Receding Horizon Control of Uncertain Systems

Behnood Gholami

A Thesis
in
The Department
of
Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Applied Science at
Concordia University
Montreal, Quebec, Canada

July 28, 2005

Your file Votre référence
ISBN: 0-494-10264-0
Our file Notre référence
ISBN: 0-494-10264-0

Abstract

Behnood Gholami

Receding Horizon Control of Uncertain Systems

In the first part of this thesis stability robustness for quasi-infinite Receding Horizon Control (RHC) of an uncertain nonlinear system is investigated. A sufficient condition is developed for stability of a general nonlinear RHC system subject to perturbations. The result is further specialized to linear systems. For this case it is demonstrated that the closed-loop system is stable out side a bounded set containing the desired equilibrium point upon satisfaction of an LMI constraint along with a bounded perturbation assumption. The new result is applied for control of a mobile robot system which demonstrates the validity of the approach. In the second part, RHC of an uncertain nonlinear system is considered where the computational time is not negligible. The existing method proposes a solution to deal with non-zero computation time by predicting the states at the next sampling time, which provides the controller with sufficient time to generate the required input signal. This work extends this previous result by applying neighboring extremal paths theory to improve the performance further through the addition of a correction phase to the algorithm. The proposed method is composed of three steps: state prediction, trajectory generation, and trajectory correction. The new approach is applied for control of a mobile robot system, which demonstrates significant performance improvements over the existing method.

Acknowledgments

This thesis was carried out in the Control and Information System Laboratory (CIS) at Concordia University, Montreal, Canada. I am grateful for having the opportunity to be a part of CIS research team.

I would like to thank my supervisor, Dr. Brandon Gordon, who not only helped me from the scientific point of view, by showing me the way whenever I was lost and giving me ideas that were the key to solve my problems, but also in teaching me the philosophy of science and control engineering. A supervisor that posses understanding of both the science and its philosophy and who supports his students throughout all the ups and downs of their student life, is rare to find and I am really happy (and lucky) that I had this opportunity to work under his supervision.

I also would like to thank my co-supervisor, Dr. C. A. Rabbath, who proposed the subject of Receding Horizon Control of Cooperative Systems to me, supported me during this period by giving me guidelines, ideas and encouragement and helped me to improve my work.

Thanks very much to my great friend whom I had the opportunity to share our office (and ideas) with, Behzad Samadi. I was lucky to spend almost all my time at Concordia with him.

I would like to thank my parents, to whom I have dedicated this thesis. Without their support this work was impossible. I will always be in debt for their never ending love, support and encouragements. Finally, my thanks go to Farnood and Farnaz for always being beside me and giving me motivation to move forward.

To Mom and Dad

Table of Contents

List of Figures

Nomenclature

A, **B**	parameters matrixes of the linear system
b	upper bound of the perturbation terms
B_r	ball of radius r
C	matrix of system parameters
F	force input to the mobile robot
$\boldsymbol{f}$	vector of governing differential equations of the system under consideration
$\boldsymbol{g}$	perturbation vector appearing in the state equations
h	execution horizon (in section 3, equivalent to δ in section 2)
H	$n \times n$ positive definite matrix
H	Hamiltonian
J	cost function
$J_T^*(.)$	optimal cost function with the time horizon of T
K	Lipschitz constant of the cost function
L	Lipschitz constant of $\boldsymbol{f}$
L_a	the distance from the middle of the wheel axis of the robot to the chosen point (x_1,x_3)
O(.)	Order operator
P, **Q**, **R**	weighting matrixes
r	radius of the ball containing the region of attraction of the closed-loop system

Q_δ	Incremental cost
$\tilde{\mathbf{Q}}$	$n \times n$ matrix
T	prediction horizon
U	Set of admissible inputs
$\boldsymbol{u}$	input vector
$\boldsymbol{u}^*$	optimal input vector
$\boldsymbol{u}^*(\tau; \boldsymbol{x}(t))$	optimal input vector with $\boldsymbol{x}(t)$ being the initial conditions
$\boldsymbol{u}_0^*$	value of the optimal input at the initial time
v	forward speed of the mobile robot
X_c	bounded set in $\Re^n$
$\boldsymbol{x}$	state vector of the nominal system
$\boldsymbol{x}(\mathrm{t};\boldsymbol{x}_0)$	state vector with x0 being the initial condition for the system
$\boldsymbol{x}_c$	Coordinates of the center of the ellipsoid
$\boldsymbol{x}_e$	coordinates of a point on the boundary of an ellipsoid centered at the origin
$\boldsymbol{x}^*$	optimal state vector
$\boldsymbol{x}_0^*$	value of the optimal state at the initial time
x_1, x_3	x-y coordinates of an off-axis point of the mobile robot
$\boldsymbol{x}_{actual}$	vector of the actual states of the system
$\boldsymbol{x}_{predict}$	vector of the predicted states of the system
x, y	coordinates of the mobile robot
y	state vector of the actual system

α	positive constant defining the terminal constraint set
$\alpha_1, \alpha_2, \alpha_3, \alpha_4$	random variables in [-1,1]
δ	execution horizon
$\boldsymbol{\lambda}$	vector of co-states variables
θ	yaw angle of the mobile robot
τ	torque input to the mobile robot
Ω	region of attraction of the closed-loop RHC system
Ω_α	terminal constraint set
ω	angular velocity of the mobile robot

1 . Introduction

1.1 Motivation

Multi-agent systems have received a growing amount of interest in the past couple of years in different disciplines, each having their specific reason for using multiple agents instead of one. Challenges involved in multi-agent systems control are

- Their dynamic environment, therefore requiring a control method capable of changing the strategy online.
- Constraints on the state space and inputs, as there are always constraints present due to the available physical space for the movement of the agents and also constraints on the actuating power.
- Goal accomplishment while minimizing a cost, as is the case for almost all engineering systems.

Receding Horizon Control (RHC), however, is considered a proper technique in dealing with such systems due to its most distinguished feature which involves repeated online solution of optimal control problems. This property, which enables application of this scheme on systems operating in dynamic environments (*i.e.* environments which are not completely identified - avoiding a pop-up threat in multi Unmanned Aerial Vehicle (UAV) missions is an example), along with its ability to handle constraints on states and inputs while optimizing a specific cost function are the main reasons that encourage researchers to apply this techniques to multi-agent systems.

Since multi-agent systems appear in the research work of people in different disciplines having different objectives, in the next section we briefly review the works done so far and categorize them into three main groups. The following discussion, reviews the huge body of work done in the multi-agent systems. This, however, still serves as the motivation for our work on RHC. A comprehensive review of the work done in RHC will follow this discussion.

1.2 A Discussion on Multi-Agent Systems

The majority of the research work addressing the problem of multi-agent systems fall into the one of the following categories:

- Physics and Mathematics
- Control Theory
- Computer Graphics

While the works appearing in the first category has more intension towards analysis of such systems, the works in the other two categories address both analysis and design issues. The remainder of this section is devoted to describe each category's point of view and some relevant works done in this area.

While there has been some literature review works addressing the multi-agent systems in general (*e.g.* [1] and [2]), this has always been limited to works in a specific discipline, where the multi-disciplinary flavor of this subject is not evident in them. A survey similar to the present section can be found in the introduction section of [3], however, the range of topics covered is not as complete as the materials discussed here.

1.2.1 Physics and Mathematics

The motivation for the researchers in this field comes from the fact that insects, animals, bacteria and other *agents* which try to form a group or swarm, usually show complex behaviors resulting from very simple rules. This includes an attraction/repulsion law between the members of a swarm and some reactions in response to environmental effects. A major body of work in this category tackles the problem of analysis of a swarm of agents, where a member of such swarm, called a motile element [4], has the ability to move around and sense neighboring agents. A common assumption in such papers is that agents are considered as point masses (see for example [5]) and the interaction forces between the members are similar to inter-molecular attraction/repulsion profile [4]. The number of the particles considered are usually large and the main objective of these articles are to understand the general characteristic of such systems resulting from a large number of individuals and a limited number of rules.

The interested reader is encouraged to refer to papers [5],[6],[7] and the references therein for a more detailed description.

1.2.2 Control Theory

It is almost always true that there is strength in numbers. Not only that a group of agents can perform tasks that a single agent is unable to do, there are several other factors, *e.g.* robustness to failure of individual units and dealing with a dynamic environment, economic cost and simplicity of design of individual agents, that has motivated the use of multiple agents instead of one in numerous applications. Multi-agent systems are believed to be useful in a wide range of applications including but not limited to search and rescue missions, deep space interferometry, fire fighting in forests, to name a few.

Although the general frame work for multi-agent systems is common, agents might have different realizations in the physical world, namely, spacecrafts [8],[9], UAVs [10], mobile robots and land vehicles [11],[12] or marine crafts [13]. While analysis of the group behavior has been considered in this context, papers in this category are more or less concerned with the design of control laws and strategies that result in a desired pre-specified group behavior.

The methods used to tackle such problems are diverse; among those use of the following methods is common:

- **Graph Theory** [14],[15], where the authors use this theory to address problems ranging from formation stabilization to changing between formation patterns.
- **Convex Optimization and Mixed-Integer Linear Programming** [16], where

the techniques used address problems such as fuel/time optimal algorithms for coordination of a group of agents.

- **Artificial Potential** [17], where it is used for collision avoidance and target tracking.
- **Decentralized Control Techniques** [11],[18], where the stabilization of the formation in the presence of local information and/or no central decision making unit is addressed.
- **Lyapunov Synthesis** [3],[19], where the authors seek a control law to stabilize the system.
- **Behavior-based Techniques** [12], which seems to be promising in tackling multi-objective problems, where the agents have to reach a goal while satisfying other requirements, *e.g.* obstacle avoidance, formation keeping,.... Although no guarantee of stability is provided in this method.
- **Receding Horizon Control (Model Predictive Control)** [20],[21], where a general (optimal) approach is being proposed for formation control in a decentralized way.
- **Game Theory** [22], where the authors use this framework mainly in the context of pursuit and evasion problems.

The reader is encouraged to refer to the papers cited above and the references therein for a full description.

We should make a distinction in the works done in this category, between flocking [3],[14] and moving in formation [18],[19]. While the desired condition in flocking is the *cohesiveness* of the members of the group (with no predefined shape) and moving towards a common target, in problems involving formation, the states of each individual member and the overall pattern of formation is important as well.

While the majority of the work done in the "Control Theory" category is somehow similar to the above mentioned papers, there exist some other articles having somehow a different taste and point of view, which are briefly summarized bellow:

- **Herding Problem (Sheepdog Problem)** [23],[24] - There is usually a swarm of agents (sheep) with local decentralized control law with a tendency of dispersion. Our means of control for guiding this group to a predefined area (*e.g.* a circle) is one or multiple *dogs*.

- **Pursuit Evasion** [25] - This type of problem, similar to the Herding Problem involves interaction between multiple groups, as there is some non-cooperative interaction between different agents. When it comes to non-cooperative games, use of concepts from Game Theory seems justified [22].

- **Entropy in Multi-Agent Systems** [26] - In this type of problems, authors have seen some connection between the *Entropy* (carrying a concept similar to what is defined in Thermodynamics) and the multi-agent systems. Few papers have addressed multi-agent systems in this context.

1.2.3 Computer Graphics

Multi-agent system has also been an interesting topic for those working on computer graphics, animation and video games. In this context, the realizations of the agents are autonomous characters that interact with each other in a cooperative or non-cooperative manner. In [27] which is one of the early works in this field, Reynolds has devised a set of *ad hock* control laws that results in a desired group behavior similar to what is seen in the nature by birds, fish, etc. Successful implementation of these algorithms in computer animation has been reported in the same article.

One of the best realizations of *autonomous characters* in the context of computer graphics is that seen in strategic games. In such games, while the user specifies a certain goal at the highest level, a set of heterogeneous characters (*e.g.* vehicles, soldiers, ...) perform the designated task in an autonomous manner. In such games, the opponents decision making, done by the software itself, is totally autonomous, both in the high-level and low-level tasks. Examples of such strategic games are [28],[29], where the goal is to defeat the opponent using the available resources. The user serves as the high level controller specifying general strategy, while the individual agents autonomously follow that strategy.

While the researchers in this field address both design and analysis of such systems, there is usually no guarantee of stability involved in their work. The overall behavior of such systems is the combination of a set of pre-specified behaviors, resembling the behavior-based approach in the previous section. Application of more rigorous techniques from other categories of this review in computer animation and video games seems to be promising and will help to construct more powerful and realistic simulations.

1.3 Research Objectives

The objective of this thesis is two-folded: In the first part of the thesis, stability robustness for quasi-infinite RHC of an uncertain nonlinear system is investigated and discussed further in the linear case. Although such robustness analysis is present for dual mode RHC, this issue has not been addressed in the context of quasi-infinite RHC. In quasi-infinite RHC, we seek an optimal input profile in a finite time span, while the terminal cost of the optimal control problem bounds the infinite horizon cost of the nonlinear system, therefore leading to the name *quasi-infinite*.

In the second part, RHC of an uncertain nonlinear system is considered where the computational time is not negligible. The existing method proposes a solution to deal with non-zero computation time by predicting the states at the next sampling time, which provides the controller with sufficient time to generate the required input signal. Although this approach solves the problem of practical implementation of RHC in the presence of computational delay, properties guaranteed by the theoretical work does not hold for such systems due to the fact that the pre-generated control signal is not optimal. This comes from the fact that uncertainties present in the system produce error in state prediction. We propose a new method that can provide the plant with the optimal input in the presence of the computational delay.

1.4 Literature Review

Receding Horizon Control (RHC), also known as Model Predictive Control (MPC), was first introduced in the process control community. It has attracted the attention of many researchers due to its ability to handle constraints on the states and inputs in multi-variable control problems [30]. Until recently this approach has found most of its applications for process control problems with slow dynamics. However, recent advances in computing performance and distributed computation has allowed the approach to be applied to mechanical systems with fast dynamics such as aerospace and mobile robot systems.

The RHC approach is essentially a repeated on-line solution to a finite horizon open loop optimal control problem. Based on the current state values, an optimal control problem is solved for a period of time called the *prediction horizon*. The first part of the computed optimal input is applied to the plant in a period of time called the *execution horizon* until the next sampling of the states becomes available, where again the same procedure is repeated. The *execution horizon* might be constant or time varying.

Since RHC is based on solving constrained optimization problems, constraints on inputs and states can be explicitly dealt with, a fact that makes it attractive in industrial applications where constraints on the states and saturation of inputs should be strictly observed. However, RHC of constrained systems is nonlinear in nature and thus Lyapunov theory must be used to study the stability of the nonlinear closed loop system [31]. As pointed out in [32], the repeated on-line solution of a finite horizon optimal control problem does not guarantee an asymptotic property such as stability. A number of methods have been proposed to guarantee closed loop stability where a terminal cost or a terminal constraint, or a combination thereof, is introduced in the optimization problem. For a detailed survey, the reader is referred to [31]. Recently, Jadbabaie [33] suggested the use of Control Lyapunov Functions to avoid the necessity of using terminal constraints, resulting in shorter computation time.

In the past, the repeated on-line solution of an open loop optimal control problem limited the application of RHC mainly to process control problems. In process control problems found in chemical industries, the dynamics of the plant is slow enough to allow for the computation of the optimal control and therefore computation time is not an issue. Today, with the current improvements in computing power available for control purposes and the introduction of faster optimization algorithms, such as the one suggested in [34], RHC can be applied to plants with fast dynamics, as found for instance in aerospace systems. Authors in [34] use an active set approach for solving the online optimization problem found in RHC and propose an algorithm which is capable of finding more accurate solutions to the optimization problem and at the same time being robust to the initial guess when compared to algorithms having the same computation time. In [35], a direct method for solving optimal control problems has been proposed based on the properties of *flat* outputs. The dimension of the optimization space is considerably reduced using such outputs, and makes it more attractive from the computational point of view. The method proposed in [35] was applied successfully to a vector thrust flight experiment in [36], which is an example of an aerospace system having fast dynamics.

The issue of robustness of RHC-based closed loop systems was addressed by Michalska and Mayne in [37], where they consider the dual-mode RHC. In such case, the receding horizon controller drives the states to a terminal region containing the desired equilibrium point. Once the state is in the terminal region, a classic linear controller is applied to stabilize the system. Later, Chen and Allgower [38] proposed a quasi-infinite horizon scheme, where the on-line optimization problem is repeatedly solved whether or not the states reach the terminal region. In quasi-infinite RHC, the terminal cost bounds the infinite horizon cost by penalizing the terminal states appropriately. This leads to a finite horizon optimal control problem having a *quasi-infinite* prediction horizon. The idea of linear feedback controller applied in a terminal region, in the vicinity of the equilibrium point, is also used in quasi-infinite RHC; however, the terminal controller is never implemented on the actual system and is solely used for the design of a stabilizing receding horizon controller. Thus, with quasi-infinite RHC there is no necessity for switching between controllers and the asymptotic stability property is achieved using a single mode controller. Also, the role of the terminal constraint penalty matrix is two-folded: it is serves as a bound for the infinite horizon cost and can be used to define the terminal region off-line. Therefore, quasi-infinite RHC appears to be more general compared to the dual-mode RHC and RHC involving terminal equality constraints [38]. However, robustness of this quasi-infinite RHC in the presence of disturbances has not been previously addressed.

On the other hand, addressing the problem of online optimization methods to solve such optimization problem and also the issue of computation time are unavoidable when the implementation of the RHC system is considered. In [45] and [46], authors propose dividing the nonlinear optimal control system architecture into two parts: An outer loop which generates the reference optimal outputs to be followed by the plant and an inner loop stabilizing the states of the system around the generated reference trajectory using neighbouring extremal paths theory. The main objective of the method described in [45] and [46] is to propose a real-time implementation strategy for optimal control schemes implemented on-line. The Legendre pseudospectral method is used to approximate the states, co-states and inputs instead of the B-Spline approximation proposed in [35]. In reference [45], the approximation method allows for rapid generation of optimal trajectories, which can potentially be useful for RHC problems.

In the literature, there are several studies on the stability of closed-loop systems obtained with a RHC strategy, albeit with a zero computation time assumption [33], [45] . In practice, however, there is no guarantee of closed-loop stability as the zero computation time assumption is violated, especially for systems with fast dynamics.

Although several methods for rapid generation of optimal trajectories exist [35], [45], the problem of non-zero computation time is unavoidable for mechanical systems. Milam *et. al.* address this issue in [44], by proposing a method involving prediction of the states at the next sampling time before hand, which gives the controller enough time to generate the optimal trajectories. The predicted states serve as initial conditions for the open loop optimal control problem, giving the controller a computation deadline equal to the sampling period to solve the optimisation problem. At the next sampling time, same prediction and optimal trajectory generation procedure is repeated.

1.5 Thesis Contribution and the Outline of the Thesis

We first review some background material including optimal control and the flatness property, which will be used in the subsequent chapters. To address the problem of stability robustness in the context of quasi-infinite RHC, we briefly review the quasi-infinite RHC in section 3.1. In section 3.2 a sufficient condition is developed for stability of a general nonlinear RHC system subject to perturbations. The result is further specialized to linear systems in section 3.3. For this case it is demonstrated that the closed loop system is stable outside a bounded set containing the desired equilibrium point upon satisfaction of an Linear Matrix Inequality (LMI) constraint along with a bounded perturbation assumption, which serves as the main contribution of the this chapter. A numerical example of a mobile robot of unicycle type presented in section 3.4, demonstrates the validity of the proposed conditions.

In the work presented in Section 4, the RHC of uncertain nonlinear systems with non-zero computation time is addressed. The theoretical background needed is presented in sections 4.1 and 4.2, where the general RHC scheme and the theory of neighboring extremal paths [47] are briefly discussed, respectively. In section 4.3 we build upon the work presented in [44] by proposing the addition of a correction phase to the prediction and trajectory generation in the RHC of systems with non-zero computation time. In our proposed method, the control signal design for the RHC system is obtained in three stages, the first two being those proposed in [44]. The prediction phase estimates the values of the states of the system in the next RHC sampling time. An optimal trajectory is designed with the initial conditions of the optimal open loop problem being the predicted values calculated in the prediction phase. The generation of this trajectory is the most time consuming part of the computation of the RHC inputs. The actual values of the states available at the next sampling period can be used to modify this pre-calculated trajectory using neighbouring extremal paths theory. This constitutes the third phase, *i.e.* the correction phase. The modification can be assumed to have zero computation time, even in systems with fast dynamics such as aerospace systems, as discussed later in this section.

Using our proposed method, the solution to the open loop optimal control problem obtained by assuming zero computation time can be recovered as long as the conditions put forth in section 4.4 are satisfied. The method proposed in this chapter allows for use of RHC in practical systems, while the theoretical results assuming zero computation time for uncertain nonlinear systems still applicable to such practical systems; a property that does not hold for the method presented in [44] as the actual solution to the optimal control problem is not obtained due to the state prediction errors. To illustrate the new approach, it is applied to a mobile robot of unicycle type. Simulations of the system, presented in 4.5, show considerable improvement in performance compared to the existing method found in [44].

2 . Background Material

2.1 Review of Optimal Control

In this section a brief review of classical optimal control will be given. We confine our attention to continues-time systems. The interested reader should refer to [47] for more details on the materials presented in the next sub-sections and a discussion on multi-stage systems.

2.1.1 Open-loop Optimal Control Problem

An optimal control problem is defined as follows:

Consider the following system of differential equations

$$\dot{x}(t) = f\big(x(t), u(t)\big) \qquad t \geq t_0 \tag{1}$$

where $x(t_0)$ is given, $x(t)$ is the state of the system and $u(t)$ is its input. Find a control law $u(t)$ such that the following performance index (scalar) is minimized

$$J = \varphi\big(x(t_f), t_f\big) + \int_{t_0}^{t_f} L\big(x(t), u(t), t\big)dt \tag{2}$$

Depending on the problem, they may be constraints on t_f, the optimization horizon, states and/or input variables. Tools from *calculus of variations* are being used to solve such a problem. A standard approach is to introduce a set of co-state variables $\lambda_i(t)$, and form the Hamiltonian of the system defined as follows:

$$H(x(t),u(t),\lambda(t),t) = L(x(t),u(t),t) + \lambda^T f(x(t),u(t),t) \tag{3}$$

As a *necessary condition* for optimality a set of differential equations subject to boundary values should be solve which is summarized in the following sub-sections.

2.1.1.1 No terminal constraints, fixed terminal time

To find the control vector $u(t)$, that produces a stationary value of the performance index J, we must solve the following differential equations:

$$\begin{aligned} \dot{x} &= f(x,u,t) \\ \dot{\lambda} &= -\left(\frac{\partial f}{\partial x}\right)^T \lambda - \left(\frac{\partial L}{\partial x}\right)^T \end{aligned} \tag{4}$$

Where $u(t)$ is determined by

$$\frac{\partial H}{\partial u} = 0 \qquad or \qquad \left(\frac{\partial f}{\partial u}\right)^T \lambda + \left(\frac{\partial L}{\partial u}\right)^T = 0 \tag{5}$$

Subject to the boundary conditions

$$
\begin{aligned}
&x(t_0) \quad given \\
&\lambda(t_f) = \left(\frac{\partial \varphi}{\partial x}\right)^T_{t=t_f}
\end{aligned}
\tag{6}
$$

The above problem is referred to as a Two Point Boundary Value Problem (TPBVP). In the next parts of this section, we will discuss on numerical methods to solve such problems.

2.1.1.2 Functions of the state variables prescribed at a fixed terminal time

Consider the system (1) and the performance index (2) with q additional constraints on the state variables at the terminal time t_f as

$$
\psi\left(x(t_f), t_f\right) = 0 \tag{7}
$$

The TPBVP is as follows

$$
\begin{aligned}
&\dot{x} = f(x, u, t) \\
&\dot{\lambda} = -\left(\frac{\partial f}{\partial x}\right)^T \lambda - \left(\frac{\partial L}{\partial x}\right) \\
&\frac{\partial H}{\partial u} = \left(\frac{\partial f}{\partial u}\right)^T \lambda + \left(\frac{\partial L}{\partial u}\right)^T = 0
\end{aligned}
\tag{8}
$$

$x_k(t_0)$ given or $\lambda_k(t_0)=0,\ k=1,\quad ,n$ as n boundary conditions as well as

$$\lambda^T(t_f) = \left(\frac{\partial \varphi}{\partial x} + v^T \frac{\partial \psi}{\partial x}\right)_{t=t_f} \tag{9}$$

With q side conditions as

$$\psi\left(x(t_f), t_f\right) = 0 \tag{10}$$

In (9) v are parameters introduced as a result of constraints on the end points to be determined by the set of equations mentioned above.

2.1.2 Numerical Methods for Solving TPBVP

In this sub-section, our objective is to review numerical algorithms to solve a TPBVP, namely, a set of differential equations

$$\dot{y}(x) = f\left(x, y(x)\right) \qquad a < x < b \tag{11}$$

And a set of boundary conditions

$$g\left(y(a), y(b)\right) = 0 \tag{12}$$

Among different methods we briefly review the Shooting Method and the Collocation Method.

2.1.2.1 Shooting Method

As there are numerous efficient algorithms to solve Initial Value Problems (IVP), a possible way of solving numerically a TPBVP is to transform it to a IVP and an algebraic equation. Consider the system (11) along with the following initial value conditions

$$y(a;s)=s \tag{13}$$

With s being an unknown set of parameters. Now the problem transforms into solving for s in

$$g(s, y(b;s))=0 \tag{14}$$

Equation (14) can be solved using any method for solving a (nonlinear) algebraic equation (*e.g.* Newton's Method). Note that each function evaluation of (14) requires solving an IVP.

Although this method is easy to understand, it has some disadvantages, *e.g.* the IVPs may be unstable although the BVPs are stable or starting with some initial guesses may lead to solutions partly defined on the interval $[a,b]$ which causes problems in the implementation of the algorithm. For a detailed discussion of this algorithm refer to [48].

2.1.2.2 Collocation Method

In this method the solution $y(x)$ is approximated by

$$v(x) = \sum_{i=1}^{K} a_i h_i(x) \tag{15}$$

Where $h_i(x)$ is some well-behaved basis function and a_i are parameters to be determined by the following conditions:

- Boundary conditions – $v(x)$ should satisfy the boundary conditions
- Collocation constraints – The approximate solution $v(x)$ satisfies the original differential equation (11) in K-1 points.

If we are not unlucky, we will end up with a unique solution for the mentioned problem and as $K \to \infty$ the approximate solution $v(x)$ will converge to $y(x)$ [49].

2.1.2.3 Example 1: VanDerPol Oscillator

As an example, we will solve the VanDerPol Oscillator problem, defined in [35]. VanDerPol Oscillator is a classical example in nonlinear system theory. For a fundamental introduction refer to [50].

Consider the system and its initial conditions

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= -x_1 + (1 - x_1^2)x_2 + u \\
x_1(0) &= 1 \\
x_2(0) &= 0
\end{aligned} \tag{16}$$

Subject to a performance index and endpoint condition of the form

$$\begin{aligned}
J(u) &= \frac{1}{2}\int_0^5 (x_1^2 + x_2^2 + u^2)dt \\
x_2(5) - x_1(5) - 1 &= 0
\end{aligned} \tag{17}$$

As mentioned in 2.1.1.2 the above problem can be transformed into the following TPBVP

$$
\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= -x_1 + (1 - x_1^2)x_2 + u \\
\dot{\lambda}_1 &= (1 + 2x_1 x_2)\lambda_2 - 2x_1 \\
\dot{\lambda}_2 &= -\lambda_1 - (1 - x_1^2)\lambda_2 - 2x_2 \\
u &= -\frac{\lambda_2}{2} \\
x_1(0) &= 1 \\
x_2(0) &= 0 \\
\lambda_1(5) &= -v \\
\lambda_2(5) &= v \\
x_2(5) - x_1(5) - 1 &= 0
\end{aligned}
\tag{18}
$$

The following solution was obtained using MATLAB which solves TPBVPs using collocation algorithm (bvp4c.m).

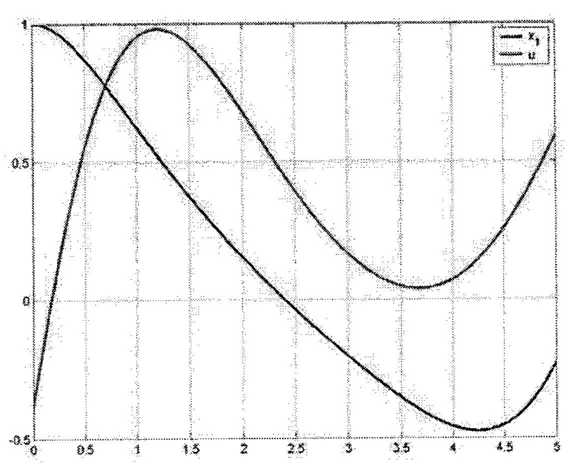

Figure 1. Open-loop control of the VanDerPol Oscillator

2.1.2.4 An Alternative Approach For Solving Open-loop Optimal Control Problems

Milam et. al. in [35] has suggested the following approach to solve the open-loop optimal control problem:

1. Transforming the original set of under-determined differential equations to the lowest dimension possible by choosing the appropriate set of outputs (possibly flat outputs – a brief discussion on flat outputs is given in the next section).
2. Parametrizing the selected outputs using B-Spline basis functions.
3. Reformulating the original constrained problem with the new parametrization. This results in an optimization problem, where the objective is to find the unknown parameters introduced by B-Splines such that the performance index is minimized.
4. Using Sequential Quadratic Programming (SQP) to solve the constrained optimization problem.

The main motivation for using such method is its speed of convergence as compared to the other methods (refer to [35] for a detailed comparison).

This method would be especially useful if we would like to solve the optimal control problem on-line.

2.2 Receding Horizon Control

2.2.1 Theoretical details of the RHC

Consider the nonlinear system

$$\dot{x}(t) = f\big(x(t), u(t)\big) \qquad x(0) = x_0 \tag{19}$$

where $x(t) \in R^n$ and $u(t) \in R^m$ are states and inputs of the system, respectively. Assume that the following constraints on input is present:

$$u(t) \in U \tag{20}$$

And the vector field *f(x(t),u(t))* satisfies a set of assumptions (Assumptions A1-A3 in [38]). Moreover, *f(0,0)=0* meaning that the origin in the presence of no control input is an equilibrium point of the system. Now the open-loop optimal control problem (that should be solved repeatedly) is stated as follows:

Problem [38],[56]: Find

$$J^*\big(x(t), T\big) = \min_{u(.)} J\big(x(t), u(.), T\big) \tag{21}$$

with

$$J(x(t),u(.),T)=\int_t^{t+T}\left(\|x(\tau;x(t))\|_Q^2+\|u(\tau)\|_R^2\right)d\tau+\|x(t+T;x(t))\|_P^2 \tag{22}$$

Subject to

$$\left.\begin{array}{l} x(s)=f(x(s),u(s)) \\ u(s)\in U \\ x(s;x(t))\in Z \end{array}\right\} \quad s\in[t,t+T] \tag{23}$$

and

$$x(t+T;x(t))\in\Omega \tag{24}$$

Where $x(\tau;x(t))$ represents the state of the system at time τ if the initial state of the system is *x(t)* and *Q*, *R* and *P* are weighting matrices . Equation (24) is called the terminal constraint and *T* is the *prediction horizon.* If the solution to the open-loop optimal control problem at time *t* and initial state *x(t)* is $u^*(\tau;x(t))$ the receding horizon control law is

$$u^*(\tau)=u^*(\tau;x(t)),\qquad \tau\in[t,t+\delta),\qquad 0<\delta\le T \tag{25}$$

In the above expression δ is the sampling time or execution horizon. After applying the calculated control law for δ time units, the open-loop optimal control problem is solved again with the current state of the system regarded as the initial state in the optimal control problem. This incorporates an implicit feedback into the closed-loop system.

The following diagram describes the mentioned procedure of the RHC method.

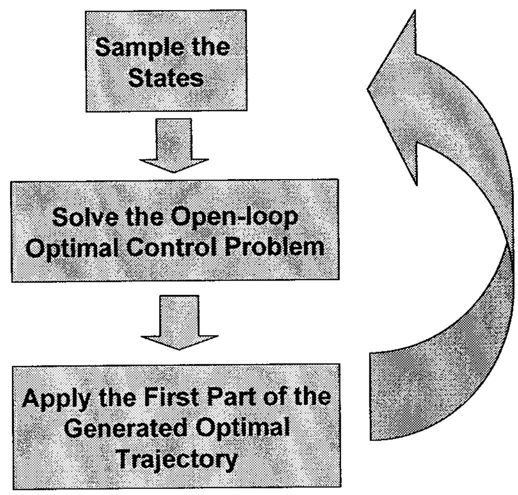

Figure 2. Diagram showing the RHC method

2.3 RHC of a Two-Dimensional Double Integrator

A 2D double integrator is described by the following set of differential equations

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= u_1 \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= u_2 \end{aligned} \tag{26}$$

This system is also known as "point mass system" since it represents a point mass with mass of unity. We choose the following performance index for this system

$$J(x(t),u(.),T) = \int_t^{t+T} \left(\|x(\tau;x(t))\|_Q^2 + \|u(\tau)\|_R^2 \right) d\tau + \|x(t+T;x(t))\|_P^2 \tag{27}$$

In this example we consider the unconstrained problem. We choose the following weighting matrixes for the definition of the performance index (28).

$$Q = I_{4x4} \qquad R = I_{2x2} \tag{29}$$

Following the procedure briefly discussed above Ω happens to be the entire state space (since the system is linear and the problem is unconstrained - a point-mass system is a simple case of a general nonlinear system) and the weighting matrix P is

$$P = \begin{bmatrix} 5.19 & -1 & 0 & 0 \\ -1 & -1.73 & 0 & 0 \\ 0 & 0 & 5.19 & -1 \\ 0 & 0 & -1 & -1.73 \end{bmatrix} \tag{30}$$

The simulation is done for δ=*0.5* and T=*5*. The figure shows that the system is asymptotically stable, as guaranteed by the method mentioned above.

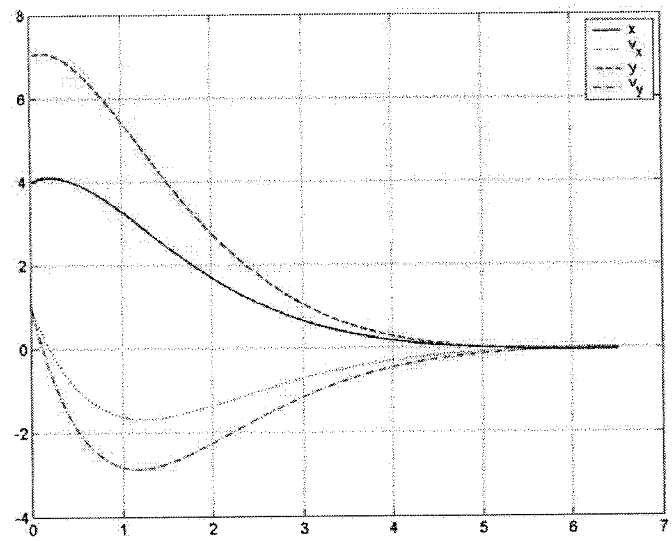

Figure 3. State trajectory of a point-mass system with RHC scheme

A snap-shot view of the particle moving in two dimensional space is shown in the next figure.

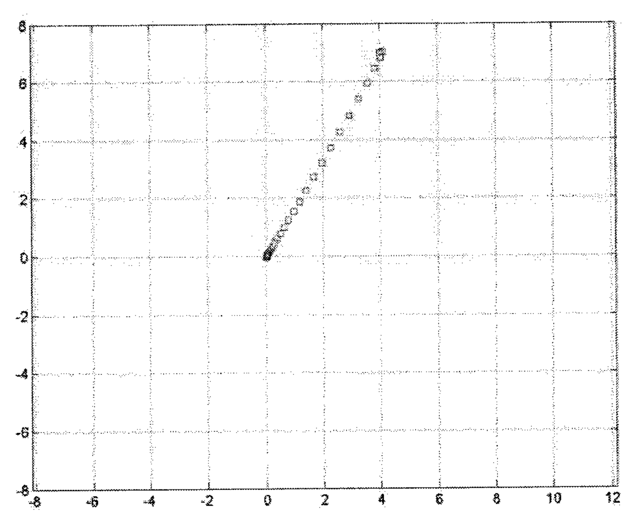

Figure 4. Snap-shot view of the point-mass in 2D space

2.4 Differentially flat systems

Underdetermined systems of Ordinary Differential Equations (ODEs) are a set of differential equations where number of dependent variables exceeds the number of equations. Most of physical and engineering models are considered as underdetermined, where the influence of the environment (*e.g.* forces, torques,...) are incorporated in the model by some additional dependent variables.

Consider the system

$$F^j(t, x, x^{(1)}, \ldots, x^{(k)}) = 0 \qquad j = 1, \ldots, N - p \tag{31}$$

Where F^j are assumed to be C^∞-smooth functions, $x = (x^1, \ldots, x^N) \in R^n$ are the dependent variables, t is the independent variable (usually time), $x^{(r)}$ stands for the rth time derivative of x and p (≥ 1) the number of equations by which the systems is underdetermined. By choosing any p dependent variables and assigning an arbitrary function to them, all other variables can be found by solving the equations (31). Therefore, the solution to the system is parametrized by p functions and a number of integration constants, depending on the order of the system.

An interesting question found in [51] is: "*Can one find a set of outputs (function of dependent variables and their derivatives) so that by assigning arbitrary functions to those outputs, all the other dependent variables can be recovered without the presence of the integration constants?*" This gives rise to the definition of the "flat outputs".

Definition [51]: The system given by (31) is said to be *differentially flat* or simply *flat* if there exists variables $y^1,...,y^p$ given by an equation of the form

$$y = h(t, x, x^{(1)},..., x^{(m)}) \tag{32}$$

such that the original variables x may be recovered from y (locally) by an equation of the form

$$x = g(t, y, y^{(1)},..., y^{(l)}) \tag{33}$$

The variables $y^1,...,y^p$ are referred to as the *flat outputs.*

In this way the original control problem can be tackled in a lower dimensional space (equal to the number of flat outputs) where there is a one-to-one correspondence between each solution in the lower dimensional space and the original space. This property has been used in the last section, for solving an optimal control problem with higher speed. For a detailed discussion on *flatness* refer to [51].

3 . Quasi-Infinite Receding Horizon Control of Systems with Bounded Perturbation

In this section the stability robustness of quasi-infinite RHC is discussed. This results can also be found in [52].

3.1 Quasi-Infinite Receding Horizon of Nonlinear Systems

In this section the scheme proposed in [38] is reviewed. The class of systems considered is described by the set of equations

$$\dot{x} = f(x(t), u(t)) \qquad x(0) = x_0 \tag{34}$$

where $x(t) \in \Re^n$ is the state of the system and $u(t) \in \Re^m$ is the input vector satisfying the constraints

$$u(t) \in U \qquad \forall t \geq 0 \tag{35}$$

U is the set of allowable input values. Furthermore, assume that the set of assumptions (A1-A3) in [38] is also satisfied; that is, f is twice differentiable, U is compact and convex, and the system (34) has a unique solution for a given initial condition. Receding horizon control is essentially the repeated solution of the following problem.

Problem 1 Find

$$J_T^*(x(t)) = \min_{u(.)} J(x(t), u(.), T) \tag{36}$$

with

$$J(x(t), u(.), T) = \int_t^{t+T} \left(\|x(\tau; x(t))\|_{\mathbf{Q}}^2 + \|u(\tau)\|_{\mathbf{R}}^2 \right) d\tau + \|x(t+T; x(t))\|_{\mathbf{P}}^2 \tag{37}$$

subject to

$$\left.\begin{aligned} \dot{x}(s) = f(x(s), u(s)) \\ u(s) \in U \end{aligned}\right\} \quad s \in [t, t+T] \tag{38}$$

$$\begin{aligned} & x(t+T; x(t)) \in \Omega_\alpha \\ & \Omega_\alpha := \left\{ x \in \Re^n \middle| x^T \mathbf{P} x \le \alpha \right\} \end{aligned} \tag{39}$$

$\mathbf{Q} \in \Re^{n \times n}$ and $\mathbf{R} \in \Re^{m \times m}$ denote positive-definite, symmetric weighting matrixes by which the states and the inputs can be penalized, T is a finite prediction time and $x(t; x_0)$ denotes the trajectory of the system (1) driven by $u(t)$ starting from the initial condition x_0. Furthermore, the weighted norms in Eq. (37) are defined as $\|x\|_{\mathbf{P}}^2 = x^T \mathbf{P} x$.

Let δ denote the receding horizon sampling period, where δ lies in the $(0,T]$ interval. The closed loop system is described by

$$\dot{x}(\tau) = f\left(x(\tau), u^*(\tau)\right)$$
$$u^*(\tau) = u^*(\tau; x(t)) \qquad \tau \in [t, t+\delta), \quad 0 < \delta \leq T \tag{40}$$

where $u^*(\tau; x(t)), \quad \tau \in [t, t+T]$ is the optimal control of the problem stated above with the initial condition *x(t)*, *t* being the time of start of the optimization process and the instant at which states are sampled.

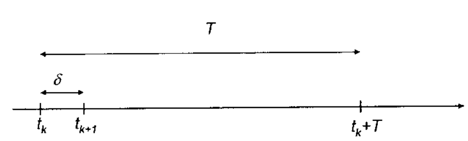

Figure 5. A Schematic diagram showing the execution and prediction horizon

3.2 Robustness Analysis

In reference [36] a brief robustness analysis of the RHC scheme proposed in [33] was discussed, where a ball of specified radius was found where the closed loop system is stable as long as the states are outside the ball. Such analysis is based on a first-order approximation of the incremental cost in the *[0, δ]* interval. This result requires a very small receding horizon update time, *δ*, to be accurate due to the order of the approximation. In the present section, we follow the methodology in [36], although we propose to use a more accurate approximation to the incremental cost function, to deal with the RHC with a quadratic cost function presented in [38] and derive a sufficient condition for the stability of the closed loop system. The higher-order approximation makes the choice of δ less conservative, hence allowing larger values to be used, namely up to 2 times the order of magnitude when compared with the method presented in [36]. We later specialize our results to the linear case in Section 2.4.

Consider the system

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \tag{41}$$

which is used as a nominal model in the RHC synthesis and

$$\dot{y} = \boldsymbol{f}(y, \boldsymbol{u}) + \boldsymbol{g}(t, y, \boldsymbol{u}) \tag{42}$$

which serves as the model of the real system. In Eq. (42), $\boldsymbol{g}(t,y,\boldsymbol{u})$ accounts for disturbances and modeling uncertainty and its 2-norm is upper bounded by b and $\boldsymbol{f}$ in Eqs. (41) and (42) is a Lipschitz continuous function. It can be shown that for sufficiently small receding horizon updates, δ, and with a RHC technique as described in Section 2.2, the closed loop system (40) is stable in the Lyapunov sense, provided that some nonlinear inequality is satisfied (49).

According to Theorem 1 in [38], the following inequality holds for the nominal system

$$J_T^*(x(t_k+\delta)) \leq J_T^*(x(t_k)) - Q_\delta(x(t_k)) \tag{43}$$

where

$$Q_\delta(x(t_k)) = \int_0^\delta \left\| x^*(\tau; x(t_k)) \right\|_Q^2 d\tau \tag{44}$$

and $x^*(\tau; x(t_k))$ and $u^*(\tau; x(t_k))$ are the optimal values of the states and control inputs for the open loop optimal control problem with initial condition $x(t_k)$.

On the other hand, the following inequality is a direct consequence of the Bellman-Grownwall Lemma [36]

$$\left\| y(t_k+\delta) - x(t_k+\delta) \right\| \le e^{L\delta} \left\| y(t_k) - x(t_k) \right\| + \frac{b}{L}\left(e^{L\delta} - 1\right) \tag{45}$$

where L is the Lipschitz constant of the function *f*. Since $J_T^*(.)$ is Lipschitz continuous with a Lipschitz constant of say *K*, the following inequality holds

$$J_T^*(y(t_k+\delta)) \le J_T^*(y(t_k)) - Q_\delta(x(t_k)) + K\frac{b}{L}\left(e^{L\delta} - 1\right) \tag{46}$$

Note that $x(t_k) = y(t_k)$ due to the zero computation time assumption. To prove the stability of the closed loop system, we take $J_T^*(.)$ as the discrete time Lyapunov function of the sampled RHC system and it is straightforward to verify that the following inequality is a sufficient condition for the stability of the closed loop system.

$$-Q_\delta(x(t_k)) + K\frac{b}{L}\left(e^{L\delta}-1\right) \le 0 \tag{47}$$

We take $\delta^3 \approx 0$ since δ is sufficiently small. Using Taylor series expansion we have

$$\boldsymbol{x}^*(t;\boldsymbol{x}(t_k)) = \boldsymbol{x}^*(t_k;\boldsymbol{x}(t_k)) + \dot{\boldsymbol{x}}^*(t_k;\boldsymbol{x}(t_k))t + \ddot{\boldsymbol{x}}^*(t_k;\boldsymbol{x}(t_k))\frac{t^2}{2} + \mathrm{O}\left(t^3\right) \tag{48}$$

It is straightforward to verify that the set of all points in the state space which satisfy the following inequality serve as the region of attraction for the nonlinear uncertain system represented by Eq. (9).

$$\boldsymbol{x}^T\mathbf{Q}\boldsymbol{x} + \delta\, \boldsymbol{f}^T(\boldsymbol{x})\mathbf{Q}\boldsymbol{x} \ge 2Kb \tag{49}$$

Note that *f(x)* represents the dynamics of the closed loop system where ***u=u(x)*** and we have used the following inequality, which holds for $\delta < 1$ [36]

$$K\frac{b}{L}\left(e^{L\delta}-1\right) \le 2Kb\delta \tag{50}$$

3.3 Robustness Analysis for Linear Systems

There is no guarantee of boundedness for the set of points for which equation (51) defined in Section 2.3 does not hold for the general case. However, such set can be bounded in the case of linear systems. In this section we show that the closed loop system is stable outside a bounded set which contains the desired equilibrium point provided an LMI constraint is satisfied. Equivalently, by introduction of perturbations, the closed loop system will be stable inside a bounded set rather than a single point. Before stating the main result of the chapter, in the form of Theorem 1, we need to state the following lemmas and assumption.

Lemma 1. Consider the ellipsoid, centered at the origin, defined by

$$\boldsymbol{x}^T\mathbf{H}\boldsymbol{x} = 1 \tag{52}$$

where $\boldsymbol{x} \in \Re^n$ and $\mathbf{H} = \mathbf{H}^T$ is a $n \times n$ positive definite matrix. This ellipsoid is contained in a ball, centered at the origin, of radius $\sqrt{\left|\lambda_{\min}^{-1}(\mathbf{H})\right|}$, where $\lambda_{min}(\mathbf{H})$ is the minimum absolute value of the eigenvalues of the matrix $\mathbf{H}$.

Proof. The length of the semi-axis of an ellipsoid is given by $\sqrt{\lambda_i^{-1}}$ where λ_i is the i^{th} eigenvalue of the matrix $\mathbf{H}$ [10]. Therefore

$$\max_{x \in E} \|\boldsymbol{x}\| = \sqrt{\left|\lambda_{\min}^{-1}(\mathbf{H})\right|} \tag{53}$$

where $E = \left\{ \boldsymbol{x} \in \Re^n \middle| \boldsymbol{x}^T\mathbf{H}\boldsymbol{x} = 1 \right\}$. $\square$

Lemma 2. Consider the set Ω, as the collection of ellipsoidal regions defined by

$$\Omega = \left\{ \boldsymbol{x} \in \Re^n \middle| (\boldsymbol{x}-\boldsymbol{x}_c)^T \mathbf{H} (\boldsymbol{x}-\boldsymbol{x}_c) \leq 1, \quad \boldsymbol{x}_c \in X_c \right\} \tag{54}$$

where $\mathbf{H}$ is an n-dimensional square matrix and X_c is a bounded set in $\Re^n$. Then Ω is bounded.

Proof. To show that Ω is bounded, we will find a ball, B_r, centered at the origin that contains Ω. Note that since Ω is a collection of ellipsoids, the point having the largest distance from the origin lies on the boundary of one of the ellipsoids. Furthermore, the coordinates of a point on the boundary of an ellipsoid, $\boldsymbol{x}_b$, can be found by the addition of two vectors: A vector connecting the origin to the center of the corresponding ellipsoid, $\boldsymbol{x}_c$, and the vector connecting the center of the ellipsoid to the corresponding point, $\boldsymbol{x}_e$, which satisfies $\boldsymbol{x}_e^T \mathbf{H} \boldsymbol{x}_e = 1$.

Now take

$$r_1 = \sup_{\boldsymbol{x} \in \Omega} \|\boldsymbol{x}\| = \sup \|\boldsymbol{x}_c + \boldsymbol{x}_e\| \tag{55}$$

Using the Schwartz's inequality and the *sup* property,

$$r_1 = \sup_{\boldsymbol{x}_c, \boldsymbol{x}_e} \|\boldsymbol{x}_c + \boldsymbol{x}_e\| \leq \sup_{\boldsymbol{x}_c} \|\boldsymbol{x}_c\| + \sup_{\boldsymbol{x}_e} \|\boldsymbol{x}_e\| \tag{56}$$

As X_c is bounded with a radius of say R, using Lemma 1 we can conclude that

$$\sup_{\boldsymbol{x}_c} \|\boldsymbol{x}_c\| + \sup_{\boldsymbol{x}_e} \|\boldsymbol{x}_e\| = R + \sqrt{\lambda_{\min}^{-1}(\mathbf{H})} \tag{57}$$

The radius of B_r can be found as follows

$$r = R + \sqrt{\lambda_{\min}^{-1}(\mathbf{H})} \tag{58}$$

□

Assumption 1. Assume a zero computational time for the on-line optimization algorithm, solving Problem 1 in Section 2.2.

Theorem 1. Consider the receding horizon control of an uncertain system defined by

$$\dot{y} = \mathbf{A}y + \mathbf{B}\boldsymbol{u} + g(t, y, \boldsymbol{u}) \tag{59}$$

where the nominal LTI system

$$\dot{x} = \mathbf{A}x + \mathbf{B}\boldsymbol{u} \tag{60}$$

is used for controller synthesis. For a small enough receding horizon update period, δ, the closed loop system (40) is stable while the states are outside the bounded set Ω containing the origin, provided that

$$\|g(t, y, \boldsymbol{u})\| \leq b \tag{61}$$

and the following LMI constraint holds

$$\mathbf{Q} + \frac{\delta}{2}\mathbf{QA} + \frac{\delta}{2}\mathbf{A}^T\mathbf{Q} > 0 \tag{62}$$

where b is a positive constant and $\mathbf{Q}$ is the weighting matrix in the optimal control problem.

Proof. Using Eqs. (48) and (60), and rearranging terms we end up with the following equation

$$x^*(t;x(t_k)) = (\mathbf{I} + t\mathbf{A} + \frac{t^2}{2}\mathbf{A}^2)x_0^* + (t\mathbf{B} + \frac{t^2}{2}\mathbf{AB})u_0^* + \frac{t^2}{2}\mathbf{B}\dot{u}(t_k;x(t_k)) + O(t^3) \tag{63}$$

Using (63) and the definition of the weighted norm we conclude that

$$\int_0^\delta \left(\left\| x^*(\tau;x(t_k)) \right\|_{\mathbf{Q}}^2 \right) d\tau = x_0^{*T} \widetilde{\mathbf{Q}} x_0^* + \delta^2 u_0^{*T} \mathbf{B}^T Q x_0^* + O(\delta^3) \tag{64}$$

where

$$\widetilde{\mathbf{Q}} = \delta\mathbf{Q} + \frac{\delta^2}{2}\mathbf{QA} + \frac{\delta^2}{2}\mathbf{A}^T\mathbf{Q} \tag{65}$$

Using (47), (50) and (64) the closed loop system is stable, while the following condition holds

$$2Kb\delta \leq x_0^{*T} \widetilde{\mathbf{Q}} x_0^* + \delta^2 u_0^{*T} \mathbf{B}^T \mathbf{Q} x_0^* \tag{66}$$

where we have used the following notation for simplicity

$$\begin{aligned} x_0^* &= x^*\left(t_k; x(t_k)\right) \\ u_0^* &= u^*\left(t_k; x(t_k)\right) \end{aligned} \tag{67}$$

As Eq. (66) defines a region in the state space (which is a specific form of the condition (49)) we require this region to be bounded and in the form of an ellipsoid. It is straightforward to show that Eq. (66) defines an ellipsoid with a boundary given by

$$(x - x_c)^T \mathbf{H}(x - x_c) = 1 \tag{68}$$

where

$$x_c = -\frac{\delta^2}{2}\widetilde{\mathbf{Q}}^{-1}\mathbf{QB}u_0^* \tag{69}$$

$$\mathbf{H} = \frac{1}{2Kb}\left(\mathbf{Q} + \frac{\delta}{2}\mathbf{QA} + \frac{\delta}{2}\mathbf{A}^T\mathbf{Q}\right) \tag{70}$$

provided that the $\widetilde{\mathbf{Q}}$ is positive definite or equivalently, the LMI constraint in Eq. (62) holds. Now we define Ω containing the origin, so that while the trajectory of the closed loop system is outside of this set the system is stable.

$$\Omega = \left\{ x \in \Re^n \middle| (x - x_c)^T \mathbf{H}(x - x_c) \leq 1, \quad u_0^* \in U \right\} \tag{71}$$

We conclude the proof by showing that Ω is bounded. As U is a bounded set and regarding the fact that x_c is the image of u_0^* given by the linear transformation (69), using the property of linear transformations we can conclude that the image set X_c is also bounded, *i.e.* it is contained in the ball centered at the origin with a radius R. Now using Lemma 2 the radius of the ball containing Ω is given by

$$r = R + \sqrt{\left|\lambda_{\min}^{-1}(\mathbf{H})\right|} \tag{72}$$

and this concludes the proof. □

Remark 1. Theorem 1 shows that if the linear approximation of a general nonlinear system subject to exogenous disturbances has norm bounded modeling error, the linear model can still be successfully used for the controller design purpose, while RHC controller stabilizes the actual nonlinear system around a bounded set containing the desired equilibrium point.

3.4 Example

In this section, we apply the previous developed results to the point stabilization of a differentially driven wheeled mobile robot, which is especially useful in formation stabilization. For this problem it is desired that each agent take a predefined position [11]. A mobile robot of unicycle type is described by the following set of kinematic equations

$$\begin{aligned} \dot{x} &= v\cos\theta \\ \dot{y} &= v\sin\theta \\ \dot{\theta} &= \omega \end{aligned} \tag{73}$$

where x and y are the coordinates of a point located at the mid-axis of the rear wheels of the robot, θ is the heading angle of the robot with respect to the positive x-axis and v and ω are the linear and angular velocity of the robot, respectively. The dynamic equations of the mobile robot are described by

$$\begin{aligned} M\dot{v} &= F \\ J\dot{\omega} &= \tau \end{aligned} \tag{74}$$

where F and τ represent the force and torque exerted on the robot (control inputs), respectively, and M and J are the mass and moment of inertia of the robot, respectively.

A schematic diagram of a mobile robot is shown in Figure 1. This set of equations, which can also serve as a description for a rotorcraft-like UAV flying at constant altitude (see Figure 2), can be transformed into a two-dimensional double integrator using feedback linearization [40], [41] and [42].

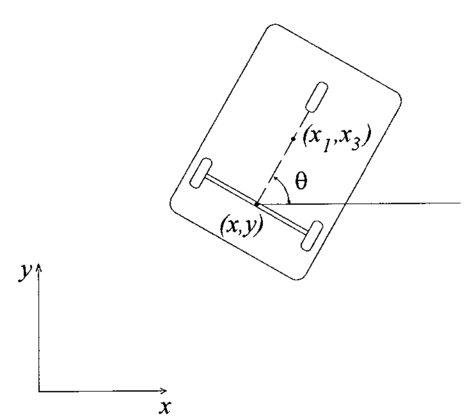

Figure 6. A schematic diagram of a mobile robot of unicycle type

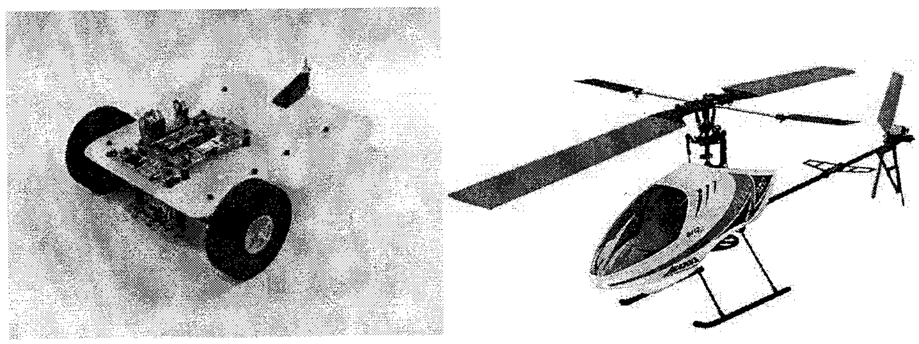

Figure 7. Lynxmotion's Carpet Rover and Piccolo Pro helicopter facilities in the CIS Lab.

We consider the coordinates of a point off the center of the wheel axis, (x_1, x_3), as the output (*e.g.* center of mass of the robot, see Figure 2). Following a series of manipulations, we end up with the following system (refer to the appendix, Section 0)

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= u_1 \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= u_2 \end{aligned} \tag{75}$$

where the relationship between the new inputs and the actual inputs to the system is given by

$$\begin{bmatrix} \frac{1}{m} F \\ \frac{L_a}{J} \tau \end{bmatrix} = \begin{bmatrix} L_a \omega^2 \\ -v\omega \end{bmatrix} + \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}^{-1} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{76}$$

In the above equation L_a is the distance from the middle of the wheel axis of the robot to the chosen point (x_1,x_3). The RHC of a two-dimensional double integrator is described in more detail in the appendix, Section 2.3.

We take the actual uncertain system under control as

$$\begin{aligned}
\dot{x}_1 &= x_2 + \alpha_1 \\
\dot{x}_2 &= u_1 + \alpha_2 \\
\dot{x}_3 &= x_4 + \alpha_3 \\
\dot{x}_4 &= u_2 + \alpha_4
\end{aligned} \tag{77}$$

where $\alpha_i \in [-1,1]$, i=1,2,3,4, are random variables, whereas the initial conditions are chosen to be $\boldsymbol{x_0}=[5\ -1\ -6\ 0]^{\mathrm{T}}$. We use the system described by Eq. (75) as the nominal system for the RHC synthesis and choose the execution horizon and prediction horizon to be 0.1 and 5 seconds, respectively. Weighting matrices **Q** and **R** are chosen as the identity matrices of appropriate dimension. Matrix **P** in (37) is found by solving the Lyapunov equation, Eq. (9) in [38]. The set of allowable input signals U is defined by

$$U = \left\{ \boldsymbol{u} = [u_1 \quad u_2]^T \in \Re^2 \middle| -1 \leq u_i \leq 1 \quad i = 1, 2 \right\} \tag{78}$$

Using Theorem 1, it is straightforward to verify that the LMI inequality constraint (62) is satisfied and that the perturbation term is bounded, namely b=2. Furthermore, we assume that δ^3=0.001 is negligible, compared to the parameters chosen above.

To implement the RHC technique, we use the direct method adopted from [35]. Note that x_1 and x_3 are the flat outputs of this system and by choosing them as the optimization variables we can reduce the dimension of the state space considerably. The flatness property and its use in control systems are addressed in [43]. We use third-order continuous piecewise polynomials to approximate the optimum outputs. The interval [0,5] has been divided into 10 intervals, which are second-order continuous at the boundaries. Since third-order polynomials have been used to approximate the outputs, the inputs, which are the second-order derivative of the outputs will be piecewise linear as shown in the figures.

As can be seen from Figures 1 and 2, RHC was able to drive the states of the system to the vicinity of the origin and prevent further deviation from this point in the presence of a stochastic perturbation in the state equations, as guaranteed by the theory.

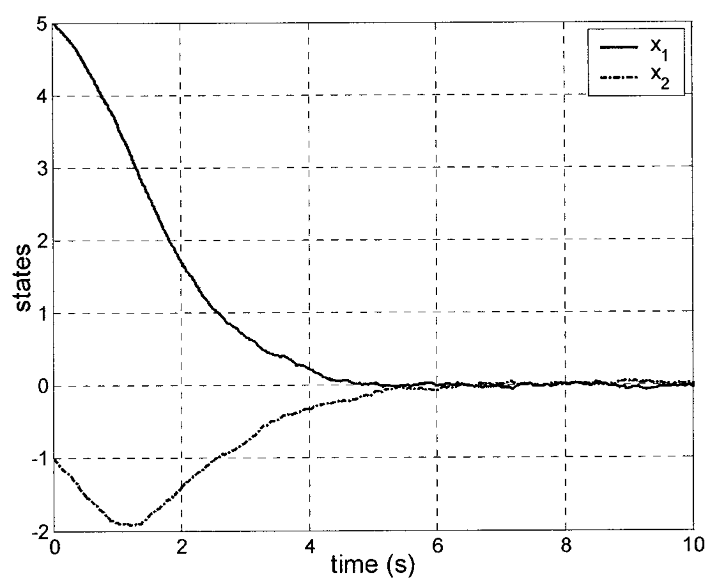

Figure 8. Time history of x_1 and x_2

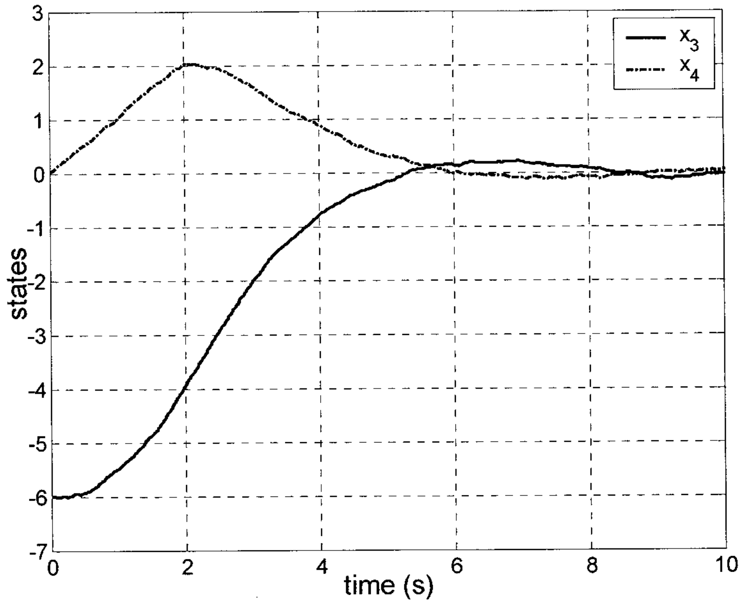

Figure 9. Time history of x_3 and x_4

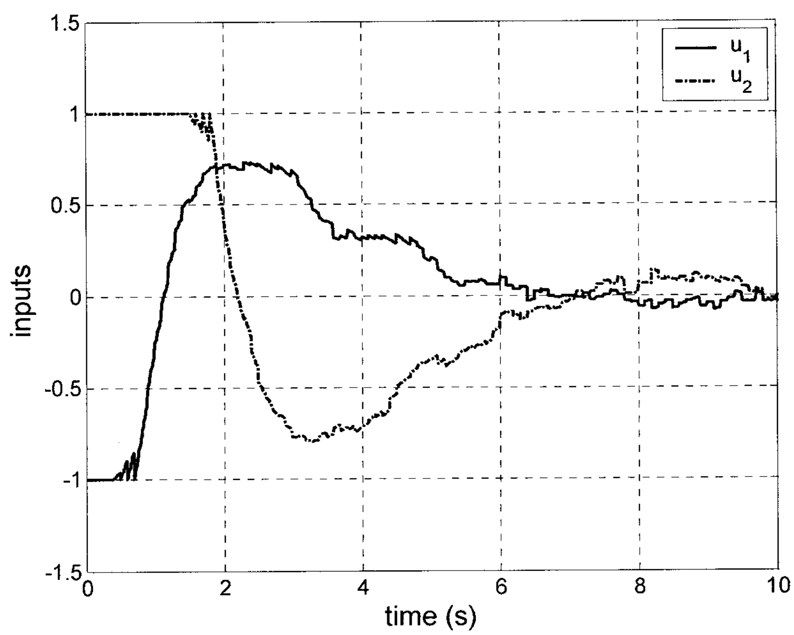

Figure 10. Time history of inputs u_1 and u_2

4 . Receding Horizon Control of Uncertain Nonlinear Systems Subject to Computational Delay

In this section, we discuss the RHC where the computation time is non-negligible. The results may also be found in [53].

4.1 Receding Horizon Control of Nonlinear Systems

In this section, we review the general RHC scheme briefly. Although the quasi-infinite RHC was discussed in Section 3.1, we would like to discuss the RHC in general in this chapter and will not limit ourselves to quasi-infinite RHC.

The class of systems considered is described by the set of equations

$$\dot{x} = f\big(x(t), u(t)\big) \qquad x(0) = x_0 \tag{79}$$

where $x(t) \in \Re^n$ is the state of the system and $u(t) \in \Re^m$ is the input vector satisfying the constraints

$$u(t) \in U \qquad \forall t \geq 0 \tag{80}$$

U is the set of allowable inputs. Furthermore, we assume that assumptions (A1-A3) in [38] are also satisfied; that is, f is twice differentiable, U is compact and convex, and system (79) has a unique solution for a given initial condition. Receding horizon control is the repeated solution of the following problem.

Problem 1 Find

$$J_T^*\big(x(t)\big) = \min_{u(.)} J(x(t), u(.), T) \tag{81}$$

with

$$J(x(t), u(.), T) = \int_t^{t+T} \Big(\|x(\tau; x(t))\|_{\mathbf{Q}}^2 + \|u(\tau)\|_{\mathbf{R}}^2 \Big) d\tau + \|x(t+T; x(t))\|_P^2 \tag{82}$$

subject to

$$\left.\begin{aligned} \dot{x}(s) &= f\big(x(s), u(s)\big) \\ u(s) &\in U \end{aligned}\right\} \quad s \in [t, t+T] \tag{83}$$

$\mathbf{Q} \in \Re^{n \times n}$ and $\mathbf{R} \in \Re^{m \times m}$ denote positive-definite, symmetric weighting matrices, T is a finite prediction time and $x(t; x_0)$ denotes the trajectory of the system (1) driven by $u(t)$ starting from the initial condition x_0. Furthermore, the weighted norms in (37) are defined as $\|x\|_{\mathbf{Q}}^2 = x^T \mathbf{Q} x$.

Let h denote the receding horizon sampling period, where h lies in the $(0,T]$ interval. The closed-loop system is described by

$$\begin{aligned} &\dot{x}(\tau) = f\big(x(\tau), u^*(\tau)\big) \\ &u^*(\tau) = u^*(\tau; x(t)) \qquad \tau \in [t, t+h), \quad 0 < h \le T \end{aligned} \tag{84}$$

where $u^*(\tau; x(t)), \quad \tau \in [t, t+T]$, is the optimal control of the problem stated above with the initial condition *x(t)*, *t* being the start time of the optimisation process and the instant at which states are sampled.

As discussed in [31], numerous methods have been suggested to guarantee the stability of closed-loop system by requiring a terminal constraint at the end-time of the optimization horizon or a special way to select the terminal cost. Therefore, it is straightforward to adapt the RHC scheme to the specific method, one would like to implement. As an instance, [38] guarantees the stability of the closed-loop system provided that the following terminal inequality constraint is added to Problem 1

$$\begin{aligned} & x(t+T; x(t)) \in \Omega_\alpha \\ & \Omega_\alpha := \left\{ x \in \Re^n \middle| x^T \mathbf{P} x \leq \alpha \right\} \end{aligned} \tag{85}$$

where α is a positive constant and the matrix **P**, the solution to the Lyapunov equation is selected as described in [38]. This method, known as quasi-infinite RHC, was discussed in Chapter 2.

4.2 Neighbouring Extremal Paths

In this section, we briefly review the perturbation analysis of the open loop optimal control problem presented in [47]. This will be used in Section 3.4, where we state our proposed solution for dealing with uncertainties in the presence of computational delay.

Assume that Problem 1 in Section 3.2 is solved for the given initial conditions $\boldsymbol{x}(t_0)$. Introduction of a small perturbation in the initial conditions, $\delta \boldsymbol{x}_0$, will cause a change in the optimal trajectories, *i.e.* $\delta \boldsymbol{x}$ and $\delta \boldsymbol{u}$. The solution to the perturbed problem can be retrieved by solving a linear optimal control problem. More specifically, this problem is composed of finding the optimal change in the input signal $\delta \boldsymbol{u}$, minimizing

$$\delta^2 J = \frac{1}{2}\left(\delta \boldsymbol{x}^T \Phi_{\boldsymbol{xx}} \delta \boldsymbol{x}\right)_{t=T} + \frac{1}{2}\int_t^{t+T} \begin{bmatrix} \delta \boldsymbol{x}^T & \delta \boldsymbol{u}^T \end{bmatrix} \begin{bmatrix} H_{\boldsymbol{xx}} & H_{\boldsymbol{xu}} \\ H_{\boldsymbol{ux}} & H_{\boldsymbol{uu}} \end{bmatrix} \begin{bmatrix} \delta \boldsymbol{x} \\ \delta \boldsymbol{u} \end{bmatrix} d\tau \tag{86}$$

subject to the following constraint

$$\begin{aligned} \delta \dot{\boldsymbol{x}} &= \boldsymbol{f}_x \delta \boldsymbol{x} + \boldsymbol{f}_u \delta \boldsymbol{u} \\ \delta \boldsymbol{x}(t_0) &= \delta \boldsymbol{x}_0 \end{aligned} \tag{87}$$

with Hamiltonian described by

$$H = \left\|\boldsymbol{x}(\tau; \boldsymbol{x}(t))\right\|_{\mathbf{Q}}^2 + \left\|\boldsymbol{u}(\tau)\right\|_{\mathbf{R}}^2 + \boldsymbol{\lambda}^T \boldsymbol{f} \tag{88}$$

where $\boldsymbol{\lambda}$ is the vector of co-state variables. As this problem is a linear optimal control problem, it is straightforward to show that the optimal input is given by

$$\delta \boldsymbol{u}(t) = -H_{\boldsymbol{uu}}^{-1}(H_{\boldsymbol{ux}} \delta x + f_u^T \delta \lambda) \tag{89}$$

where the perturbation in the states and co-states is given by

$$
\begin{aligned}
\delta\dot{x} &= \mathbf{A}(t)\delta x - \mathbf{B}(t)\delta\lambda \\
\delta\dot{\lambda} &= -\mathbf{C}(t)\delta x - \mathbf{A}^T(t)\delta\lambda
\end{aligned} \tag{90}
$$

Matrices $\mathbf{A}(t)$, $\mathbf{B}(t)$ and $\mathbf{C}(t)$ are defined as follows

$$
\begin{aligned}
\mathbf{A}(t) &= f_x - f_u H_{uu}^{-1} H_{ux} \\
\mathbf{B}(t) &= f_u H_{uu}^{-1} f_u^T \\
\mathbf{C}(t) &= H_{xx} - H_{xu} H_{uu}^{-1} H_{ux}
\end{aligned} \tag{91}
$$

Among the possible ways to solve the Two Point Boundary Value Problem (TPBVP) in (90), we choose the backward sweep method described in [47].

Remark 1. Although Problem 1 assumes a quadratic cost for the performance index J in (37), in the perturbation analysis, the theory of neighbouring extremal paths is general enough to be applied to optimal control problems with a nonlinear cost in the performance index. This allows its application to RHC schemes such as those described in [33], where the cost in the performance index J is not necessarily quadratic and can have a general nonlinear form.

4.3 Problem Statement and Proposed Method of Solution

Consider the RHC of system (79), as described in Problem 1 in Section 3.2. The optimization problem has to be solved on-line implying that the process of finding the optimal value will require a certain computation time, not known *a priori*. As proposed in [44], the following computation algorithm can be used allowing the RHC scheme to be applied to practical systems: At time t, predict the state of the system at time $t+h$, using the current state values available from the sampling operation. Then, solve the optimal control problem using the predicted states as the initial conditions. This gives the system a computation deadline equal to h to compute the optimal input.

If there is no uncertainty present in the modelling of system (79), the predicted and the actual values of the states are the same. However, uncertainties in the model and exogenous disturbances cause a mismatch in the predicted and the actual values of the states. We propose to modify this pre-computed input before its application to the plant using the theory of neighbouring extremal paths [47] reviewed in Section 3.3. The modification process is composed of two parts: (i) Solving a differential equation by the backward sweep method, a differential Riccati equation resulting from the TPBVP (90); (ii) Solving an initial value differential equation to calculate the changes in the input profile using (89).

The process of modification can be regarded as a zero computation task even in the case of fast dynamic mechanical systems, as it is composed of the solution of two initial value problems for $t \in [0,T]$ corresponding to the first initial value problem and $t \in [0,h]$ for the second one. Note that the parameters of the differential Riccati equation are computed off-line (refer to Chapter 5 in [47]), therefore only the solution of such initial value problems has to be carried out on-line.

The complete algorithm is summarized below.

Algorithm 1. (a) Assume a zero input for the first execution horizon, *i.e.* $\boldsymbol{u}_0=0$. Let $k=0$.

(b) Sample the states at times $t=t_k$

(c) At time t_k, predict the states of the system at time $t=t_{k+1}=t_k+h$ based on the current states and current input $\boldsymbol{u}_k$.

(d) Solve the open loop optimal control problem using the predicted states calculated in step (c). The solution of the optimisation takes place in the time interval $[t_k,t_{k+1}]$ having a computation deadline of h.

(e) Sample the states $\boldsymbol{x}_{actual}$, at $t=t_{k+1}$.

(f) Calculate the difference between the predicted states, $\boldsymbol{x}_{predict}$ and the actual states $\boldsymbol{x}_{actual}$ at $t=t_{k+1}$.

(g) Solve for the change in the optimal input, $\delta\boldsymbol{u}$, and update the input $\boldsymbol{u}_{k+1}$.

(h) $k=k+1$. Goto step (c).

4.4 Validity of the Proposed Algorithm

As the neighbouring extremal paths theory, used in Algorithm 1 in Section 3.4, is only valid in a sufficiently small neighbourhood of the original extremal trajectories, in this section we find a more rigorous description for the condition mentioned above. We show that satisfaction of a condition stated in Proposition 1 is sufficient for the validity of the proposed method for a general nonlinear system subject to bounded perturbations. We require the following assumptions to hold.

Assumption 1. In the open loop optimal control problem described in Problem 1, small perturbation in the initial conditions will result in perturbation in other variables, δx, $\delta \boldsymbol{u}$ and $\delta \boldsymbol{\lambda}$, of the same order, *i.e.*

$$\mathrm{O}(\delta x_0) = \mathrm{O}(\delta x) = \mathrm{O}(\delta \boldsymbol{u}) = \mathrm{O}(\delta \lambda) \tag{92}$$

Assumption 2. The RHC problem defined in Problem 1 has no constraints on the input, *i.e.* $U = \Re^m$, where m is the dimension of the input space. This is due to the fact that the perturbation analysis in Section 3.3 is based on the optimal control problem with no constraints on the input. This assumption can be removed, if the corresponding theory is modified accordingly.

Assumption 3. Among the possible RHC schemes discussed in [31],[33],[38], we choose those for which a terminal constraint is not used to guarantee stability of the closed loop. In order to remove this assumption, the perturbation analysis should be appropriately changed (refer to [47]).

Remark 2. Assumption 3 confines the use of the method described in [38] to linear unconstrained problems. A novel RHC technique to stabilize nonlinear systems has been recently introduced in [33]. The latter does not require a terminal constraint. Therefore, the RHC method described in [33] satisfies Assumption 3.

Assumption 4. The computation time of the parts (f) and (g) of Algorithm 1 are negligible compared to the dynamics of the closed-loop system.

Remark 3. Note that Assumption 4 is only necessary from the practical point of view and not for the validity of the correction phase in Algorithm 1. It is practically achievable, considering the fact that two initial value problems (with the parameters of the differential equations calculated off-line) have to be solved in a time span equal to the execution and prediction horizon, for the corresponding initial value problems. Use of the method described in [45] can even reduce the computation time further. TPBVP can be avoided using the pseudospectral approximation. Instead, a set of coupled algebraic equations should be solved on-line, reducing the computation burden significantly.

Assumption 5. The computation time required to solve the optimal control problem in step (d) is less than the execution horizon.

Proposition 1. Consider the system

$$\dot{x} = f(x, u) \tag{93}$$

which is used as a nominal model for the RHC synthesis and

$$\dot{x}_{actual} = f(x_{actual}, u) + g(t, x_{actual}, u) \tag{94}$$

which serves as the model of the real system. In (42), $g(t, x_{actual}, u)$ accounts for disturbances, uncertainties and unmodelled dynamics. Algorithm 1 is valid provided that the following condition is satisfied

$$\|g(t, x_{actual}, u)\| \le b \tag{95}$$

where b is a positive constant. In addition, $(bh)^2$ is negligible (h is the execution horizon) and Assumptions 1-3 hold.

Proof. As previously mentioned, for the algorithm proposed in Section 3.4 to be valid, the perturbation in the initial conditions of the open loop optimal control problem should be sufficiently small. This allows the use of neighbouring extremal paths theory to modify the pre-computed input signal. As discussed in Section 3.3, the introduction of a small perturbation in the initial conditions results in a linear optimal control problem, where the first-order necessary conditions for optimality is described by

$$\begin{aligned}
&\dot{x} = f(x, u, t) \\
&\dot{\lambda}^T = -\frac{\partial H}{\partial x} \\
&\frac{\partial H}{\partial u} = 0 \\
&\lambda^T(t_f) = \left(\frac{\partial \phi}{\partial x}\right)_{t=T}
\end{aligned} \tag{96}$$

with $x_0 = x(t_{k+1})$ given, is linearized around the available optimum solution leading to

$$
\begin{aligned}
&\delta\dot{x} = f_x \delta x + f_u \delta u \\
&\delta\dot{\lambda} = -H_{xx}\delta x - f_u^T \delta\lambda - H_{xu}\delta u \\
&H_{ux}\delta x + f_u^T \delta\lambda + H_{uu}\delta u \\
&\delta\lambda(t_f) = (\phi_{xx})_{t=t_f}
\end{aligned}
\tag{97}
$$

with $\delta x(t_0)$ given. In the above equations ϕ is defined as

$$
\phi = \left\| x\left(t+T; x(t)\right) \right\|_{\mathbf{P}}^2 \tag{98}
$$

As we have used Taylor series expansion to derive these equations, the higher order terms are neglected, implying that $\delta x_i^n \approx 0$, $\delta u_i^n \approx 0$ and $\delta\lambda_i^n \approx 0$ for $n>1$, where δx_i, δu_i and $\delta\lambda_i$ stand for the i-th component of the perturbation vectors. Using norm notation, the following assumption is used implicitly in the derivation of equation (97)

$$
\begin{aligned}
&\|\delta x\|^2 \approx 0 \\
&\|\delta u\|^2 \approx 0 \\
&\|\delta\lambda\|^2 \approx 0
\end{aligned}
\tag{99}
$$

Equation (99) is therefore a condition for using neighbouring extremal paths theory. This condition can be stated in terms of the perturbation in the initial condition δx_0, assuming Assumption 1 holds; that is,

$$
\|\delta x_0\|^2 \approx 0 \tag{100}
$$

is a sufficient condition for the use of neighbouring extremal paths theory. It should be noted that δx_0, the mismatch between the predicted and the actual states at time t_{k+1}, is described by

$$\delta x_0 = x_{actual}(t_{k+1}) - x(t_{k+1}) = \int_{t_k}^{t_{k+1}} g(t, x, u) d\tau \tag{101}$$

Using equation (95) and (101), equation (100) holds if $(bh)^2$ is negligible and this concludes the proof. □

4.5 Example

In this section, we apply the proposed Algorithm 1 to the point stabilization of a differentially driven wheeled mobile robot, which is especially useful in formation stabilization, where it is desirable that each agent take a predefined position. The reader is referred to [11] for a discussion of cooperative control of mobile robots. For a complete discussion of the transformation of such system to the double integrator described by

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= u_1 \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= u_2 \end{aligned} \tag{102}$$

refer to Section 3.4.

We take the actual uncertain system under control as

$$
\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= u_1 + 0.5\sin(tx_1) \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= u_2 + 0.5\sin(tx_3)
\end{aligned}
\tag{103}
$$

with initial conditions chosen to be x_0=[6 2 5 -4]^T. We use the system described by equation (75) as the nominal system for the RHC synthesis and select the execution and prediction horizons to be 0.5 and 5 seconds, respectively. Weighting matrices **Q** and **R** are taken as identity matrices of appropriate dimension. Matrix **P** in (37) is found by solving the Lyapunov equation, equation (9) in [38].

As can be seen in Figures 8 to 11, the proposed modification in the generated control signal has improved the performance of the system considerably compared to the method pointed out in [44] (referred in the figures as *unmodified*). The introduced disturbances have resulted in some oscillations in the states in the unmodified case using the algorithm described in [44], whereas no oscillation is present when the proposed method of Algorithm 1 was used. In Figure 11 the magnitude of oscillation is growing, which shows that the existing method presented in [44] is not successful in stabilizing the states of the system. The value of $(bh)^2$=0.125 was determined to be negligible compared to the values selected above as required by Proposition 1. As the nominal system (75) is a linear system, Assumption 1 holds. Since the problem is linear with no constraints on the input, the RHC scheme described in [38] can be used, consequently, Assumptions 2 and 3 are satisfied.

The finite horizon open loop optimal control problem was solved numerically using MATLAB, where collocation method (see reference [55]) was utilized. The TPBVP was solved using backward sweep method [47].

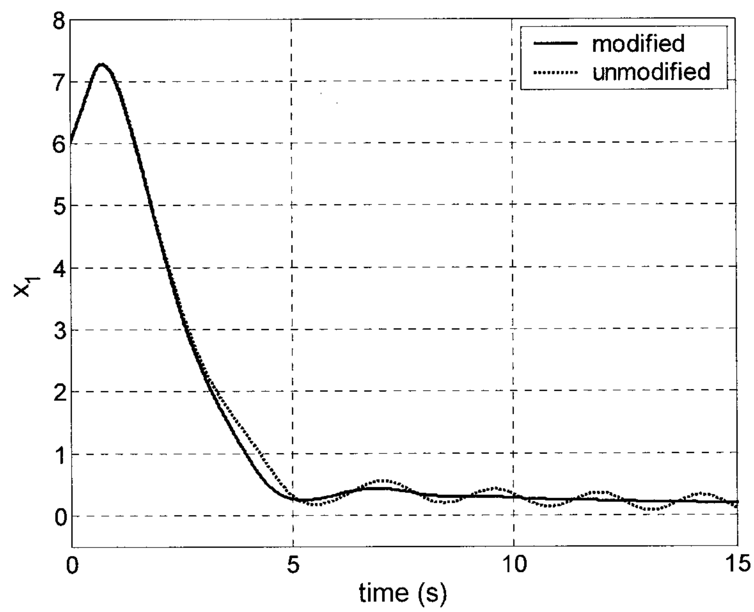

Figure 11. Time history of x_1

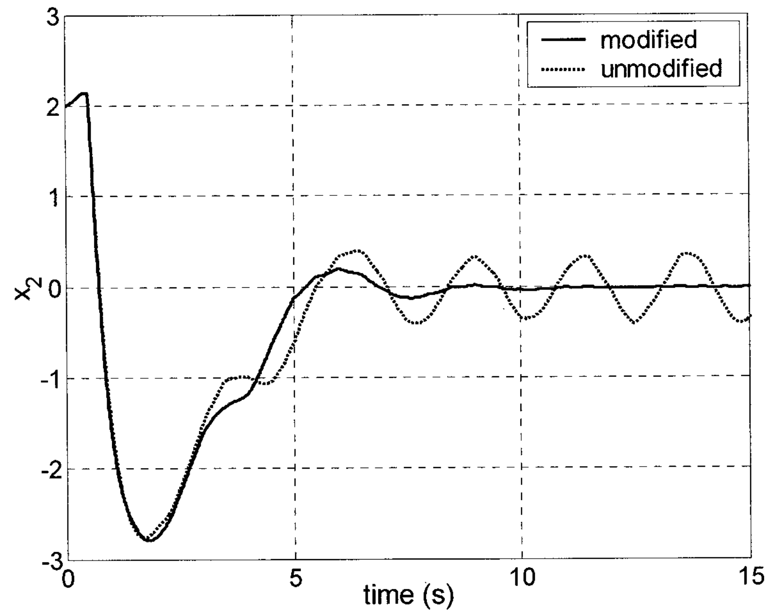

Figure 12. Time history of x_2

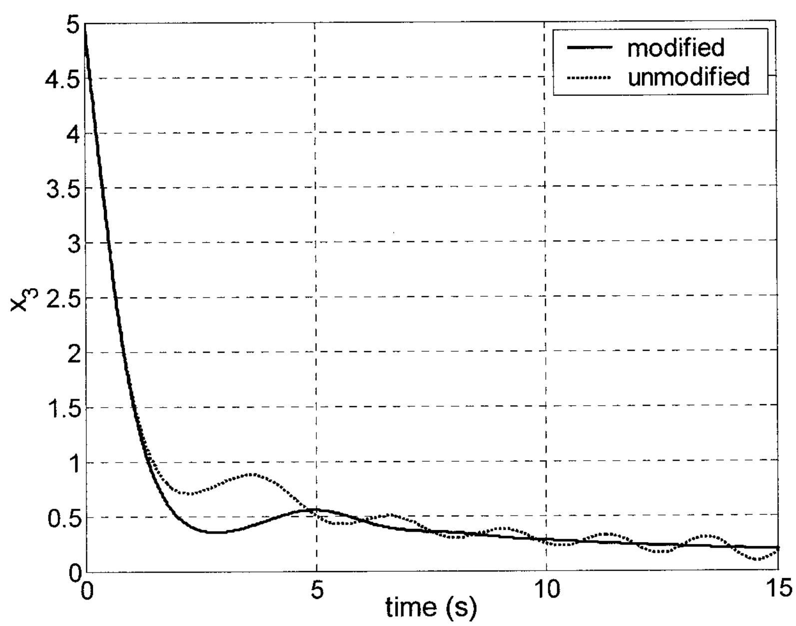

Figure 13. Time history of x_3

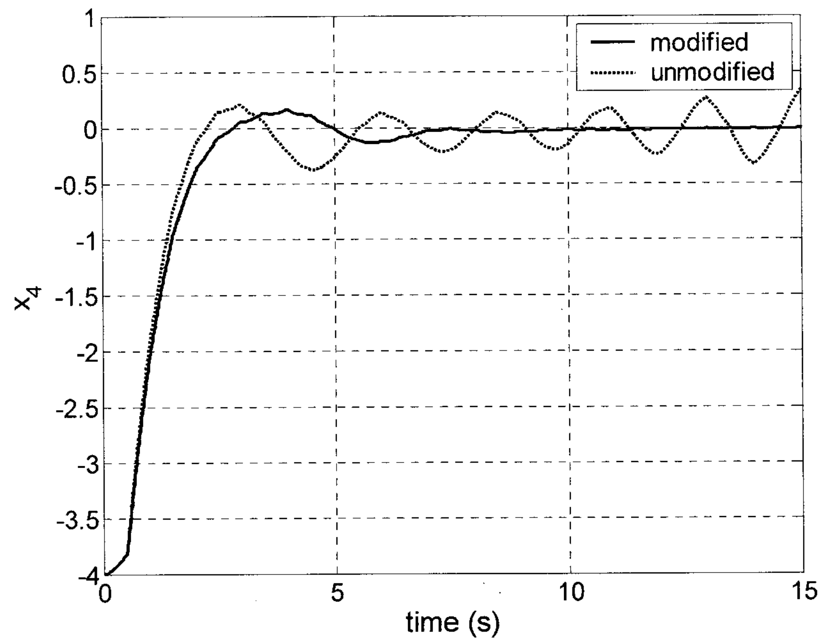

Figure 14. Time history of x_4

5 . Conclusions and Future Works

5.1 Conclusions

In this thesis, in Chapter 3, stability robustness for quasi-infinite Receding Horizon Control (RHC) of an uncertain nonlinear system was investigated. It was shown that satisfaction of a nonlinear inequality constraint is a sufficient condition for stability of the closed loop system. This condition was further specialized for linear systems. For this case sufficient conditions were derived to guarantee the stability of the closed loop system, as long as the states are outside a bounded set containing the desired equilibrium point for an uncertain linear system under the assumption of bounded perturbations. Simulations of a mobile robot of unicycle type with constraints on the inputs demonstrated the validity of the proposed analysis.

In Chapter 4, a novel Receding Horizon Control strategy for uncertain nonlinear systems was proposed considering the effect of computational delay. The approach is composed of state prediction, trajectory generation, and trajectory correction. To allow for the computation of the optimal trajectories, states are predicted at the next sampling time. The predicted values of the states are used as initial conditions for the finite horizon open loop optimal control problem, allowing the optimal input profiles to be computed one sampling time in advance. At the time of implementation, when the new data from the states becomes available, the pre-computed input is modified using the perturbation analysis done off-line. The on-line modification analysis is composed of the solution to two initial value problems, assumed to be computed in negligible time. The proposed method is valid as long as the perturbation in the states and the sampling time are sufficiently small. The method was applied to simulations of a mobile robot of unicycle type, where the proposed method shows significant improvement in performance compared to existing methods.

It is anticipated that these new results will find significant utility for control design of mobile robot and unmanned aerial vehicle (UAV) systems, especially cooperative control of a group of them.

5.2 Future Works

In this section, we state the possible future directions for the approaches proposed in this thesis. Future directions for the approach presented in Chapter 3 include experimental applications to mobile robot and UAV systems, and generalization to account for the effect of computational delays. For the approach in Chapter 4, experimental applications to mobile robot and UAV systems, and the addition of sufficient conditions for robust stability constitute the future works. The generalization of the method for RHC schemes requiring terminal inequality constraints and input constraints should also be addressed for the method discussed in Chapter 3.

6 . References

[1] L. E. Parker, "Current State of the Art in Distributed Robot Systems", *Distributed Autonomous Robotic Systems 4*, Lynne E. Parker, George Bekey, and Jacob Barhen (eds.), Springer, 2000: 3-12.

[2] Y. Cao, A. S. Fukunaga, A. B. Kahng, "Cooperative Mobile Robotics: Antecedents and Directions", *Autonomous Robots vol 4*, 1997. Reprinted as a chapter in the book: Robot Colonies, Arkin RC and Bekey GA, eds., Kluwer Academic Press.

[3] V. Gazi and K. M. Passino, "Stability Analysis of Swarms", *IEEE Transactions on Automatic Control*, Vol. 48, No. 4, pp. 692-697, April 2003.

[4] N. Shimoyama, K. Sugawara, T. Mizuguchi, Y. Hayakawa, M. Sano, "Collective Motion in a System of Motile Elements", *Physical Review Letters*, Vol. 76, No. 20, 1996.

[5] T. Vicek, A. Czirok, E. Ben-Jacob, I. Cohen, O. Shochet, "Novel Type of Phase Transition in a System of Self-Driven Particles", *Physical Review Letters*, Vol. 75, No. 6, 1995.

[6] Czirok, E. Ben-Jacob, I. Cohen, T. Vicek, "Formation of Complex Bacterial Colonies Via Self-Generated Vortices", *Physical Review E*, Vol. 54, No.2, 1996.

[7] Huepe, M. Aldana, "Intermittency and Clustering in a System of Self-Driven Particles", *Physical Review Letters*, Vol. 92, No. 16, 2004.

[8] W. Ren, R. W. Beard, "Virtual Structure Based Spacecraft Formation Control with Formation Feedback", *AIAA Guidance, Navigation, and Control Conference and Exhibition*, August 2002, Monterey, California.

[9] M. Mesbahi, F. Y. Hadaegh, "Formation Flying Control of Multiple Spacecraft via Graph, Matrix Inequility and Switching", *AIAA Journal of Guidance, Control, and Dynamics*, March 2001.

[10] J. Bellingham, M. Tillerson, A. Richards, and J. How, "Multi-Task Allocation and Path Planning for Cooperating UAVs", *Proc. of the Second Annual Conference on Cooperative Control and Optimization*, Nov 2001.

[11] J. T. Feddema, C. Lewis, D. A. Schoenwald, "Decentralized Control of Cooperative Robotic Vehicles: Theory and Application", *IEEE Transactions on Robotics and Automation*, Vol. 18, No. 5, October 2002.

[12] T. Balch, R. C. Arkin, "Behavior-based Formation Control for Multi-Robot Teams", *IEEE Transactions on Robotics and Automation*, Vol. 14, No. 6, December 1998, pp. 926-939.

[13] R. Skjetne, S. Moi, T. I. Fossen, "Nonlinear Formation Control of Marine Craft", *Proceedings of the 41st IEEE Conference on Decision and Control*, Las Vegas, December 2002.

[14] R. Olfati-Saber, R. M. Murray, "Graph Rigidity and Distributed Formation Stabilization of Multi-Vehicle Systems", *IEEE Conference on Decision and Control*, Las Vegas, 2002.

[15] J. P. Desai, V. Kumar, J. P. Ostrowski, "Control of changes in formation for a team of mobile robots", *IEEE International Conference on Robotics and Automation*, Detroit, May 1999, vol.2, pp. 1556 -1561.

[16] M. Tillerson, G. Inalhan, J. P. How, "Co-ordination and control of distributed spacecraft systems using convex optimization techniques", *Int. J.*

Robust Nonlinear Control, **12**, pp. 207–242, 2002.

[17] N. E. Leonard, E. Fiorelli, Virtual Leaders, "Artificial Potentials and Coordinated Control of Groups", *Proceedings of the 40th IEEE Conference on Decision and Control*, 2001.

[18] J. Lawton, R. Beard, B. Young, "A Decentralized Approach To Formation Maneuvers", *IEEE Transactions on Robotics and Automation*, **19** (6), pp. 933-941, December 2003.

[19] P. Ogren, M. Egerstedt, X. Hu, "A Control Lyapunov Function Approach to Multiagent Coordination", *IEEE Transactions on Robotics and Automation*, **18** (5), October 2002.

[20] W. B. Dunbar, "Distributed Receding Horizon Control of Multiagent Systems", *Ph.D Thesis*, California Institute of Technology, 2004.

[21] A. Richards, J. How, "Model Predictive Control of Vehicle Maneuvers with Guaranteed Completion Time and Robust Feasibility", *American Control Conference*, June 2003.

[22] R. Vidal, O. Shakernia, H. J. Kim, D. Shim, S. Sastry, "Probabilistic Pursuit-Evasion Games: Theory, Implementation, and Experimental Evaluation", *IEEE Transactions on Robotics and Automation*, October 2002.

[23] R. Vaughan, N. Sumpter, A. Frost, S. Cameron, "Robot Sheepdog Project Achieves Automatic Flock Control", *Proceedings of the International Conference on Simulation of Adaptive Behaviour*, Zurich, Switzerland, 1998.

[24] R. Vaughan, N. Sumpter, J. Henderson, A. Frost, S. Cameron, "Robot Control of Animal Flocks", *Proceedings of the International Symposium on*

Intelligent Control, Gaithersburg, Maryland, USA, 1998.

[25] J. P. Hespanha, H. J. Kim, S. Sastry, "Multiple-agent Probabilistic Pursuit-Evasion Games", *Proceedings of the IEEE Conference on Decision and Control*, pp. 2432-2437, 1999.

[26] H. Van Dyke Parunak, S. Brueckner, "Entropy and Self-Organization in Multi-Agent Systems", *Proceedings of the International Conference on Autonomous Agents (Agents 2001)*, pp. 124-130.

[27] C. W. Reynolds, "Boids: Background and Update", http://www.red3d.com/cwr/boids/

[28] Command and Conquer, Red Alert, Westwood Studios, http://www.eagames.com/official/cc/redalertpc/us/.

[29] Age of Empires, Microsoft, http://www.microsoft.com/games/empires/.

[30] C. R. Cutler, B. L. Ramaker, "Dynamic matrix control – A computer control algorithm", *Proceedings of Joint Automatic Control Conference*, San Francisco, CA, Paper WP5-B, 1980.

[31] D. Q. Mayne, J. B. Rawlings, C. V. Rao, P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality", *Automatica*, **36**, pp. 789-814, 2000.

[32] R. R. Bitmead, M. Gevers, V. Wertz, "Adaptive optimal control – the thinking man's GPC", International series in systems and control engineering, *Prentice-Hall*, 1990.

[33] A. Jadbabaie, "Receding horizon control of nonlinear systems: A control Lyapunov function approach", *Ph.D Thesis, California Institute of Technology,* Pasadena, CA, 2000.

[34] M. Cannon, B. Kouvaritakis, J. A. Rossiter, "Efficient active set optimization in triple mode MPC", *IEEE Transactions on Automatic Control*, **46** (8), pp. 1307-1312, 2001.

[35] M. B. Milam, K. Mushambi, R. M. Murray, "A new computational approach to real-time trajectory generation for constrained mechanical systems", *Proceedings of the 39th IEEE Conference on Decision and Control*, Sydney, Australia, pp. 845 – 851, 2002.

[36] M. B. Milam, "Real-time optimal trajectory generation for constrained dynamical systems", *PhD Thesis, California Institute of Technology,* Pasadena, CA, 2003.

[37] H. Michalska, D. Q. Mayne, "Robust Receding Horizon Control of Constrained Nonlinear Systems", *IEEE Transactions on Automatic Control*, **38** (11), pp. 1623-1633, 1993.

[38] H. Chen, F. Allgower, "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability", *Automatica*, **34**, pp.1205-1217, 1998.

[39] S. Boyd, L. Vandenberghe, "Convex Optimization", Cambridge University Press, Cambridge, UK, pp. 29-30, Chapter 2, 2004.

[40] B. J. Young, R. W. Beard, J. M. Kelsey, "A control scheme for improving multi-vehicle formation maneuvers", *American Control Conference*, Arlington, VA, pp. 704-709, 2001.

[41] J. P. Desai, J. Ostrowski, V. Kumar, "Controlling formations of multiple mobile robots", *Proceedings of the IEEE International Conference on Robotics and Automation*, Leuven, Belgium, pp. 2864-2869, 1998.

[42] J.-B. Pomet, B. Thuliot, G. Bastin, G. Campion, "A hybrid strategy for the feedback stabilization of nonholonomic mobile robots", *Proceedings of the IEEE International Conference on Robotics and Automation*, Nice, France, pp. 129 – 134, 1992.

[43] Ph. Martin, R. M. Murray, P. Rouchon, "Flat systems, equivalence and trajectory generation", CDS Technical Report, California Institute of Technology, Pasadena, CA, 2003.

[44] M. B. Milam, R. Franz, J. E. Hauser, R. M. Murray, "Receding horizon control of a vectored thrust flight experiment", *submitted to IEE Proceedings on Control Theory and Applications*.

[45] J. Strizzi, I. M. Ross, F. Fahroo, "Towards real-time computation of optimal controls for nonlinear systems", *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Monterey, California, August 2002, pp. 1967-1976.

[46] H. Yan, F. Fahroo, I. M. Ross, "Real-time computation of neighboring optimal control laws", *AIAA Guidance, Navigation, and Control Conference and Exhibit,* Monterey, California, August 2002, pp. 902-908.

[47] A. E. Bryson, Y.-C. Ho, "Applied optimal control, optimization, estimation and control", *John Wiley & Sons*, 1975.

[48] M. Ascher, R. M. M. Mattheij, R. D. Russell, Numerical Solution of Boundary Value Problems for Ordinary Differential Equations, U. Prentice Hall, 1988.

[49] W. Enright, Numerical Solution of Boundary Value Problems for Ordinary Differential Equations, Lecture Notes, available: http://www.cs.toronto.edu/~enright/teaching/CSC2322/BV.ps

[50] H. Khalil, Nonlinear Systems, Prentice Hall, 2002.

[51] M. Rathinam, Differentially Flat Nonlinear Control Systems, Ph.D Thesis, California Institute of Technology, 1997.

[52] B. Gholami, B. W. Gordon, C. A. Rabbath, "Quasi-infinite receding horizon control of systems with bounded perturbation", Submitted to ASME Journal of Dynamic Systems, Measurement, and Control.

[53] B. Gholami, B. W. Gordon, C. A. Rabbath, "Receding Horizon Control of uncertain nonlinear systems subject to computational delay", Submitted to Proceedings of IMECHE, Part I, Journal of Systems and Control Engineering.

[54] P. Ogren, "Formations and obstacle avoidance in mobile robot control", *Ph.D Thesis*, Royal Institute of Technology, Stockholm, Sweden, 2003.

[55] C. Hargraves, S. Paris, "Direct trajectory optimization using nonlinear programming and collocation", *AIAA Journal of Guidance and Control*, 10, 1987, pp. 338–342.

[56] W. B. Dunbar, Distributed Receding Horizon Control of Multiagent Systems, Ph.D Thesis, California Institute of Technology, 2004.

Appendix

Transforming the Equations of a Mobile Robot of Unicycle Type to a Two-Dimensional Double Integrator

We would like to show that the equations of a mobile robot of unicycle type can be transformed to a 2D double integrator by a choosing a different set of coordinates [54]. Take

$$\begin{aligned} x_1 &= x + L_a \cos\theta \\ x_3 &= y + L_a \sin\theta \end{aligned} \tag{104}$$

Therefore

$$\begin{aligned} \dot{x}_1 &= \dot{x} - L_a \omega \sin\theta = v\cos\theta - L_a \omega \sin\theta \\ \dot{x}_3 &= \dot{y} + L_a \omega \cos\theta = v\cos\theta + L_a \omega \cos\theta \end{aligned} \tag{105}$$

Now defining the inputs as

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}}_{R(\theta)} \begin{bmatrix} v \\ L_a \omega \end{bmatrix} \tag{106}$$

Differentiating (107) one more time results in

$$\begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_3 \end{bmatrix} = \dot{R}(\theta)\begin{bmatrix} v \\ L_a\omega \end{bmatrix} + R(\theta)\begin{bmatrix} \dot{v} \\ L_a\dot{\omega} \end{bmatrix} = R(\theta)\begin{bmatrix} \frac{F}{m} - L_a\omega^2 \\ \frac{L_a\tau}{J} + v\omega \end{bmatrix} \tag{108}$$

As a result, the relationship between the newly assigned inputs and the dynamic parameters of the system is described by

$$\begin{bmatrix} \frac{1}{m}F \\ \frac{L_a}{J}\tau \end{bmatrix} = \begin{bmatrix} L_a\omega^2 \\ -v\omega \end{bmatrix} + \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}^{-1}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{109}$$

Finally the transformed equations of the mobile robot of unicycle type is given by the following equation

$$\begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_3 \end{bmatrix} = R(\theta)\begin{bmatrix} \frac{F}{m} - L_a\omega^2 \\ \frac{L_a\tau}{J} + v\omega \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{110}$$