTS Controller

---

```

[Initialization]
1:  $\hat{\mathbf{x}}_{t|t} = \mathbf{x}_t, \mathbf{P}_{t|t} = \mathbf{0}$ 
[Prediction]
2: for  $\tau = t + 1, \dots, t + N$  do
3:    $\hat{\mathbf{x}}_{\tau|\tau-1} = \mathbf{f}(\hat{\mathbf{x}}_{\tau-1|\tau-1})$ 
[linearization]
4:    $\mathbf{A}_{\tau-1} = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}_{\tau-1}} \right|_{\mathbf{x}_{\tau-1} = \hat{\mathbf{x}}_{\tau-1|\tau-1}}$ 
5:    $\mathbf{H}_\tau = \left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}_\tau} \right|_{\mathbf{x}_\tau = \hat{\mathbf{x}}_{\tau|\tau-1}}$ 
6:    $\mathbf{P}_{\tau|\tau-1} = \mathbf{A}_{\tau-1} \mathbf{P}_{\tau-1|\tau-1} \mathbf{A}_{\tau-1}^T + \mathbf{B}_{\tau-1} \mathbf{R}_{\tau-1}^{-1} \mathbf{B}_{\tau-1}^T$ 
7:    $\mathbf{K}_\tau = \mathbf{P}_{\tau|\tau-1} \mathbf{H}_\tau^T (\mathbf{H}_\tau \mathbf{P}_{\tau|\tau-1} \mathbf{H}_\tau^T + \mathbf{Q}_\tau^{-1})^{-1}$ 
8:    $\mathbf{P}_{\tau|\tau} = (\mathbf{I} - \mathbf{K}_\tau \mathbf{H}_\tau) \mathbf{P}_{\tau|\tau-1}$ 
9:    $\hat{\mathbf{x}}_{\tau|\tau} = \hat{\mathbf{x}}_{\tau|\tau-1} + \mathbf{K}_\tau (\mathbf{s}_\tau - \mathbf{h}(\hat{\mathbf{x}}_{\tau|\tau-1}))$ 
10: end for
[Smoothing]
11: for  $\tau = t + N - 1, t + N - 2, \dots, t + 1$  do
12:    $\mathbf{L}_\tau = \mathbf{P}_{\tau|\tau} \mathbf{A}_\tau^T \mathbf{P}_{\tau+1|\tau}^{-1}$ 
13:    $\hat{\mathbf{x}}_{\tau|t+N} = \hat{\mathbf{x}}_{\tau|\tau} + \mathbf{L}_\tau (\hat{\mathbf{x}}_{\tau+1|t+N} - \hat{\mathbf{x}}_{\tau+1|\tau})$ 
14: end for
[Control computation]
15:  $\mathbf{u}_t = (\mathbf{B}_t^T \mathbf{B}_t)^{-1} \mathbf{B}_t^T (\hat{\mathbf{x}}_{t+1|t+N} - \mathbf{f}(\mathbf{x}_t))$ 

```

---

Due to simplification (19) evaluation of  $\mathbf{P}_{\tau|t+N}$  can be omitted in the smoothing algorithm. Also, note that in order to provide a valid solution for the smoothing algorithm, matrix  $\mathbf{P}_{\tau+1|\tau}$  must be full-rank, otherwise, its Moore-Penrose pseudo-inverse should be used in line 12. Indeed, the pseudo-inverse would provide zero correction in line 13 of the algorithm in the state directions in which the reference dynamics noise does have zero variance. From a duality point of view, that amounts to requiring zero control action effect on uncontrollable states, as intuitively expected.

3) *URTS control algorithm:* Algorithm 2 implements the Unscented RTS (URTS) controller as an alternative to the ERTS one. The steps are the same with the difference that it uses the Unscented Transform (UT) on lines 5, 8 and 16 to compute the mean and covariance. The UT accepts the

mean and covariance of a variable  $\mathbf{x}$  and returns the mean and covariance of  $\mathbf{s}$  and (optionally as third argument) cross-covariance between  $\mathbf{x}$  and  $\mathbf{s}$  passed through a given function  $\mathbf{s} = \mathbf{f}(\mathbf{x})$ , where  $\alpha$ ,  $\beta$  and  $\kappa$  are well-known parameters used to spread sigma-points [22].

---

**Algorithm 2** URTS Controller

---

```

[Initialization]
1:  $\hat{\mathbf{x}}_{t|t} = \mathbf{x}_t$ ,  $\mathbf{P}_{t|t} = \mathbf{0}$ ,  $\alpha = 1$ ,  $\beta = 0$ ,  $\kappa = 3 - \dim(\mathbf{x}_t)$ 
[Prediction]
2: for  $\tau = t + 1, \dots, t + N$  do
  [Augmented State (system noise)]
3:  $\mathbf{x}_{\tau-1|\tau-1}^a = [\hat{\mathbf{x}}_{\tau-1|\tau-1}^T \mathbf{0}^T]^T$ 
4:  $\mathbf{P}_{\tau-1|\tau-1}^a = \text{block\_diag}(\mathbf{P}_{\tau-1|\tau-1}, \mathbf{B}_{\tau-1} \mathbf{R}_{\tau-1}^{-1} \mathbf{B}_{\tau-1}^T)$ 
  [Unscented Transform & Predict]
5:  $[\hat{\mathbf{x}}_{\tau|\tau-1}, \mathbf{P}_{\tau|\tau-1}] = \text{UT}(\mathbf{x}_{\tau-1|\tau-1}^a, \mathbf{P}_{\tau-1|\tau-1}^a, \mathbf{f}, \alpha, \beta, \kappa)$ 
  [Augmented State (measurement noise)]
6:  $\mathbf{x}_{\tau|\tau-1}^a = [\hat{\mathbf{x}}_{\tau|\tau-1}^T \mathbf{0}^T]^T$ 
7:  $\mathbf{P}_{\tau|\tau-1}^a = \text{block\_diag}(\mathbf{P}_{\tau|\tau-1}, \mathbf{Q}_{\tau}^{-1})$ 
  [Unscented Transform & Update]
8:  $[\hat{\mathbf{s}}_{\tau|\tau-1}, \mathbf{P}_{\tau|\tau-1}^{ss}, \mathbf{P}_{\tau|\tau-1}^{xs}] = \text{UT}(\mathbf{x}_{\tau|\tau-1}^a, \mathbf{P}_{\tau|\tau-1}^a, \mathbf{h}, \alpha, \beta, \kappa)$ 
9:  $\mathbf{K}_{\tau} = \mathbf{P}_{\tau|\tau-1}^{xs} (\mathbf{P}_{\tau|\tau-1}^{ss})^{-1}$ 
10:  $\mathbf{P}_{\tau|\tau} = \mathbf{P}_{\tau|\tau-1} - \mathbf{K}_{\tau} \mathbf{P}_{\tau|\tau-1}^{ss} \mathbf{K}_{\tau}^T$ 
11:  $\hat{\mathbf{x}}_{\tau|\tau} = \hat{\mathbf{x}}_{\tau|\tau-1} + \mathbf{K}_{\tau} (\mathbf{s}_{\tau} - \hat{\mathbf{s}}_{\tau|\tau-1})$ 
12: end for
[Smoothing]
13: for  $\tau = t + N - 1, t + N - 2, \dots, t + 1$  do
  [Augmented State (system noise)]
14:  $\mathbf{x}_{\tau|\tau}^a = [\hat{\mathbf{x}}_{\tau|\tau}^T \mathbf{0}^T]^T$ 
15:  $\mathbf{P}_{\tau|\tau}^a = \text{block\_diag}(\mathbf{P}_{\tau|\tau}, \mathbf{B}_{\tau} \mathbf{R}_{\tau}^{-1} \mathbf{B}_{\tau}^T)$ 
  [Unscented Transform & Smooth]
16:  $[\hat{\mathbf{x}}_{\tau+1|\tau}, \mathbf{P}_{\tau+1|\tau}, \mathbf{C}_{\tau+1|\tau}] = \text{UT}(\mathbf{x}_{\tau|\tau}^a, \mathbf{P}_{\tau|\tau}^a, \mathbf{f}, \alpha, \beta, \kappa)$ 
17:  $\mathbf{L}_{\tau} = \mathbf{C}_{\tau+1|\tau} \mathbf{P}_{\tau+1|\tau}^{-1}$ 
18:  $\hat{\mathbf{x}}_{\tau|t+N} = \hat{\mathbf{x}}_{\tau|\tau} + \mathbf{L}_{\tau} (\hat{\mathbf{x}}_{\tau+1|t+N} - \hat{\mathbf{x}}_{\tau+1|\tau})$ 
19:  $\mathbf{P}_{\tau+1|t+N} = \mathbf{P}_{\tau|\tau} + \mathbf{L}_{\tau} (\mathbf{P}_{\tau+1|t+N} - \mathbf{P}_{\tau+1|\tau}) \mathbf{L}_{\tau}^T$ 
20: end for
[Control computation]
21:  $\mathbf{u}_t = (\mathbf{B}_t^T \mathbf{P}_{t+1|t+N}^{-1} \mathbf{B}_t)^{-1} \mathbf{B}_t^T \mathbf{P}_{t+1|t+N}^{-1} (\hat{\mathbf{x}}_{t+1|t+N} - \mathbf{f}(\mathbf{x}_t))$ 

```

---

#### IV. APPLICATION EXAMPLE: TRAJECTORY TRACKING IN WHEELED ROBOTS

In this section, we analyse and discuss the benefits of the proposed algorithms compared to iLQR, NMPC and AICO. The study carried on is based on the application to kinematic control of non-holonomic wheeled mobile robot [26].

A vehicle state  $\mathbf{x}_t = (x_t, y_t, \theta_t, v_t, \omega_t)^T$  in time instant  $t$  is characterized by coordinates  $(x_t, y_t)$ , orientation  $(\theta_t)$ , linear velocity  $(v_t)$  and angular velocity  $(\omega_t)$  and it evolves through input  $\mathbf{u}_t = (a_t, \alpha_t)^T$  given by linear acceleration  $(a_t)$  and angular acceleration  $(\omega_t)$  as

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t) + \mathbf{B}(\mathbf{x}_t) \mathbf{u}_t \quad (20)$$

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \\ v_{t+1} \\ \omega_{t+1} \end{bmatrix} = \begin{bmatrix} x_t + v_t \Delta t \cos \theta_t \\ y_t + v_t \Delta t \sin \theta_t \\ \theta_t + \omega_t \Delta t \\ v_t \\ \omega_t \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \begin{bmatrix} a_t \\ \alpha_t \end{bmatrix} \quad (21)$$

The transition matrix of linearized system is:

$$\mathbf{A}(\mathbf{x}_t) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}_t} = \begin{pmatrix} 1 & 0 & -v_t \Delta t \sin \theta_t & \Delta t \cos \theta_t & 0 \\ 0 & 1 & v_t \Delta t \cos \theta_t & \Delta t \sin \theta_t & 0 \\ 0 & 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

being  $\Delta t = 0.05s$  the simulation step (Euler integration).

Let us discuss the detailed setting of each of the compared control strategies, all of them geared towards minimising the quadratic cost (13) with  $\mathbf{h}_t(\mathbf{x}_t) = \mathbf{x}_t$ . The aim is to drive a vehicle around a desired trajectory  $\mathbf{s}_{0:N}$ . The trajectory is given by a reference speed of  $\bar{v}_t = 0.5ms^{-1}$ , the reference positions,  $\bar{x}_t = \bar{v}_t t$ ,  $\bar{y}_t = 0m$ , orientation  $\bar{\theta}_t = 0rad$  and angular velocity  $\bar{\omega}_t = 0rads^{-1}$ , with

$$\mathbf{s}_t = (\bar{x}_t \ \bar{y}_t \ \bar{\theta}_t \ \bar{v}_t \ \bar{\omega}_t)^T \quad (22)$$

Penalization matrices have been set to  $\mathbf{Q}_t = \text{diag}(25, 25, 1, 1, 1)$  and  $\mathbf{R}_t = \text{diag}(0.5, 1)$ . The tracking problem will be set up with a horizon  $N = 60$ , assuming that the whole the trajectory is known in advance. Note that, intentionally, due to the weights in absolute speeds (4th and 5th states) and accelerations (input variables) the optimal trajectory will have some tracking position error (small, as the position error weight is high). A comparative between ERTS and iLQR in a case with abrupt reference changes can be found in [12].

The following strategies will be compared:

- 1) *Iterative Linear Quadratic controller (iLQR)*. This algorithm initializes with  $\mathbf{u}_{0:N-1} = \mathbf{0}$  and uses the open-loop trajectory  $\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t) + \mathbf{B}(\mathbf{x}_t) \mathbf{u}_t$  with  $\mathbf{x}_0$  as initial linearization point. This algorithm computes on every iteration a variation of control input  $\delta \mathbf{u}_t$  based on a Hessian approximation with Levenberg-Marquardt (LM) method and re-linearizes the process at the new trajectory as indicated in [21] (and implemented in [27]).
- 2) *ERTS approach*, in two settings:
  - Non-iterative (standalone) ERTS. Algorithm 1.
  - *Iterative Linear Quadratic controller (iLQR) initialized with ERTS solution, denoted as ERTS+*. Obviously, the cost for iteration 1 equals the non-iterative ERTS cost.
- 3) *URTS approach*. Similar to ERTS approach but using Algorithm 2 instead, in the two above settings. The iterative version is denoted as URTS+.
- 4) *Non-linear Model Predictive Control*. This algorithm performs the open-loop optimization of performance index using non-linear model (of course, using iterations in order to compute gradients and Hessians at the optimal points). Initialization  $\mathbf{u}_{0:N-1} = \mathbf{0}$  is used (implementation based on [5]).
- 5) *Approximate Inference Control (AICO)*. The iterative forward-backward message passing algorithm implementation is Algorithm 2 in [9]. AICO tuning parameters are set to  $\alpha = 0.9$  and  $\theta = 0.1$ , as used in [9].

### A. Analysis setup

A set of  $L = 100$  simulations have been considered with a uniformly distributed random initialization over robot state  $\mathbf{x}_0^i \sim \mathcal{U}(0, [-1, 1], [-\frac{\pi}{2}, \frac{\pi}{2}], [-0.5, 0.5], [-0.5, 0.5])$ , for  $i = 1, \dots, L$ . For each experiment  $i$  and optimizer iteration  $k$ , the achieved performance for each of the five strategies will be denoted as  $J_s(i, k)$ , for  $s = 1, \dots, 5$ .

On each iteration  $k$ , the method with the best cost is taken as reference  $J_{best}(i, k)$  to compare against other methods, i.e.,  $J_{best}(i, k) = \min_{1 \leq s \leq 5} (J_s(i, k))$ ,

A cost-ratio for each experiment and iteration is defined as:

$$R_s(i, k) = \frac{J_s(i, k)}{J_{best}(i, k)}, \quad s = 1, \dots, 5 \quad (23)$$

Based on the definition of the cost-ratio, some useful metrics will be also defined:

- Mean cost-ratio (MCR) per iteration, i.e., average of cost ratio over all simulations:  $MCR_s(k) = \frac{1}{L} \sum_{i=1}^L R_s(i, k)$ .
- Worst cost-ratio (WCR) per iteration:  $WCR_s(k) = \max_{1 \leq i \leq L} (R_s(i, k))$ .

In order to suitably compare the five algorithms, some unifying conditions must be also set up:

- Note that ERTS, URTS and AICO produce an estimated optimal state trajectory whereas iLQR and NMPC produce a whole batch of  $N$  future inputs. In order to compare the finite-horizon cost estimates from a particular initial condition  $\mathbf{x}_0$ , trajectory  $\{\hat{\mathbf{x}}_{1|N}, \dots, \hat{\mathbf{x}}_{N|N}\}$  is computed for strategies #2, #3 and #5, and future inputs  $\mathbf{u}_{0:N-1}$  for the #1, #4 ones. Then, the optimal smoothed state trajectory (#2, #3, #5) is taken as “reference” to compute control inputs  $\mathbf{u}_{0:N-1}$  via the Least-Squares fit (18) (or simplification (19) in ERTS) and the open-loop model (20). Obviously, due to linearization errors, the actual forward simulation will not be exactly coincident with the original estimated state trajectory but, anyway, performance indices will be computed with such input and state sequences.
- The maximum number of iterations will be fixed to 30, and cost indices are evaluated at each iteration to assess convergence speed. Here the concept of iteration means carrying out the necessary computations such that the cost is monotonically reduced (which, in a general case, does involve *nested* iterations to assess suitable step sizes).
- Convergence (in iterative strategies) is achieved when the relative performance improvement is below a given relative threshold value  $\left| \frac{J_s(i, k) - J_s(i, k-1)}{J_s(i, k-1)} \right| < 10^{-4}$ .

### B. Mean-cost and Worst-cost ratios

Figures 1 and 2 depict the above-defined mean-cost ratio  $MCR_s(k)$  and worst-cost ratio  $WCR_s(k)$ , respectively, in logarithmic scale for first 10 iterations. It can be clearly appreciated that strategies #2, #3 and #5 provide an initial solution ( $k = 1$ ) with a significantly lower cost over strategies #1 and #4. This was expected, due to the approximately

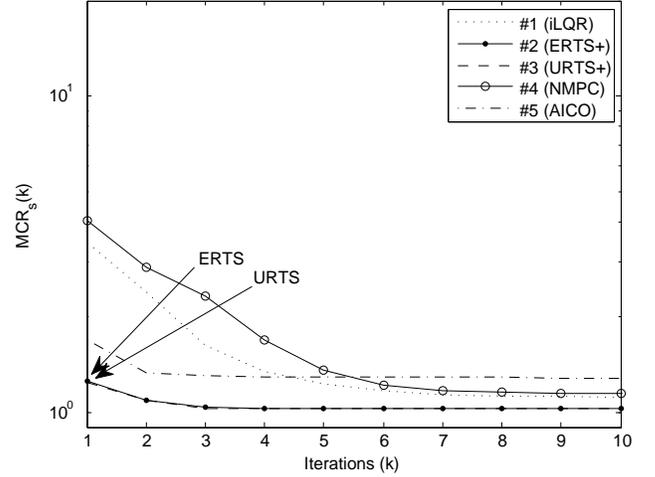


Fig. 1. Mean-cost ratio (MCR) per iteration.

optimal output of the proposed non-iterative ERTS, URTS and the first iteration of message-passing algorithm in AICO (#5). It can be seen that AICO ( $k = 1$ ) shows worse performance than ERTS and URTS, likely because mean and covariance of Gaussian messages use a linearized approach (recall that ERTS use the non-linear system to predict states and linearization to predict covariances and URTS uses the UT to account non-linearities). After 10 iterations, iLQR initialized with ERTS or URTS provides the best performance (actually, they converge in around 4 iterations). Table I shows the numeric results of the analysis for the convergence values for different performance. Some remarks about particular numeric values of Table I can be made:

- The mean cost ratio of (non-iterative) ERTS algorithm is 1.25, and the worst cost is 2.48. Interestingly, 45% of cases yield a performance cost nearly optimal.
- The mean cost ratio of URTS algorithm is 1.24, and the worst cost is 2.41, providing only marginally better accuracy than ERTS. In this case, 48% of cases provide a cost nearly optimal.
- The mean cost ratio for ERTS+ and URTS+ is 1.03, providing even better accuracy than iLQR due to its initialization. Table I shows that 90%–91% of cases are below 10% penalty after convergence, which is achieved in 4 iterations (in mean).
- AICO is able to improve over ERTS only in 54% of cases, while NMPC gets stuck in 18% of cases (or needs more iterations to converge).

### C. Computational resources

Table II shows the mean computational effort for every strategy and simulation relative to ERTS. It can be clearly shown that ERTS computational costs is significantly lower for the same performance with respect to iterative approaches. URTS and iLQR have a computational cost 2.6 and 2.65 times higher than ERTS, respectively.

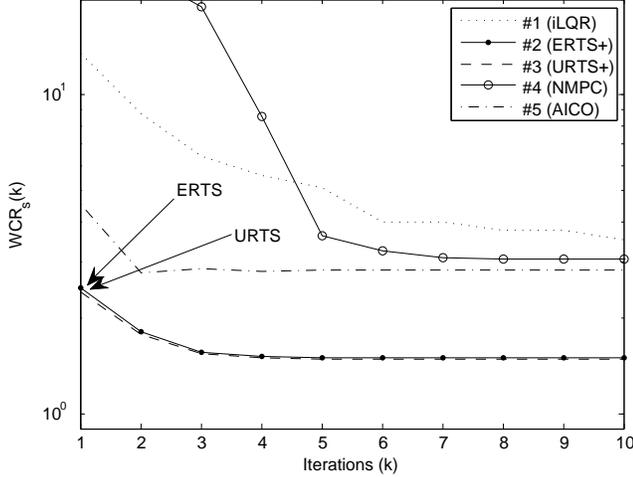


Fig. 2. Worst-cost ratio (WCR) per iteration.

The ERTS+ also improves the computational cost with respect to iLQR in  $\frac{6.75}{5.41} \rightarrow 23\%$ , providing even more accurate results.

As a consequence, ERTS can be considered a computationally efficient way of obtaining near-optimal results in practice, either by itself or as a “seed” for other iterative approaches.

In addition to this, ERTS+ is also an attractive option considering cost/computational time trade-off. In this sense, under real-time deadlines with we can iterate ERTS+ until deadline hit or with just very few iterations. The performance, in terms of accuracy, robustness and computational time is expected to be better than iLQR. URTS didn’t outperforms ERTS in any meaningful way, and the other options required far more computing time and parameter tweaking to obtain the best results.

TABLE I. MEAN AND WORST-CASE COST RATIO WITH RESPECT TO BEST RESULT; PERCENTAGE OF CASES BETTER THAN SOME PERFORMANCE LEVELS

	ERTS	URTS	iLQR	ERTS+	URTS+	NMPC	AICO
MCR	1.25	1.24	1.07	1.03	1.03	1.15	1.28
WCR	2.48	2.41	2.11	1.5	1.49	3.05	2.81
$\leq$ ERTS	—	100%	96%	100%	100%	91%	54%
$\leq$ URTS	0%	—	96%	93%	100%	91%	52%
$\approx$ Best <sup>†</sup>	45%	48%	89%	90%	91%	82%	29%

<sup>†</sup> Percentage of cases with an overcost below 10% w.r.t.  $J_{best}$ .

TABLE II. MEAN EXECUTION TIME RATIO WITH RESPECT TO ERTS FOR SOME PERFORMANCE LEVELS

	ERTS	URTS	iLQR	ERTS+	URTS+	NMPC	AICO
$\leq$ ERTS	1	2.6	2.65	1	2.6	77.6	4
$\leq$ URTS	—	2.6	2.68	1.29	2.6	79.3	4.1
Converged	—	—	6.75	5.41	7.04	175.5	19.6

#### D. Experiment: path following with Pioneer 3DX

In this section, a trajectory following experiment using a Pioneer 3DX mobile robotic platform [28], see Figure 4, has been developed in order to validate the ERTS controller. The implementation was carried out with a receding horizon policy. Actual performance was stable as, indeed, the finite-time horizon was chosen long enough, so the resulting control actions were not significantly modified by variations on such horizon (indeed, stability for short horizons might have needed additional tweaks with the terminal cost [29]).

The Pioneer P3-DX has differential configuration. An embedded board, Odroid-U2, with 1.7GHz ARM Quad-Core (Cortex-A9) processor and 2GB RAM is mounted inside the Pioneer robot. Results obtained are based on our implementation of ERTS algorithm using ROS Hydro [24] in Ubuntu 13.10. Our C++ code implementation of ERTS takes approximately 11.28 ms with horizon  $N = 60$ . Obviously, computing time increases linearly with  $N$ .

The robot includes a Hokuyo URG 04LX range laser which has been used jointly with odometry measurements using an AMCL particle filter [30] to estimate the robot position and speed without the drift that odometry-only sensing would have produced. The sensory system operates at a sampling period of 0.1 s. This is also the sampling period for the controller, which accepts linear and angular reference velocities and internally regulates wheels velocities based on a PID control. As our cost index intentionally includes acceleration weighting, the integrators of linear and angular accelerations to get velocities in the model (21) were actually included in the controller equations.

The goal is driving the P3-DX robot around an  $\infty$ -shape path:

$$\bar{x}_t = A \cos(\alpha_t) \quad (24)$$

$$\bar{y}_t = B \sin(2\alpha_t) \quad (25)$$

where  $\alpha_t = \frac{2\pi}{T}t \in [0, 2\pi]$  rad is the angle parametrizing curve, with  $T = 50$  being the trajectory period (the time closing the  $\infty$ -shape). In our experimentation, the ellipsoid major and minor semiaxis containing the  $\infty$ -shape have been set to  $A = 3\text{m}$  and  $B = 1\text{m}$ , respectively. From this path, references for angles and speeds are easily computed as:

$$\bar{\theta}_t = \arctan(\dot{y}_t/\dot{x}_t) \quad (26)$$

$$\bar{v}_t = \sqrt{\dot{x}_t^2 + \dot{y}_t^2} \quad (27)$$

$$\bar{\omega}_t = \frac{\ddot{y}_t\dot{x}_t - \dot{y}_t\ddot{x}_t}{\bar{v}_t^2} \quad (28)$$

so the overall state reference trajectory  $\mathbf{s}_t$  in (22) is available. The chosen penalization matrices and horizon for the experimentation are identical to the ones in the previous section, as well as  $\mathbf{h}(\mathbf{x}_t)$ .

Figure 3 shows a screenshot at time instant  $t_1 = 3\text{s}$  produced by Rviz software in ROS, where the robot is still far from the reference trajectory. This figure shows the reference trajectory (the yellow dot indicates current value  $\bar{x}_{t_1}, \bar{y}_{t_1}$ ; magenta line indicates the future values until  $t_2 = t_1 + N \cdot \Delta t = 9\text{s}$  and cyan

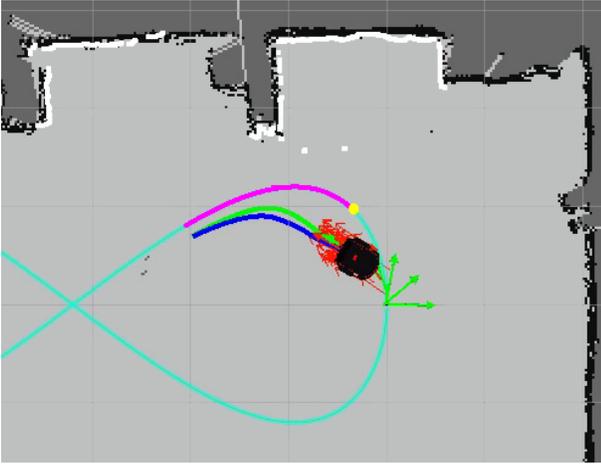


Fig. 3. Screenshot of experiment at time  $t^* = 3s$ . Yellow dot, magenta line and cyan line are current, future and full reference values, respectively; blue and green are predicted and smoothed trajectories; red arrows are position estimates; gray and black is the map; white are laser dots.

line is the full reference), the predicted and smoothed ERTS Cartesian trajectories (blue and green, respectively), AMCL particle position estimates (red arrows) and the map (in gray and black) and laser scan over-imposed dots (in white).

Figures 4 to 6 show the results obtained from the experimentation of the robot following the above described trajectory.

In Figure 5(a) the XY-trace is depicted, superimposed with the map used for positioning. An actual set of frames (superimposed photographs) appears in Figure 4. Figure 5(b) shows system's states along time:  $x$  and  $y$  positions, orientation, linear and angular speeds. Computed inputs are shown in Figure 6 (linear and angular accelerations depicted in black solid line), together with robot accelerations (in red) estimated from measurements by suitable differentiation and filtering of odometry data.

As a conclusion of the experiments, the ERTS successfully performs the trajectory control as expected. As error figures are quite low, given that ERTS is a non-iterative algorithm with predictable computation time, it is a viable alternative to other iterative approaches whose execution time might depend on initialization and stop conditions so their real-time suitability is harder to assess beforehand.

## V. CONCLUSIONS

A duality-based Extended Rauch-Tung-Striebel (ERTS) has been presented. The ERTS controller solves the control task via the transformation of the original problem to a dual estimation problem, which uses future reference states as observations. The dual problem is solved via a Rauch-Tung-Striebel smoother for linearized system (non-iterative). In a linear time-varying case, the algorithm would be equivalent to the well-known unconstrained LQR control.

An extension using Unscented Transform has been also presented, coined as URTS. However, the performance improvement over ERTS was almost unnoticeable and the computational cost was almost 3 times higher.

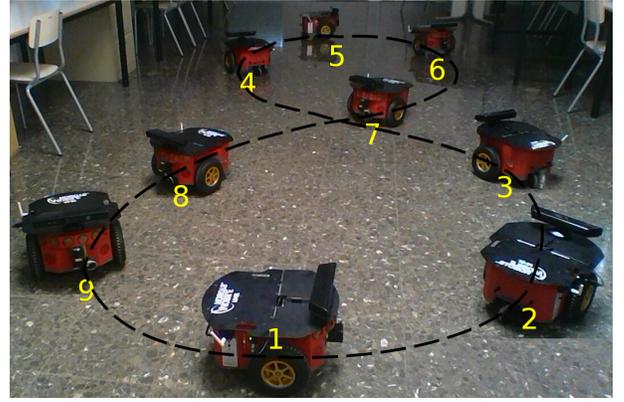
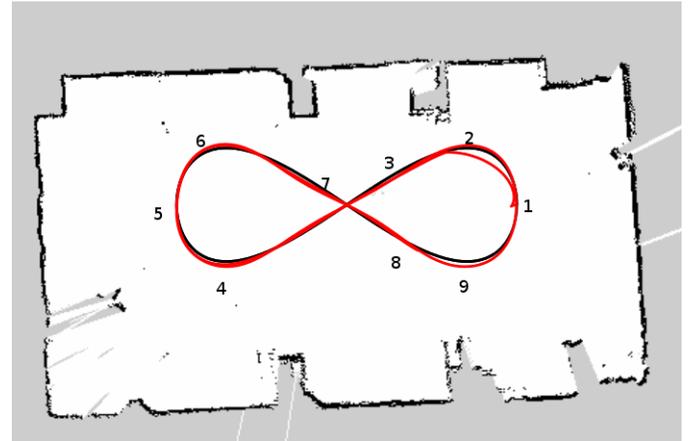
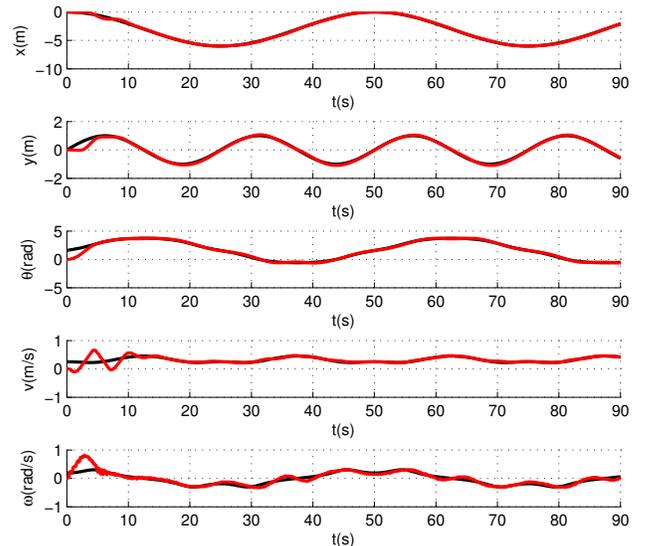


Fig. 4. Screenshots of the trajectory following experiment with AMCL filter (driftless position system).



(a)



(b)

Fig. 5. States of the system when following an infinite trajectory, (a) XY reference trajectory and robot trace, (b) reference and measured states. In both subfigures, references are depicted in black solid line and measured states in red solid line.

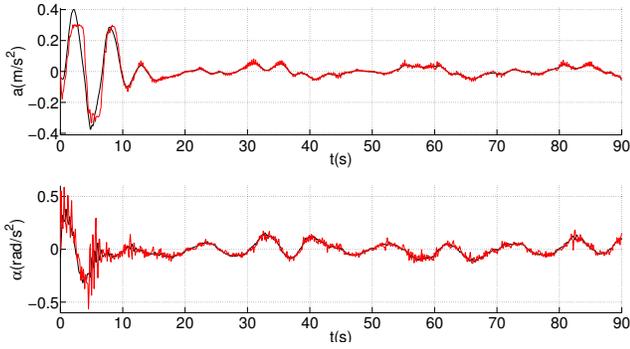


Fig. 6. Applied control inputs (black solid line) and accelerations estimated from measurements (red solid line).

The performance of the proposed controllers is studied on a trajectory-following problem of a 5th-order mobile robot and compared with nonlinear iterative linearization-based algorithms such as iLQR, NMPC and AICO. Simulations show that ERTS proposes nearly the same control as iLQR controller do after convergence, but with a significant computational cost reduction. In fact, ERTS can be considered a good choice for initialization of the above iterative algorithms. Hence, accuracy and reduced computational cost makes ERTS an interesting option for real-time control. Experimental implementation with a Pioneer P3-DX robot confirms these conclusions.

## APPENDIX

### A. Nonlinear Stochastic Optimal Control Problem

Results from Section II.A on duality of estimation and control is now elaborated in detail. Substituting (4) into (3) and using simplified notation  $p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t) \triangleq p$ ,  $\bar{p}(\mathbf{x}_{t+1}|\mathbf{x}_t) \triangleq \bar{p}$ ,  $q_t(\mathbf{x}_t, \mathbf{s}_t) \triangleq q_t$ ,  $\pi_t(\mathbf{x}_t, \mathbf{s}_{t:N}) \triangleq \pi_t$  and  $J_t(\mathbf{x}_t, \mathbf{s}_{t:N}) \triangleq J_t$  we obtain

$$\begin{aligned} J_t &= q_t + \min_{\pi_t} \int_{\mathbb{R}^{n_x}} \left[ J_{t+1} + \log \frac{p}{\bar{p}} \right] p \, d\mathbf{x}_{t+1} \\ &= q_t - \log c_t + \min_{\pi_t} \text{KL} \left( p \parallel \frac{1}{c_t} e^{-J_{t+1}} \bar{p} \right) \end{aligned} \quad (29)$$

where the normalizer  $c_t \triangleq c_t(\mathbf{x}_t, \mathbf{s}_{t+1:N})$  is equal to

$$c_t(\mathbf{x}_t, \mathbf{s}_{t+1:N}) = \int_{\mathbb{R}^{n_x}} e^{-J_{t+1}(\mathbf{x}_{t+1}, \mathbf{s}_{t+1:N})} \bar{p}(\mathbf{x}_{t+1}|\mathbf{x}_t) d\mathbf{x}_{t+1}$$

The second argument of the KL divergence (29) is thus a proper probability density  $p_\pi$  (10).

If there exists a control policy  $\pi_t$  such that  $\text{KL}(p||p_\pi) = \zeta_t$ , where  $\zeta_t$  is a constant independent of  $\mathbf{x}_t, \mathbf{u}_t$ , then (29) can be rewritten as:

$$e^{-J_t} \propto e^{\zeta_t - q_t} c_t = e^{\zeta_t} e^{-q_t} \int_{\mathbb{R}^{n_x}} e^{-J_{t+1}} \bar{p} d\mathbf{x}_{t+1} \quad (30)$$

### B. Dual estimation problem

Assume now a state-space model in the form

$$\mathbf{x}_{t+1} \sim \bar{p}(\mathbf{x}_{t+1}|\mathbf{x}_t) \quad (31)$$

$$\mathbf{s}_t \sim p(\mathbf{s}_t|\mathbf{x}_t) \quad (32)$$

where (31) stands for the reference dynamics from the original control problem and (32) is an observation model.

The marginal predictive distribution is:

$$\begin{aligned} p(\mathbf{s}_{t:N}|\mathbf{x}_t) &= p(\mathbf{s}_t|\mathbf{x}_t)p(\mathbf{s}_{t+1:N}|\mathbf{x}_t) \\ &= p(\mathbf{s}_t|\mathbf{x}_t) \int_{\mathbb{R}^{n_x}} p(\mathbf{s}_{t+1:N}|\mathbf{x}_{t+1}) \bar{p}(\mathbf{x}_{t+1}|\mathbf{x}_t) d\mathbf{x}_{t+1} \end{aligned} \quad (33)$$

Comparing (33) with (30) we can establish duality for

$$p(\mathbf{s}_t|\mathbf{x}_t) \propto e^{-q_t(\mathbf{x}_t, \mathbf{s}_t)} \quad (34)$$

$$p(\mathbf{s}_{t+1:N}|\mathbf{x}_{t+1}) \propto e^{-J_{t+1}(\mathbf{x}_{t+1}, \mathbf{s}_{t+1:N})} \quad (35)$$

establishing a duality between the Bellman function and marginal prediction.

Moreover, we note that

$$p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{s}_{t+1:N}) \propto p(\mathbf{s}_{t+1:N}|\mathbf{x}_{t+1}) \bar{p}(\mathbf{x}_{t+1}|\mathbf{x}_t) \quad (36)$$

establishing duality

$$p_\pi(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{s}_{t+1:N}) \propto p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{s}_{t+1:N}), \quad (37)$$

where the normalization constants absorb the constant terms in the Bellman function (i.e. the  $\zeta_t$  term in (30)). The dual estimation problem is defined as the smoothing problem of  $\mathbf{x}_{t+1}$  knowing  $\mathbf{x}_t$  and the whole observation sequence  $\mathbf{s}_{t+1:N}$ , [8].

### C. Linear quadratic regulator from KL cost

Let us consider the LTV system with Gaussian noise (1) and its reference dynamics (14), for a special case of linear system  $\mathbf{f}(\mathbf{x}_t) = \mathbf{A}_t \mathbf{x}_t$ . The KL divergence (5) is

$$\begin{aligned} \text{KL}(p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t) || \bar{p}(\mathbf{x}_{t+1}|\mathbf{x}_t)) &= \\ &= \frac{1}{2} \mathbf{u}_t^T \mathbf{B}_t^T \bar{\mathbf{V}}_t^{-1} \mathbf{B}_t \mathbf{u}_t + c_{KL}, \end{aligned} \quad (38)$$

$$c_{KL} = \frac{1}{2} \left( \text{tr}(\bar{\mathbf{V}}_t^{-1} \mathbf{V}_t) - \dim(\mathbf{x}_t) - \ln \frac{\det(\mathbf{V}_t)}{\det(\bar{\mathbf{V}}_t)} \right), \quad (39)$$

where  $\dim()$  denotes dimension of a vector. Matching the quadratic term in (38) with  $\mathbf{u}_t^T \mathbf{R}_t \mathbf{u}_t$  in the loss function (13) establishes the choice of  $\bar{\mathbf{V}}_t$  in (14).

Without loss of generality, let us assume a reference trajectory  $\mathbf{s}_{0:N} = \mathbf{0}$ .

Let us now show that, indeed, both the duality-based result and the standard Riccati equations give coincident control laws. Consider Bellman function  $J_N = \frac{1}{2} \mathbf{x}_N^T \mathbf{S}_N \mathbf{x}_N$  with  $\mathbf{S}_N = \mathbf{Q}_N$ , for  $t = N-1$  the following holds:

$$p_\pi(\mathbf{x}_{t+1}|\mathbf{x}_t) \propto e^{-\mathbf{x}_N^T \mathbf{S}_N \mathbf{x}_N} \mathcal{N}(\mathbf{A}_{N-1} \mathbf{x}_{N-1}, \bar{\mathbf{V}}_{N-1}) \quad (40)$$

$$= \mathcal{N}(\hat{\mathbf{x}}_\pi, \Sigma_\pi), \quad (41)$$

with moments  $\hat{\mathbf{x}}_\pi = \Sigma_\pi^{-1} \bar{\mathbf{V}}_{N-1}^{-1} \mathbf{A}_{N-1} \mathbf{x}_{N-1}$ ,  $\Sigma_\pi = (\mathbf{S}_N + \bar{\mathbf{V}}_{N-1}^{-1})^{-1}$

The KL divergence (9) to minimize is then:

$$\text{KL}(p||p_\pi) = \frac{1}{2} \mathbf{e}_{N-1} \Sigma_\pi^{-1} \mathbf{e}_{N-1} + c_{KL} \quad (42)$$

$$\mathbf{e}_{N-1} = \mathbf{A}_{N-1} \mathbf{x}_{N-1} + \mathbf{B}_{N-1} \mathbf{u}_{N-1} - \Sigma_\pi^{-1} \bar{\mathbf{V}}_{N-1}^{-1} \mathbf{A}_{N-1} \mathbf{x}_{N-1}, \quad (43)$$

which is minimized for

$$\begin{aligned} \mathbf{B}_{N-1} \mathbf{u}_{N-1} &= -\mathbf{B}_{N-1} \mathbf{K}_{N-1} \mathbf{x}_{N-1} \\ \mathbf{K}_{N-1} &= (\mathbf{R}_{N-1} + \mathbf{B}_{N-1}^T \mathbf{S}_N \mathbf{B}_{N-1})^{-1} \mathbf{B}_{N-1}^T \mathbf{S}_N \mathbf{A}_{N-1} \end{aligned}$$

with minimum  $\min \text{KL}(p||p_\pi) = c_{KL}$  which is independent of the state and input. Here,  $\mathbf{K}_{N-1}$  was obtained using the matrix inversion lemma.

Because the KL divergence term in (29) is constant, the cost-to-go is equal to

$$J_{N-1} = \frac{1}{2} \mathbf{x}_{N-1}^T \mathbf{Q}_{N-1} \mathbf{x}_{N-1} - \log c_{N-1}(\mathbf{x}_{N-1}) + c_{KL}$$

where  $c_{N-1}(\mathbf{x}_{N-1}) \propto e^{-\frac{1}{2} \mathbf{x}_{N-1}^T \mathbf{A}_{N-1}^T (\mathbf{S}_N + \bar{\mathbf{V}}_{N-1}^{-1}) \mathbf{A}_{N-1} \mathbf{x}_{N-1}}$ . This implies that the cost-to-go remains quadratic  $J_{N-1} = \frac{1}{2} \mathbf{x}_{N-1}^T \mathbf{S}_{N-1} \mathbf{x}_{N-1} + C'$  with some additional terms that do not affect the minimization and  $\mathbf{S}_{N-1}$  takes the form of the well-known Riccati equation:

$$\begin{aligned} \mathbf{S}_{N-1} &= \mathbf{Q}_N + \mathbf{A}_{N-1}^T (\mathbf{S}_N^{-1} + \bar{\mathbf{V}}_{N-1})^{-1} \mathbf{A}_{N-1} \\ &= \mathbf{Q}_N + \mathbf{A}_{N-1}^T (\mathbf{S}_N [\mathbf{I} - \mathbf{B}_{N-1} \mathbf{K}_{N-1}]) \mathbf{A}_{N-1} \quad (44) \end{aligned}$$

The recursion for  $t = N - 2, \dots$  leads to the standard LQR control law.

#### ACKNOWLEDGMENT

This research has been supported by the European Regional Development Fund and Ministry of Education, Youth and Sports of the Czech Republic under project No. CZ.1.05/2.1.00/03.0094: Regional Innovation Centre for Electrical Engineering (RICE) and the Czech Science Foundation under the project GAR P 102/11/0437, and from VALi+d Program and PrometeoII/2013/004 (Generalitat Valenciana), project DPI2011-27845-C02-01 from Spanish Government.

#### REFERENCES

- [1] M. Morari and J. H. Lee, "Model predictive control: Past, present and future," *Computers and Chemical Engineering*, vol. 23, pp. 667–682, 1997.
- [2] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3 – 20, 2002.
- [3] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control engineering practice*, vol. 11, no. 7, pp. 733–764, 2003.
- [4] L. Blackmore, M. Ono, A. Bektassov, and B. Williams, "A probabilistic particle-control approximation of chance-constrained stochastic predictive control," *Robotics, IEEE Transactions on*, vol. 26, no. 3, pp. 502–517, June 2010.
- [5] L. Grüne and J. Pannek, *Nonlinear Model Predictive Control: Theory and Algorithms*, 1st ed., ser. Communications and Control Engineering. Springer, 2011.
- [6] J. Yong and X. Zhou, *Stochastic Controls: Hamiltonian Systems and HJB Equations*, ser. Applications of mathematics. Springer New York, 1999.
- [7] W. Li and E. Todorov, "Iterative linear quadratic regulator design for nonlinear biological movement systems," in *ICINCO*, 2004, pp. 222–229.
- [8] E. Todorov, "General duality between optimal control and estimation," in *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, 2008, pp. 4286–4292.
- [9] M. Toussaint, "Robot trajectory optimization using approximate inference," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML '09. New York, NY, USA: ACM, 2009, pp. 1049–1056. [Online]. Available: <http://doi.acm.org/10.1145/1553374.1553508>
- [10] E. Todorov, "Finding the most likely trajectories of optimally controlled stochastic systems," in *In World Congress of the International Federation of Automatic Control (IFAC)*, 2011.
- [11] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations," *Exploring artificial intelligence in the new millennium*, vol. 8, pp. 236–239, 2003.
- [12] M. Zima, L. Armesto, V. Girbes, A. Sala, and V. Smidl, "Extended rauch-tung-striebl controller," in *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, Dec 2013, pp. 2900–2905.
- [13] S. Julier, J. Uhlmann, and H. Durrant-Whyte, "A new method for the nonlinear transformation of means and covariances in filters and estimators," *Automatic Control, IEEE Transactions on*, vol. 45, no. 3, pp. 477–482, Mar 2000.
- [14] V. Girbes, L. Armesto, and J. Tornero, "Path following hybrid control for vehicle stability applied to industrial forklifts," *Robotics and Autonomous Systems*, vol. 62, pp. 910–922, 2014.
- [15] K. Jiang and L. Seneviratne, "A sensor guided autonomous parking system for nonholonomic mobile robots," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 1, 1999, pp. 311–316.
- [16] I. Papadimitriou and M. Tomizuka, "Fast lane changing computations using polynomials," in *American Control Conference, 2003. Proceedings of the 2003*, vol. 1, 2003, pp. 48–53.
- [17] M. Michalek and K. Kozowski, "Vector-field-orientation feedback control method for a differentially driven vehicle," *Control Systems Technology, IEEE Transactions on*, vol. 18, no. 1, pp. 45–65, Jan 2010.
- [18] J. Villagra and D. Herrero-Perez, "A comparison of control techniques for robust docking maneuvers of an AGV," *Control Systems Technology, IEEE Transactions on*, vol. 20, no. 4, pp. 1116–1123, July 2012.
- [19] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 1951.
- [20] H. E. Rauch, F. Tung, and C. T. Striebel, "Maximum likelihood estimates of linear dynamic systems," pp. 1445–1450, 1965.
- [21] E. Todorov and W. Li, "A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems," in *American Control Conference, 2005. Proceedings of the 2005*. IEEE, 2005, pp. 300–306.
- [22] S. Sarkka, "Unscented rauch–tung–striebl smoother," *Automatic Control, IEEE Transactions on*, vol. 53, no. 3, pp. 845–849, 2008.
- [23] C. K. Chui and G. Chen, *Kalman Filtering with Real-time Applications*. New York, NY, USA: Springer-Verlag New York, Inc., 1987.
- [24] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.
- [25] C. Leondes, J. Peller, and E. B. Stear, "Nonlinear smoothing theory," *Systems Science and Cybernetics, IEEE Transactions on*, vol. 6, no. 1, pp. 63–71, 1970.
- [26] A. Scheuer and M. Xie, "Continuous-Curvature Trajectory Planning for Manoeuvrable Non-Holonomic Robots," in *IEEE-RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 3, 1999, pp. 1675–1680.

- [27] Y. Tassa, "iLQG/DDP package," Sept. 2011. [Online]. Available: <http://www.cs.washington.edu/people/postdocs/tassa/code/>
- [28] A. MobileRobots, "Pioneer p3-dx." [Online]. Available: <http://www.mobilerobots.com/ResearchRobots/PioneerP3DX.aspx>
- [29] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789 – 814, 2000. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0005109899002149>
- [30] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.