Periodic terminal ingredients for stabilizing nonlinear model predictive control

Tetteroo, Martin

Award date:
2018

EINDHOVEN UNIVERSITY OF TECHNOLOGY

MASTER: SYSTEMS AND CONTROL

DEPARTMENT OF MECHANICAL ENGINEERING

RESEARCH GROUP: CONTROL SYSTEMS EE

Periodic Terminal Ingredients for Stabilizing Nonlinear Model Predictive Control

MASTER'S THESIS

Date of final presentation:
August 14, 2018

Author:
Martin Tetteroo
ID:0652892

Supervisor:
Dr. Mircea Lazăr

Committee:
Prof.Dr. Siep Weiland
Dr.Ir Erjen Lefeber

August 8, 2018

Technische Universiteit
Eindhoven
University of Technology

Declaration concerning the TU/e Code of Scientific Conduct for the Master's thesis

I have read the TU/e Code of Scientific Conduct[i].

I hereby declare that my Master's thesis has been carried out in accordance with the rules of the TU/e Code of Scientific Conduct

Date

30-07-18

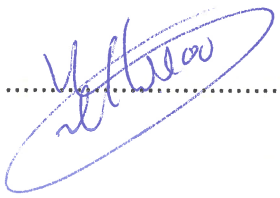
Name

Martin Tetterov

ID-number

0652892

Signature



Submit the signed declaration to the student administration of your department.

[i] See: http://www.tue.nl/en/university/about-the-university/integrity/scientific-integrity/
The Netherlands Code of Conduct for Academic Practice of the VSNU can be found here also.
More information about scientific integrity is published on the websites of TU/e and VSNU

Contents

Summary

The terminal cost and terminal set method for guaranteeing stability of nonlinear model predictive control (MPC) closed–loop systems is theoretically appealing but often impractical. This is due to the difficulty of computing invariant sets and control Lyapunov functions for general nonlinear systems. In this thesis a novel method is proposed for computing time–varying terminal costs and sets by means of first order or second order Taylor approximations of the nonlinear system dynamics. The method first solves a set of linear matrix inequalities to compute the terminal ingredients for the approximated dynamics. Then, a small scale global nonlinear optimization problem is solved to check the validity of the terminal ingredients for the nonlinear dynamics. The proposed method also allows for time–varying linear or nonlinear terminal control laws. Nonlinear reachable ellipsoidal sets have been researched to better approximate the nonlinear system's behaviour, allowing for less constrictive sets. The developed method can result in significant enlargements of the domain of attraction of the nonlinear MPC closed–loop system, as demonstrated by academic examples.

1 Introduction

In this thesis current methods are considered for construction of terminal ingredients (i.e., terminal cost and set) for guaranteeing stability of nonlinear model predictive control (NMPC), while proposing two novel methods. The addition of terminal ingredients in model predictive control remains the main approach to guaranteeing recursive feasibility and stability, as originally proven in [1] for the continuous–time NMPC setting. The overview [2] describes the various implementations of the terminal cost and set approach as well as alternative approaches that do not require terminal ingredients, [3] and [4], for the discrete–time NMPC setting. A recent useful review and comparison of discrete–time nonlinear MPC schemes with guaranteed stability was provided in [5], along with implementation details for specific types of ingredients (e.g., polytopic or ellipsoidal sets, and quadratic or infinity norm based costs).

The generally adopted methodology for computing terminal costs and sets in NMPC linearizes the nonlinear system around the zero equilibrium and computes the terminal cost as a local Lyapunov function for a locally stabilizing linear state feedback controller, typically taken as the linear quadratic regulator (LQR). Then a suitable terminal set is computed as the maximal admissible invariant set for the linearized dynamics in closed–loop with the LQR controller or, as an admissible sublevel set of the terminal cost. This method was successfully applied to nonlinear dynamics by correcting for the remaining linearization error in [1], [6] and [7] for continuous–time NMPC. Later on, a corresponding method with correction for the linearization error was worked out for discrete–time NMPC in [8] and [9]. These works consider linear (e.g., first order Taylor) approximations of the nonlinear dynamics, linear local control laws, quadratic terminal costs and bound the nonlinear remainder of the approximation using worst case norm bounds or global nonlinear optimization over the candidate terminal set. The nice feature of this approach is its simplicity, but the resulting terminal sets can be quite conservative. In comparison, other approaches as the ones in [10] and [5] use additionally polytopic linear difference inclusions (pLDIs) over–approximations instead of compensating for the linearization error. pLDIs over–approximations can be less conservative but they are somewhat difficult to construct for general nonlinear systems, especially when the state–space dimension increases.

To alleviate conservativeness, efforts have been made to improve the design and/or computation of terminal ingredients for nonlinear systems in a number of works, as for example: by the use of interval arithmetics in [11], by the use of difference of convex functions in [12], by the use of extended (actually, periodic) invariance in [13], or by using polytopic sets in [14]. Whenever relying on polytopic objects as terminal sets, however, these methods would lack tractability for higher order systems. The LQR local control was extended to a nonlinear state feedback control law in [15]. A notable recent effort to obtain a systematically applicable method for general continuous–time nonlinear MPC was made in [16] by using higher order Taylor series expansions and sum–of–squares techniques. This framework is able to yield tight approximations of the nonlinear dynamics, but it can become computationally troublesome for higher order approximations and/or state–space dimension.

At the core, the difficulty of computing stabilizing terminal ingredients for nonlinear MPC lies within the difficulty of computing invariant sets and control Lyapunov functions for nonlinear systems. A new design that circumvents this difficulty was proposed in [17] which uses a sequence of terminal sets and local control laws in the off–line design and a single terminal set and terminal cost on–line, in combination with a cyclic time–varying prediction horizon. By resorting to finite–step (non–monotone) control Lyapunov functions and invariant sets, a new design of terminal ingredients for nonlinear MPC was proposed in [18], which does not necessarily require a cyclic horizon. Therein, it was shown that stability of nonlinear MPC can be guaranteed by using a periodically time–varying terminal set and a time–invariant terminal cost and local control

law, while using a standard horizon. The cyclic horizon implementation allows using a single time–invariant terminal set for the MPC scheme in [18] as well (see Section 3.1 therein), which is consistent with the idea originally proposed in [17].

In this thesis a systematic framework is aimed to be attained for setting up and computing terminal costs and terminal sets for discrete–time nonlinear MPC schemes that achieves a good trade–off between conservatism of the resulting terminal sets (which is important for enlarging the domain of attraction of the MPC closed–loop system), general applicability to any nonlinear dynamics, and ease of implementation. To this end periodically time–varying terminal costs, terminal sets and terminal control laws, are proposed which reduce conservatism. Approximated nonlinear reachable sets with coupled time–varying terminal cost to construct time–varying terminal sets are researched. A new method is proposed, where first order or quasi–second order Taylor approximations of the nonlinear dynamics are employed in combination with quadratic terminal costs and linear local control laws to attain scalable linear matrix inequalities (LMIs) formulations. An extension to nonlinear local control laws as proposed in [15] is allowed. Finally, solving a global nonlinear optimization problem to correct for the approximation error is still required; since this optimization problem is of small scale and it is performed over an ellipsoidal set, effective nonlinear solvers are available (e.g., Ipopt [19], Knitro [20], Baron [21], or a sequential quadratic programming (SQP) solver with line search). The result is a finite collection of quadratic terminal costs and ellipsoidal terminal sets. The developed method can be easily automated and implemented in Matlab. Results on academic examples from the nonlinear MPC literature are encouraging.

The thesis is structured as follows, in Section 2 the nonlinear system and MPC problem are defined, terminal ingredients introduced and a proposal for theoretic design is given. In Section 3 a method of computation of terminal ingredients for nonlinear systems and the Matlab implementation is presented. The construction of terminal sets via reachable sets is researched in Section 4. Section 5 presents examples and Section 6 concludes the thesis including recommendations.

2 MPC problem set–up and terminal ingredients design

The discrete–time dynamical nonlinear systems are considered

$$x(k+1) = \phi(x(k), u(k)), \quad k \in \mathbb{N}, \tag{1}$$

where $\phi : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is the system dynamics, $x(k)$ is the state and $u(k)$ the input at discrete–time instant k. Assuming that $\phi(0,0) = 0$. A model predictive control (MPC) optimization problem is defined in (2) for a prediction horizon N. The prediction model in the MPC problem (2b), is a copy of the system dynamics, i.e.,

$$x_{i+1|k} = \phi(x_{i|k}, u_{i|k}), \quad i \in \{0, \dots, N-1\}, \quad k \in \mathbb{N}.$$

By defining the sequence of predicted inputs as $U_k = \{u_{0|k}, \dots, u_{N-1|k}\}$ yields:

$$\min_{U_k} J(x_{0|k}, U_k) = \min_{U_k} \big\{ F(x_{N|k}, k) + \sum_{i=0}^{N-1} l(x_{i|k}, u_{i|k}) \big\} \tag{2a}$$

subject to:

$$x_{i+1|k} = \phi(x_{i|k}, u_{i|k}), \quad i \in \{0, \dots, N-1\} \tag{2b}$$
$$x_{i|k} \in \mathbb{X} \subseteq \mathbb{R}^n, \quad i \in \{0, \dots, N-1\} \tag{2c}$$
$$u_{i|k} \in \mathbb{U} \subseteq \mathbb{R}^m, \quad i \in \{0, \dots, N-1\} \tag{2d}$$
$$x_{N|k} \in \mathbb{X}_T(k) \subseteq \mathbb{R}^n \tag{2e}$$

where $x_{0|k} = x(k)$, $u_{0|k} = u(k)$, $F : \mathbb{R}^n \times \mathbb{N} \to \mathbb{R}_+$ is the terminal cost, $\mathbb{X}_T$ is the terminal set and $l : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_+$ the stage cost. Note that the terminal cost and set are allowed to be time–varying. Also, the stage cost $l(\cdot,\cdot)$ is assumed $l(0,0) = 0$ and $l(x,\cdot) \geq \alpha_3(||x||)$ $\forall x \in \mathbb{X}$ where α_3 is a $\mathcal{K}_\infty$ class function.

2.1 Terminal ingredients design

For any $M \in \mathbb{N}_{\geq 1}$, let $\{(F_j(\cdot), \mathcal{S}_j, h_j(\cdot))\}_{j=0,\dots,M-1}$ denote a set of terminal ingredients, where $F_j : \mathbb{R}^n \to \mathbb{R}_+$ with $F_j(0) = 0$ are terminal costs, $\mathcal{S}_j$ are terminal sets and $h_j : \mathbb{R}^n \to \mathbb{R}^m$ with $h_j(0) = 0$ are terminal control laws for all $j = 0, \dots, M-1$. Define $\Phi_j(x) := \phi(x, h_j(x))$ and for a subset $\mathcal{S}$ of $\mathbb{R}^n$ define $\Phi_j(\mathcal{S}) := \{\Phi_j(x) \ : \ x \in \mathcal{S}\}$.

Assumption 2.1 *Terminal ingredients properties.*

- *(i) Terminal sets:*

$$\Phi_j(\mathcal{S}_j) \subseteq \mathcal{S}_{j+1}, \quad j = 0, \dots, M-2 \tag{3a}$$
$$\Phi_{M-1}(\mathcal{S}_{M-1}) \subseteq \mathcal{S}_0, \tag{3b}$$
$$h_j(\mathcal{S}_j) \subseteq \mathbb{U}, \quad j = 0, \dots, M-1, \tag{3c}$$
$$\mathcal{B}_\varepsilon(0) \subset \mathcal{S}_j \subseteq \mathbb{X}, \quad j = 0, \dots, M-1. \tag{3d}$$

- *(ii) Terminal costs:*

$$F_{j+1}(\Phi_j(x)) - F_j(x) + l(x, h_j(x)) \leq 0, \quad \forall x \in \mathcal{S}_j, j = 0, \dots, M-2, \tag{4a}$$
$$F_0(\Phi_{M-1}(x)) - F_{M-1}(x) + l(x, h_{M-1}(x)) \leq 0, \quad \forall x \in \mathcal{S}_{M-1}. \tag{4b}$$

Above $\mathcal{B}_\varepsilon(0)$ denotes an ε radius ball in $\mathbb{R}^n$ centered at zero. Observe that if $M = 1$ the standard assumptions of the terminal set and terminal cost for discrete–time nonlinear MPC [2] are obtained. Observe also that when $\mathcal{S}_j := \{x \ : \ F_j(x) \leq \alpha\}$ is chosen for some common $\alpha > 0$ and all $j = 0, \ldots, M-1$, Assumption 2.1-(ii) already implies (3a) and (3b), while (3c) and (3d) can be met by suitably scaling $\alpha > 0$. Indeed, for example, (4a) yields:

$$F_{j+1}(\Phi_j(x)) \leq F_j(x) \leq \alpha, \quad \forall x \in \mathcal{S}_j.$$

Hence, $\Phi_j(x) \in \mathcal{S}_{j+1} = \{x \ : \ F_{j+1}(x) \leq \alpha\}$ for all $x \in \mathcal{S}_j$ and (3a) holds.

To make use of the multiple terminal costs and sets in the MPC problem (2) the approach of [18] is followed and the time–varying terminal cost and set are defined, where mod is the standard modulo operator, as follows.

$$\begin{aligned} &\text{If } k = 0 \ \& \ \mathbb{X}_T(0) = \mathcal{S}_j, \quad \forall j = 0, \ldots, M-1 \\ &\quad \Rightarrow \mathbb{X}_T(k) := \mathcal{S}_{(j+k) \bmod M}, \qquad (5a) \\ &\quad \Rightarrow F(x,0) = F_j(x) \ \& \ F(x,k) = F_{(j+k) \bmod M}(x). \qquad (5b) \end{aligned}$$

As indicated above, it is possible to assign any set $\mathcal{S}_j$ as the terminal set at time $k = 0$, which triggers the corresponding terminal cost, and then as time varies, the terminal sets and costs are varying periodically accordingly.

Remark 2.2 *In comparison, [18] used the same time–varying terminal set conditions, but a different time–invariant terminal cost construction, while [17] used the set $\mathcal{S}_0$ as terminal set, a time–invariant common terminal cost and a time–varying cycling prediction horizon. The approach of [18] allows to use $\cup_{j=0,\ldots,M-1}\mathcal{S}_j$ as a standard, time–invariant terminal set at the cost of adding a mixed–integer terminal constraint, due to the specific time–invariant terminal cost constructed therein.*

Theorem 2.3 *Let $\{(F_j(\cdot), \mathcal{S}_j, h_j(\cdot))\}_{j=0,\ldots,M-1}$ for any $M \in \mathbb{N}_{\geq 1}$ satisfy Assumption 2.1. Suppose that the MPC problem (2) with the terminal set and terminal cost assigned as in (5) is feasible for all $x(0) \in \mathbb{X}_f(N) \subseteq \mathbb{X}$. Then the MPC problem (2) remains feasible for all $k \in \mathbb{N}$ and corresponding $x(k) \in \mathbb{X}$ and the closed–loop system (1) with $u(k) = u^*_{0|k}$ [1] for all $k \in \mathbb{N}$ is asymptotically stable for all $x(0) \in \mathbb{X}_f(N)$.*

Proof *The proof of recursive feasibility follows as in the proof of Theorem 5-(i) in [18] by replacing the terminal control law $h(x)$ with an appropriate time–varying correspondent, i.e., $h(x,k) := h_j(x)_{(j+k)modM}$, since the conditions on the terminal sets therein are the same as the ones in Assumption 2.1-(i). For any $k \in \mathbb{N}$ and $x(k) \in \mathbb{X}_f(N)$ there exists a $p \in \{0, 1, \ldots, M-1\}$ such that $p = (j+k)modM$. Following the usual steps [2] in proving stability of MPC and keeping*

[1] $u^*_{0|k}$ is the first element of the optimum U^*_k of problem (2) at time $k \in \mathbb{N}$.

in mind that $\mathbf{u}_{k+1} = \{u^*_{1|k}, \ldots, u^*_{N-1|k}, h(x^*_{N|k}, k)\}$ *is suboptimal but feasible, yields:*

$$\begin{aligned} V(x(k+1)) - V(x(k)) & \\ &:= J(x(k+1), \mathbf{u}^*_{k+1}) - J(x(k), \mathbf{u}^*_k) \\ &\leq J(x(k+1), \mathbf{u}_{k+1}) - J(x(k), \mathbf{u}^*_k) \\ &= F_{p+1}(x_{N|k+1}) + \sum_{i=0}^{N-1} l(x_{i|k+1}, u_{i|k+1}) \\ &\quad - F_p(x^*_{N|k}) - \sum_{i=0}^{N-1} l(x^*_{i|k}, u^*_{i|k}) \\ &\leq F_{p+1}(\Phi_p(x^*_{N|k})) - F_p(x^*_{N|k}) \\ &+ l(x^*_{N|k}, h_p(x^*_{N|k})) - l(x^*_{0|k}, u^*_{0|k}). \end{aligned}$$

Then directly obtained from (4) *and the usual class* $\mathcal{K}_\infty$ *lower bound of the stage cost:*

$$\begin{aligned} V(x(k+1)) - V(x(k)) & \\ &\leq F_{p+1}(\Phi_p(x^*_{N|k})) - F_p(x^*_{N|k}) \\ &+ l(x^*_{N|k}, h_p(x^*_{N|k})) - l(x^*_{0|k}, u^*_{0|k}) \\ &\leq -\alpha_3(\|x(k)\|). \end{aligned}$$

Above is used that $u_{i|k+1} = u^*_{i+1|k}$ *for* $i \in \{0, \ldots, N-2\}$, $x_{i|k+1} = x^*_{i+1|k}$ *for* $i \in \{0, \ldots, N-1\}$ *and* $F_{p+1} = F_0$ *for* $p = M - 1$. *The statement then follows from the standard discrete–time Lyapunov stability theorem for MPC [2].* □

This concludes the proposal of theoretical design for terminal sets and terminal costs, which gives possibility of using periodic terminal ingredients and terminal control laws while keeping the standard terminal ingredient design as a particular case where $M = 1$. The following section presents a method of computation of terminal sets and terminal costs that satisfy Assumption 2.1 for nonlinear systems.

3 Computation of terminal costs and terminal sets

In order to arrive at systematic computational algorithms for any nonlinear dynamics $\phi(\cdot,\cdot)$ and obtain scalable algorithms (this rules out polytopic terminal sets for higher order systems) the following choices are made.

Set $l(x,u) := x^TQx + u^TRu$ with $Q > 0, R > 0$. For all $j = 0,\dots,M-1$ set $F_j(x) := x^TP_jx$ with $P_j > 0$, set $\mathcal{S}_j := \{x \ : \ F_j(x) \le \alpha\}$ with $\alpha > 0$ and set $h_j(x) = K_jx$. Further more, for all $j = 0,\dots,M-1$ define

$$r_j(x) := \phi(x, K_jx) - (A + BK_j)x = \Phi_j(x) - (A + BK_j)x$$

as the approximation error between $\phi(x, h_j(x))$ and its first order Taylor linearization (A, B) around the origin. Defining $A_{K_j} := A + BK_j$, obtained from (4):

$$(A_{K_j} + r_j(x))^TP_{j+1}(A_{K_j} + r_j(x)) - x^TP_jx + x^T(Q + K_j^TRK_j)x \le 0,$$

or, equivalently,

$$x^T\left(A_{K_j}^TP_{j+1}A_{K_j} - P_j + Q + K_j^TRK_j\right)x + r_j(x)^TP_{j+1}r_j(x) + 2x^TA_{k_j}^TP_{j+1}r_j(x) \le 0.$$

For any $0 < \kappa_j < 1$ the above inequality can be split into a matrix inequality:

$$A_{K_j}^TP_{j+1}A_{K_j} - (1-\kappa_j)P_j + Q + K_j^TRK_j \le 0 \tag{6}$$

and a nonlinear inequality:

$$\mathcal{R}_j(x) := r_j(x)^TP_{j+1}r_j(x) + 2x^TA_{k_j}^TP_{j+1}r_j(x) - \kappa_jx^TP_jx \le 0.$$

Note that the above inequalities only have to hold for all $x \in \mathcal{S}_j$. Hence, to find a solution to the original terminal cost inequalities (4) it suffices to find a solution to (6) and to check that $\max_{x\in\mathcal{S}_j}\mathcal{R}_j(x) \le 0$. Using Schur's complement (6) can be written as (3).

$$\begin{bmatrix} (1-\kappa_j)P_j & (A+BK_j)^T & I_n & K_j^T \\ (A+BK_j) & P_{j+1}^{-1} & 0 & 0 \\ I_n & 0 & Q^{-1} & 0 \\ K_j & 0 & 0 & R^{-1} \end{bmatrix} \ge 0$$

By applying pre- and post multiplication with

$$\begin{bmatrix} P_j^{-1} & 0 \\ 0 & I_{2n+m} \end{bmatrix}$$

and the substitution of variables $O_j = P_j^{-1}$, $Y_j = K_jP_j^{-1}$, $j = 0,\dots,M-1$, $P_j > 0$ and $P_M := P_0$, the matrix inequalities (6) corresponding to all inequalities in (3) for $j = 0,\dots,M-1$ can be written as a linear matrix inequality (LMI):

$$\begin{bmatrix} (1-\kappa_j)O_j & (AO_j+BY_j)^T & O_j & Y_j^T \\ AO_j+BY_j & O_{j+1} & 0 & 0 \\ O_j & 0 & Q^{-1} & 0 \\ Y_j & 0 & 0 & R^{-1} \end{bmatrix} \ge 0. \tag{7}$$

To check that $\max_{x\in\mathcal{S}_j}\mathcal{R}_j(x) \le 0$, several global optimization solvers can be used, including Ipopt, Knitro, Baron and SQP with line search. Whenever the check fails, one needs to appropriately scale $\alpha > 0$ until the check holds. The volume of the terminal sets can be optimized by

maximizing over $\log\det(O_j)$ and by introducing α in the LMI (7). I.e., set $\mathcal{S}_j := \{x \,:\, F_j(x) \leq 1\}$, let $\bar{\alpha} := \frac{1}{\alpha}$ and set $F_j(x) = \frac{1}{\alpha} x^T P_j x$, $O_j = P_j^{-1}$ and replace Q^{-1}, R^{-1} by $\bar{\alpha}Q^{-1}$, $\bar{\alpha}R^{-1}$ in (7).

$$\begin{bmatrix} (1-\kappa_j)O_j & (AO_j + BY_j)^T & O_j & Y_j^T \\ AO_j + BY_j & O_{j+1} & 0 & 0 \\ O_j & 0 & \bar{\alpha}Q^{-1} & 0 \\ Y_j & 0 & 0 & \bar{\alpha}R^{-1} \end{bmatrix} \geq 0 \tag{8}$$

Indeed, solving (8) while minimizing over $\bar{\alpha}$ corresponds to maximizing α and enlarging $\mathcal{S}_j$.

Conservatism can be further reduced, while maintaining the same computational complexity and problem structure as above, by using a quasi–second order Taylor approximation of $\phi(\cdot,\cdot)$ around the origin. In this case define:

$$r_j(x) := \Phi_j(x) - (\bar{A}(x) + \bar{B}(x)K_j)x, \tag{9}$$

where

$$\bar{A}(x) := A + \begin{bmatrix} x^T G_{xx}^1 \\ \vdots \\ x^T G_{xx}^n \end{bmatrix}, \quad \bar{B}(x) := B + \begin{bmatrix} x^T G_{xu}^1 \\ \vdots \\ x^T G_{xu}^n \end{bmatrix},$$

and G_{xx}^i and G_{xu}^i are corresponding elements of appropriate dimension of the Hessian of $\phi_i(\cdot,\cdot)$ (i.e., $\phi(\cdot,\cdot) = [\phi_1(\cdot,\cdot), \ldots, \phi_n(\cdot,\cdot)]^T$). Observing that $\bar{A}(x)$ and $\bar{B}(x)$ are affine functions of x, by constraining the sets $\mathcal{S}_j$ to lie within a bounding polytope $\mathcal{P} \subseteq \mathbb{X}$ with set of vertices $\mathcal{P}_v$, the LMI corresponding to (7) becomes:

$$\begin{bmatrix} (1-\kappa_j)O_j & (\bar{A}(v)O_j + \bar{B}(v)Y_j)^T & O_j & Y_j^T \\ \bar{A}(v)O_j + \bar{B}(v)Y_j & O_{j+1} & 0 & 0 \\ O_j & 0 & Q^{-1} & 0 \\ Y_j & 0 & 0 & R^{-1} \end{bmatrix} \geq 0, \quad \forall v \in \mathcal{P}_v. \tag{10}$$

By evaluation of the LMIs on the vertices v of $\mathcal{P}_v$, the LMIs hold for all $x \in \mathcal{P}_v$ by convex combination. The check $\max_{x\in\mathcal{S}_j} \mathcal{R}_j(x) \leq 0$ remains the same for the $\mathcal{R}_j(x)$ corresponding to $r_j(x)$ as defined in (9).

If one increases the order of the Taylor approximation further, as proposed in [16] for continuous–time nonlinear MPC, computation of the terminal cost and set could still be brought down to linear matrix inequalities by using sum–of–squares (SOS) techniques, but it becomes much harder to automate the computational procedure and SOS methods are not as numerically stable or scalable as the LMIs (10). Using the proposed quasi–second order Taylor approximation already gives quite good results. Also it provides an exact approximation for bilinear systems.

Next, inspired by the approach of [15], the local control laws are extended to a nonlinear feedback law. The method therein, based on the more general H–infinity Riccati inequality for robust NMPC, is presented for disturbance free nonlinear systems as a stabilization method. This is possible if $\phi(x,u) = f(x) + g(x)u$ for appropriate functions $f(\cdot)$ and $g(\cdot)$. Then define:

$$h_j(x) := -(g^T(x)P_j g(x) + R)^{-1} g^T(x) P_j f(x). \tag{11}$$

Similarly as done above the approximation error can then be redefined:

$$r_j(x) = \Phi_j(x) - (\bar{A}(x) + \bar{B}(x)K_j)x, \tag{12}$$

where $r_j(x) = r_{1,j}(x) + g(x)r_{2,j}(x)$ is decomposed for

$$r_{1,j}(x) = f(x) + g(x)K_j x - (\bar{A}(x) + \bar{B}(x)K_j)x$$

and $r_{2,j}(x) = h_j(x) - K_j x$. Then the terminal cost inequalities (4) can be verified using the LMIs (10) and the check $\max_{x \in \mathcal{S}_j} \mathcal{R}_j(x) \leq 0$ for the $\mathcal{R}_j(x)$ corresponding to $r_j(x)$ defined in (12). Clearly, the nonlinear control law can also be used in combination with the first order Taylor approximation.

In summary, the key tuning knobs of the proposed computational method are the number M of terminal costs, terminal sets, and local control laws, the approximation of the nonlinear dynamics (first or quasi–second order Taylor approximation), and linear or nonlinear control laws. Additionally, the parameters κ_j and the level α can also be optimized, along with the volume of the sets $\mathcal{S}_j$. Polytopic constraints on states and inputs can be added as additional LMIs to (10), as shown next. Whenever the quasi–second order approximation and the pLDI provided by the vertices of $\mathcal{P}$ is an over–approximation of the nonlinear dynamics (in the sense of the solution set), then the nonlinear optimization check can be dropped. Therefore, compared to most of the methods cited in the Introduction, the proposed method shares the same computational features while adding additional design parameters for reducing conservatism.

3.1 Algorithmic Implementation

Implementation is done using Matlab, YALMIP [22], SDPT3 [23] and Knitro [20]. By convex optimisation the LMIs (7) or (10) can be solved while maximising the volume of the terminal sets. In order to also guarantee properties (3c) and (3d), the following LMIs can be used if the state and input constraints are polytopes according to [24]. I.e., the state constraints are described by $\mathbb{X} = \{x \ : \ H_x x \leq 1\}$ and the input constraints are described by $\mathbb{U} = \{u \ : \ H_u u \leq 1\}$ for appropriate matrices H_x and H_u. Letting $[H_x]_{p:}$ denote the p-th line of n_x size H_x, the LMIs

$$[H_x]_{p:} O_j [H_x]_{p:}^T \leq 1 \quad \forall p : 1, \ldots, n_x, \quad \forall j \in \{0, \ldots, M-1\}, \tag{13}$$

ensure that (3d) holds for $\mathcal{S}_j = \{x \ : \ F_j(x) \leq 1\}$. Similarly, the NLMIs

$$[H_u]_{p:} K_j O_j K_j^T [H_u]_{p:}^T \leq 1$$

can, by application of Schur's complement, be written as

$$\begin{bmatrix} 1 & [H_u]_{p:} K_j \\ K_j^T [H_u]_{p:}^T & O_j^{-1} \end{bmatrix} \geq 0$$

and by pre- and post multiplication of

$$\begin{bmatrix} 1 & 0 \\ 0 & O_j \end{bmatrix}$$

the equation is written as the LMIs (14) for $[H_u]_{p:}$ denoting the p-th line of n_u size H_u.

$$\begin{bmatrix} 1 & [H_u]_{p:} Y_j \\ Y_j^T [H_u]_{p:}^T & O_j \end{bmatrix} \geq 0 \quad \forall p : 1, \ldots, n_u, \quad \forall j \in \{0, \ldots, M-1\}, \tag{14}$$

The LMIs ensure that (3c) holds for $\mathcal{S}_j = \{x \ : \ F_j(x) \leq 1\}$.

For the case of using a quasi–second order approximation, define $\mathcal{P}_v := \eta \mathbb{X} = \{x \ : \ H_{\mathcal{P}} x \leq 1\}$ for $0 < \eta \leq 1$ and $[H_{\mathcal{P}}]_{p:}$ denote the p-th line of $n_{\mathcal{P}}$ size $H_{\mathcal{P}}$. The LMIs

$$[H_{\mathcal{P}}]_{p:} O_j [H_{\mathcal{P}}]_{p:}^T \leq 1 \quad \forall p : 1, \ldots, n_{\mathcal{P}}, \quad \forall j \in \{0, \ldots, M-1\} \tag{15}$$

ensure that $\mathcal{S}_j \subset \mathcal{P}$. Consider next the LMIs:

$$\begin{bmatrix} (1-\kappa_j)O_j & (\bar{A}(v)O_j + \bar{B}(v)Y_j)^T & O_j & Y_j^T \\ \bar{A}(v)O_j + \bar{B}(v)Y_j & O_{j+1} & 0 & 0 \\ O_j & 0 & \bar{\alpha}Q^{-1} & 0 \\ Y_j & 0 & 0 & \bar{\alpha}R^{-1} \end{bmatrix} \geq 0, \ \ \forall j \in \{0,\ldots,M-1\}, \forall v \in \mathcal{P}_v, \tag{16}$$

where $\bar{\alpha}$ is used to obtain the largest possible terminal sets and improve numerical aspects, while maximizing over the $\log(\det(O_j))$, where $j = \max(1, \lfloor \frac{M}{2} \rfloor)$ gives best results. Notice that when computing periodic control Lyapunov functions, a valid solution is for all functions to become equal. Forcing variance can be done by inclusion of the following constraint:

$$Y_{j+1} \geq \rho Y_j, \quad \forall j \in \{0,\ldots,M-2\}, \tag{17}$$

where $\rho > 1$ is a scaling factor. The second phase of the computation is the verification of $\max_{x \in \mathcal{S}_j} \mathcal{R}_j(x)$ using Knitro Multistart. When the validation fails the sequence is scaled $[0, \bar{\alpha})$, the scaling is optimized via bisection. The algorithms implementing the two phases described above are summarized next.

Alg. 1

Input $M \in \mathbb{N}_{\geq 1}$, $Q > 0$, $R > 0$, $\mathcal{P}_v$

Output $\{F_j(\cdot), \mathcal{S}_j, h_j(\cdot)\}_{j=0,\ldots,M-1}$, $\bar{\alpha}$

Begin

- $\min\{-\log(\det(O_j)) + \bar{\alpha}\}$ over (13)–(17)

IF feasible

- END

Alg. 2

Input $\{F_j(\cdot), \mathcal{S}_j, h_j(\cdot), r_j\}_{j=0,\ldots,M-1}, \bar{A}, \bar{B}, \bar{\alpha}, TOL$

Output α

Begin $\alpha_{upper} = \bar{\alpha}, \alpha_{lower} = 0$

- FOR$\{j \in \{0, \ldots, M-1\}\}$

{

– $\mathcal{R}_j^* = \max_{x \in \mathcal{S}_j} \mathcal{R}_j(x)$ where $\mathcal{S}_j = \{x | x^T P_j x \leq \alpha_{upper}\}$

IF $\mathcal{R}_j^* > 0$

{

– $\alpha = 0.5(\alpha_{upper} - \alpha_{lower})$

WHILE $(\alpha_{upper} - \alpha_{lower}) > TOL$

{

∗ $\mathcal{R}_j^* = \max_{x \in \mathcal{S}_j} \mathcal{R}_j(x)$ where $\mathcal{S}_j = \{x | x^T P_j x \leq \alpha\}$

IF $\mathcal{R}_j^* > 0$

· $\alpha_{upper} = \alpha$

· $\alpha = 0.5(\alpha_{upper} - \alpha_{lower})$

· return

IF $\mathcal{R}_j^* \leq 0$

· $\alpha_{lower} = \alpha$

· $\alpha = 0.5(\alpha_{upper} - \alpha_{lower})$

· return

}

}

}

- return α
- END

4 Reachability problem

While the LMI construction is a valid method of computing terminal sets based on the level set of terminal cost functions, it is based on the linearization of the nonlinear system. For an increase in accuracy of the dynamics of the system, nonlinear reachability is explored by multiple approximation techniques, keeping tractability in mind. A (possibly invariant) reachable ellipsoidal set $\mathcal{E}_j := \{x \ : \ x^T P_j x \leq \alpha\}$ is defined for nonlinear systems as in (18):

$$\{\phi(x,u) : x \in \mathcal{E}_j\} \subseteq \mathcal{E}_{j+1}. \tag{18}$$

In Section 3 terminal sets are constructed as level sets of the terminal cost functions. Using reachable sets it is possible to construct terminal sets directly from an initial input admissible set $\mathcal{S}_0$ without associated terminal cost.

A sequence $\mathcal{S}$ is calculated to have $\mathcal{S}_M \subseteq \mathcal{S}_0$ while keeping all sets in the sequence input admissible as in [18] using a single state feedback controller $u = Kx$, creating periodic invariance of the sets. Therefore the size of the sequence M is not known in advance. The set $\mathcal{S}_0$, constructed by LQR based LMIs and a level set of $\mathcal{S}_0 := \{x \ : \ F_0(x) \leq \alpha\}$, is scaled using bisection, when the sequence fails the input admissible bounds, followed by a recalculation of the sequence.

4.1 Lagrange remainder

This construction of the nonlinear reachable set is based on linearization and a method of linearization error bounding. The reachability problem for linear systems and ellipsoidal sets is described in [25]. In [26] the Intlab toolbox is used to overapproximate the Lagrange remainder. By application of a first order Taylor approximation on the nonlinear system, an error is introduced. The linearization error is described by the Lagrange remainder (19) $\forall s \in [1,n]$, the extended state $\xi = \begin{bmatrix} x^T & u^T \end{bmatrix}^T$ and the approximation point $\bar{\xi}$.

$$\mathcal{L}_i(\bar{\xi}, z) = \frac{1}{2}(\xi - \bar{\xi})^T \frac{\partial^2 \phi_s(\xi_{[1:n]}, \xi_{[n+1:n+m]})}{\partial^2 \xi}\bigg|_{\xi=z} (\xi - \bar{\xi}) \tag{19}$$

The value of z in (19) lies within within the interval bound and the approximation point as in (20).

$$z \in \{\alpha\xi + (1-\alpha)\bar{\xi} | \alpha \in [0,1]\} \tag{20}$$

The Lagrange remainder can be described by an ellipsoidal set and is combined with the linear reachable set by Minkowski addition to create the nonlinear reachable set as described in the following Section.

4.1.1 Algorithmic implementation

A linearization is made at the $(x,u) = (0,0)$ point to create the matrices A, B. Using LMIs to describe the LQR based Lyapunov inequality, a stabilizing linear state feedback controller $u = Kx$ and the quadratic Lyapunov function $V(x) = x^T P x$ are synthesized. The maximum input admissible level set of the Lyapunov function is used as $\mathcal{S}_0$. The ellipsoidal toolbox (ET) [27] uses ellipsoids to calculate reachability for linear systems with convex constraints as described by [25]. ET is used to calculate the reachable set for the closed loop linear system $x(k+1) = (A+BK)x(k)$. The current ellipsoidal set $\mathcal{S}_j$ is overapproximated using intervals, creating a (hyper)box of the extended state $\lfloor \xi \rceil = [\lfloor \mathcal{S}_j \rceil^T, \lfloor U_j \rceil^T]^T$, where the input space $\lfloor U_j \rceil = K \lfloor X_j \rceil$. $\bar{\xi}$ is considered the center of the ellipsoidal sets, hence $\bar{\xi} = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$.

The (hyper)box is constructed using the extreme values of ellipsoidal matrices. The extreme values are calculated using the diagonal terms of the shape matrix $\sqrt{diag(P_j^{-1})}$.

Since ξ, z are interval objects, the remainder (19) results in an interval object $\mathcal{L}_{box}$. The second derivative in (19) is calculated for the points within the (hyper)box $\lfloor \xi \rceil$ using the *hessianinit* function in intlab evaluated for function $f(\xi_{[1:n]}, \xi_{[n+1:n+m]})$. This remainder box $\mathcal{L}_{box}$ is overapproximated by an ellipsoid $\mathcal{S}_{\mathcal{L}j}$ this is done using the method described in [24]. Constructing the linear reachable set $\mathcal{S}_{Lj}$ as an ellipsoid and the nonlinear error set $\mathcal{S}_{\mathcal{L}j}$ as an ellipsoid, it is possible to apply Minkowski addition $\mathcal{S}_j = \mathcal{S}_{Lj} \bigoplus \mathcal{S}_{\mathcal{L}j}$ using ET.

4.1.2 Intlab affine arithmetic

Since the error set $\mathcal{S}_{\mathcal{L}j}$ is constructed using two overapproximations, the reach set for the nonlinear system $\mathcal{S}_j = \mathcal{S}_{Lj} \bigoplus \mathcal{S}_{\mathcal{L}j}$ is constructed very conservatively. An improvement is to use affine interval arithmetic, which is included in intlab. The affari implementation is an alternative method of describing intervals carrying some "memory" causing it to possibly produce better inclusions. Using the affari approximation the eigenvalues $\sqrt{eig(P_j^{-1})}$ are used to create a (hyper)box which is rotated using the singular value decomposition (SVD) afterwards. Since the input set for the calculation of the Lagrange remainder is approximated more closely, the linearization error is less conservatively computed.

4.2 Random evaluation

Another method of reachable set construction is a random evaluation. By evaluating the system for random points an ellipsoidal overapproximation is made. This method is correct for the number of evaluations $T \to \infty$. To have a tractable implementation of this method of reachability, T should be small enough, hence a compromise is made. To have a minimal T, while keeping confidence in the result further research has to be done. The points can be found by uniformly distributed random generation over the interval $[-\sqrt{diag(P_j^{-1})}, \sqrt{diag(P_j^{-1})}]$ to create a $n \times T$ matrix. Each column of points v_{Tt} $\forall t \in [1, T]$, is validated to be inside of $\mathcal{S}_j$. A system evaluation is applied on these points, the result stored in polyhedron $\mathcal{P}_T = \mathbf{Co}\{\phi(v_{T1}, Kv_{T1}), \ldots, \phi(v_{T\bar{T}}, Kv_{T\bar{T}})\}$. The polyhedron is used to find the smallest ellipsoid $\mathcal{E}_{nl}$ as described in [24] using the Schur's complement of (21)

$$v_{\bar{t}}^T Q^{-1} v_{\bar{t}} \leq 1, \quad \forall \bar{t} = 1, \ldots, \bar{T} \tag{21}$$

to find (22) for $Q = P_{nl}^{-1} > 0$

$$\begin{bmatrix} 1 & v_{\bar{t}}^T \\ v_{\bar{t}} & Q \end{bmatrix} \geq 0, \quad \forall \bar{t} = 1, \ldots, \bar{T} \tag{22}$$

such that $\mathcal{P}_T \subseteq \mathcal{E}_{nl}$.

4.3 Terminal cost verification

Since the terminal set sequence $\mathcal{S}$ is calculated using the nonlinear reachable set construction, the original LQR based quadratic Lyapunov function is not guaranteed to hold as the terminal cost function.

A proposed method of terminal cost verification is to include the linearization in the calculation of the reachable sets. First the nonlinear set $\mathcal{E}_{nl} := \{x \ : \ x^T P_{nl} x \leq \alpha\}$ is calculated,

either by Lagrange remainder or random evaluation, then this set is overapproximated by a set corresponding to the LQR inequality. The LMIs

$$\begin{bmatrix} O_j & (AO_j + BY_j)^T & O_j & Y_j^T \\ AO_j + BY_j & O_{j+1} & 0 & 0 \\ O_j & 0 & \bar{\alpha}Q^{-1} & 0 \\ Y_j & 0 & 0 & \bar{\alpha}R^{-1} \end{bmatrix} \geq 0,$$

(13), with the added constraint

$$O_{j+1} \geq O_{nl} \tag{23}$$

are used to minimize the $trace(O_{j+1})$. Here $O_j = P_j^{-1}$, which is the cost function of the current set. Then $P_{j+1} = O_{j+1}^{-1}$ verifies the LQR for the smallest ellipsoidal level set $\mathcal{S}_{j+1} := \{x \ : x^T P_{j+1} x \leq \alpha\}$ containing the nonlinear set.

Terminal feasibility While the reachable sets are calculated without associated terminal cost function, a subsequent verification is applied. This however does not necessarily give feasible results, since the reachable sets are based on approximations.

Table 1: Volume comparison of terminal sets for undamped oscillator.

Set	Volume
Linear $M = 10, \mathcal{S}_{max}$	3.836848414601913
Linear $M = 10, \mathcal{S}_0$	2.985386253003532
pLDI $M = 10, \mathcal{S}_{max}$	4.084069235906584
pLDI $M = 10, \mathcal{S}_0$	3.712899763995698
pLDI $M = 1, \mathcal{S}_0$	4.381223423896796
[8]	1.035606448654549
$\mathcal{S}_p$	1.858045336306206
Random eval. $M = 33, \mathcal{S}_{max}$	5.242066229019890

5 Illustrative examples

In this section the application of the new method and the reachable construction are applied to examples presented in academic papers. All systems have a bound on the $\bar{\alpha} < 1000$. In every provided picture, in light blue (magenta) a plot of closed–loop NMPC trajectories for the initial conditions where the NMPC problem was feasible for the same prediction horizon ($N = 5$) and in red circles the initial conditions where the NMPC problem was not feasible. The terminal sets are shown as green ellipsoids, with the starting set $\mathcal{S}_0$ being black and the last set $\mathcal{S}_{M-1}$ is blue. The trajectories are calculated using ACADO toolkit [28]. With ACADO it is possible to create M mex files for each terminal set and associated cost. The mex files use the optional acado.MexInput, which is used to define the initial state. For each initial state, the closest set is found and this set is used to initialize the position j within the sequence. At each time instant k, the $(j + k)\text{mod}M$ ACADO mex file is used over a N horizon. Initialization of the optimal control problem can be done by ACADO or by use of the associated terminal controller.

Undamped Oscillator The following system is taken from [5] and it describes an undamped oscillator:

$$\phi_1(x, u) = x_1 + 0.1x_2 + 0.09u + 0.01x_1u, \tag{24a}$$
$$\phi_2(x, u) = x_2 + 0.1x_1 + 0.09u - 0.04x_2u. \tag{24b}$$

The stage cost is defined using $Q = \frac{1}{20}I_2, R = 0.1$. The terminal set obtained in [5] (named S_p) is compared with the sequence of terminal sets obtained for $M = 10$ and $M = 1$ in Table 1, using $\kappa_j = \kappa = 0.05$ for all $j = 0, \ldots, M - 1$, $\eta = 0.75$ and $\rho = 1.05$. The terminal set obtained via the standard linearization method with error correction from [8] is included in the comparison.

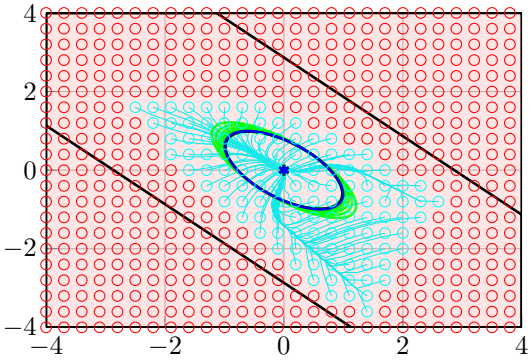


Figure 1: $M = 10$ for the undamped oscillator, first order Taylor approximation, nonlinear terminal control law (11).

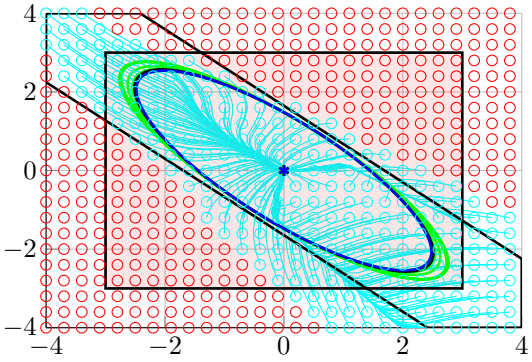


Figure 2: $M = 10$ for the undamped oscillator, quasi–second order Taylor approximation with pLDI bounding, nonlinear terminal control law (11).

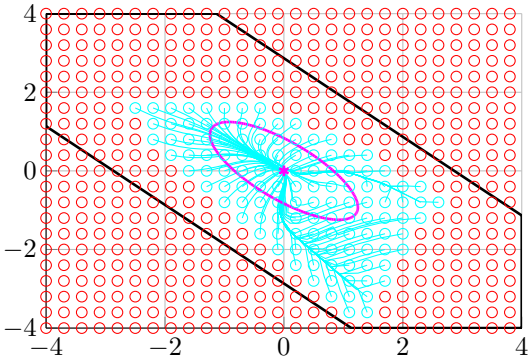


Figure 3: Method of [8] for the undamped oscillator.

The results are better in this case when using a quasi–second order Taylor approximation with pLDI and $M = 1$ because the approximation is exact for bilinear dynamics. The developed

method yields larger terminal sets than the tested existing methods. The proposed method results in a larger domain of attraction in general. Increasing κ_j can be beneficial for the size of standard method set, however influencing the terminal cost and terminal controller for an increased control input.

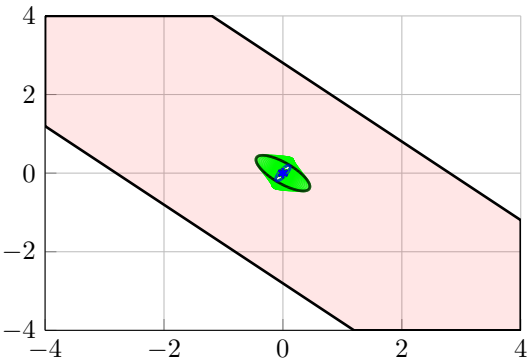


Figure 4: Intlab approximated sequence construction of size $M = 96$.

Figure 4 shows a stabilizing sequence constructed using intlab approximation, in figure 5 the affari approximation is used. Both implementations give a sequence of high M, while the volume compared to the standard [8] method decreases. Therefor the ACADO simulation is omitted. Comparing figure 5 to figure 4, the size is increased, however still smaller than the standard method therefore both intlab and affari are not very suitable for terminal set application.

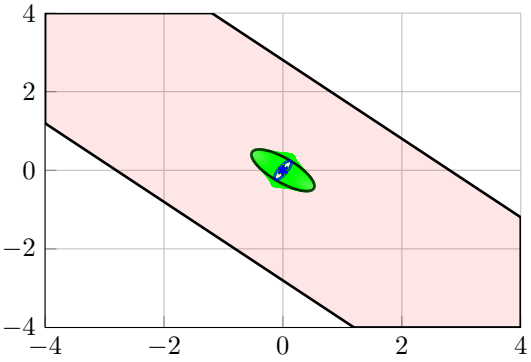


Figure 5: Affari approximated sequence construction of size $M = 113$.

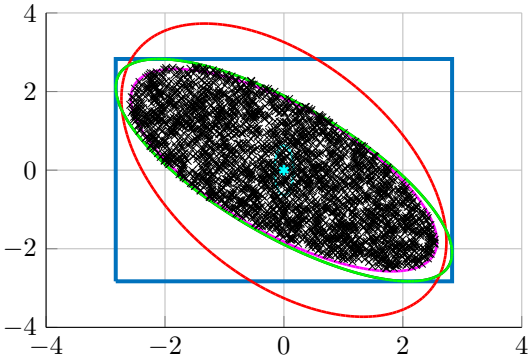


Figure 6: Intval approximation undamped oscillator. A comparison of random evaluation (x), linear reachable set (purple) and the intlab approximation (red).

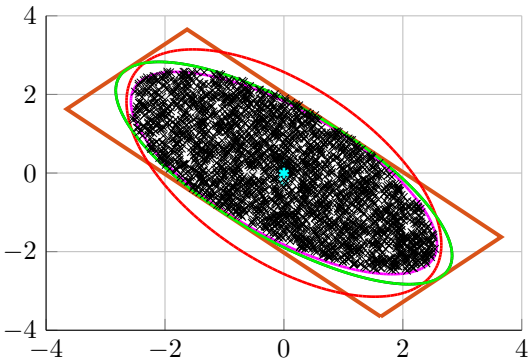


Figure 7: Affari approximation undamped oscillator. A comparison of random evaluation (x), linear reachable set (purple) and the affari approximation (red).

The figures 6 and 7 are used to show the conservativeness of the Lagrange remainder method. The figures show the set $\mathcal{S}_j$(green) and the approximation $\lfloor \mathcal{S}_j \rceil$ together with the random evaluation points (x), the linear reachable set (purple), the Lagrange remainder (cyan) and the twice overapproximated Minkowski addition (red). The implementation of the Lagrange remainder shows the inherent problem with this method, the goal of a large set creates a large linearization error and therefore the twice overapproximated ellipsoidal set (purple $\bigoplus$ cyan = red) is very conservative. The increase in size causes the sequence to leave the input admissible space quickly, therefore the sequence is scaled close to zero, making it less suitable for application in terminal set construction.

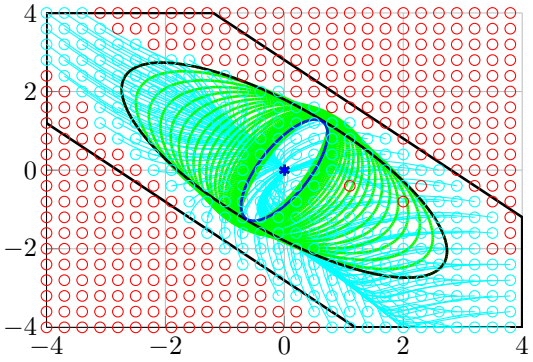


Figure 8: Undamped oscillator random evaluation with $T = 750$ of size $M = 33$.

Since the sequence $\mathcal{S}$ is constructed using random evaluation for $T = 750$ points the computation becomes less tractable as the size of the sequence increases. The shape of the sets, however, show that the sequence is less constricted by the behaviour of the linearization and therefore increase in size as shown in table 1. Since the construction is done by $T < \infty$, the sets are not guaranteed to be reachable sets, therefore the sets are not guaranteed to have feasible solutions. This is noticable in figure 8, where some initial conditions can give infeasible solutions.

Inverted Pendulum The system of [29] is used with a constant $C_m = 14$, $mgL = 2$ and parameters $Q = \frac{1}{20} I_n, R = 0.1, T_s = 0.2, |x| \leq 10, |u| \leq 2$ describes an inverted pendulum:

$$\phi_1(x, u) = x_1 + T_s mgL \sin(x_2) + C_m T_s u, \tag{25a}$$

$$\phi_2(x, u) = x_2 + T_s x_1. \tag{25b}$$

Figure 9 shows a benefit of $M > 1$, as the ellipsoids cover more input admissible space than only one set could, while the use of the nonlinear control law (11) increases the size significantly. This result is for $\kappa_j = 0.05$ while using a first order Taylor approximation.

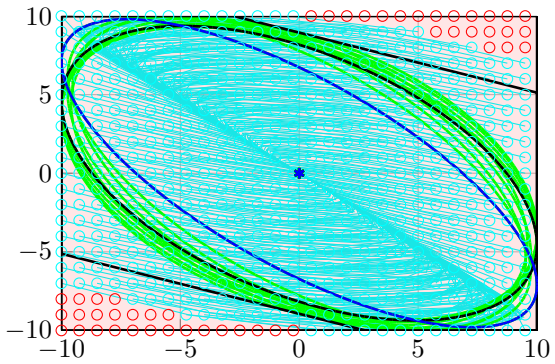


Figure 9: $M = 10$ for the inverted pendulum, first order Taylor approximation, nonlinear terminal control law (11).

Table 2: Volume comparison of terminal sets for inverted pendulum.

Set	Volume
Linear $M = 10, \mathcal{S}_{max}$	89.405147602400604
Linear $M = 10, \mathcal{S}_0$	81.675380231847839
[8]	0.840519749009968
Random eval. $M = 70, \mathcal{S}_{max}$	20.324971273919424

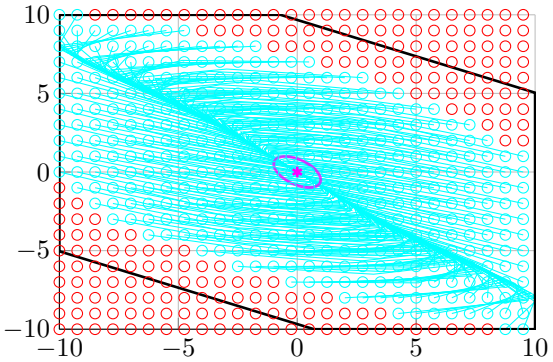


Figure 10: Method of [8] for the inverted pendulum.

In figure 11 the reachable set is calculated by random evaluation with $T = 750$ random points. The sequence shows more movement in the state space and improves the size of the set in terms of [8]. Comparing the result to the method in figure 9, the reachable set construction could possibly gain from an implementation of multiple state feedback controllers.

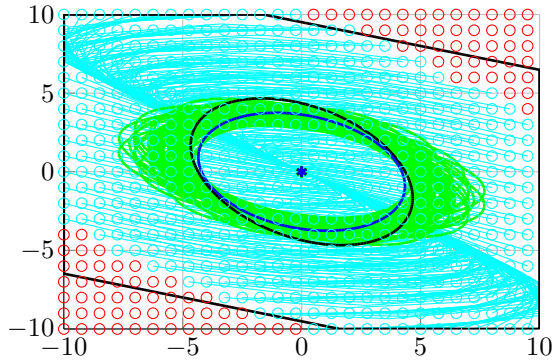


Figure 11: Inverted Pendulum random evaluation with $T = 750$ points and $M = 70$.

Nonlinear spring The example in [15] describes a connected cart system with a nonlinear spring $k = k_0 e^{-x_1}$:

$$\phi_1(x,u) = x_1 + T_s x_2, \tag{26a}$$

$$\phi_2(x,u) = x_2 - T_s \frac{k_0}{M_c} e^{-x_1} x_1 + Ts \frac{h_d}{M_c} x_2 + T_s \frac{u}{M_c}. \tag{26b}$$

The parameters are chosen $k_0 = 0.33, M_c = 1, h_d = 1.1$ and $T_s = 0.4$ and stage cost $Q = 0.01 I_n, R = 10 I_m$ and $|x_1| \leq 2.65, |x_2| \leq 5, |u| \leq 4.5$. The value of $\kappa_j = 0.4$ and $\rho = 1.05$ and $\mu = 1$. This system shows again a benefit of the new method using quasi–second order approximation and nonlinear control law (11), as the sets increase and by movement cover more of the state space.

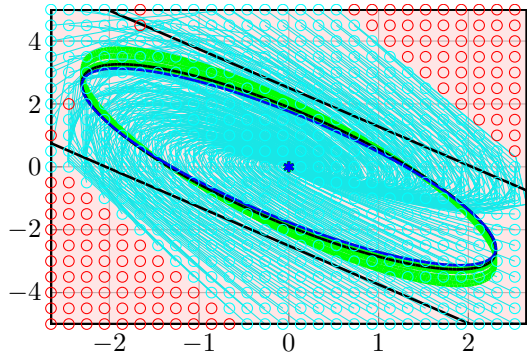


Figure 12: Nonlinear spring cart system, quasi-second order approximation using nonlinear control law (11).

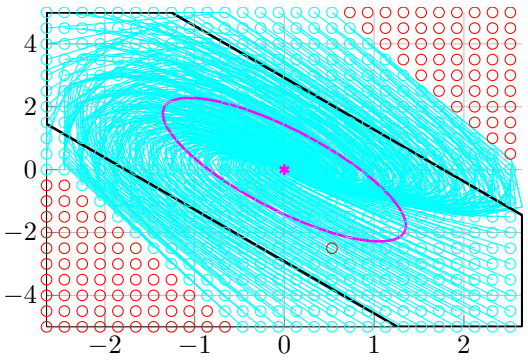


Figure 13: Nonlinear spring cart system, standard [8] method.

Table 3: Volume comparison of terminal sets for nonlinear spring.

Set	Volume
pLDI $M = 10, \mathcal{S}_{max}$	5.002556880598027
pLDI $M = 10, \mathcal{S}_0$	4.360821197278540
pLDI, linear control law $M = 10, \mathcal{S}_{max}$	2.321923276054616
pLDI, linear control law $M = 10, \mathcal{S}_0$	2.024063390452235
[8]	1.926296923378072

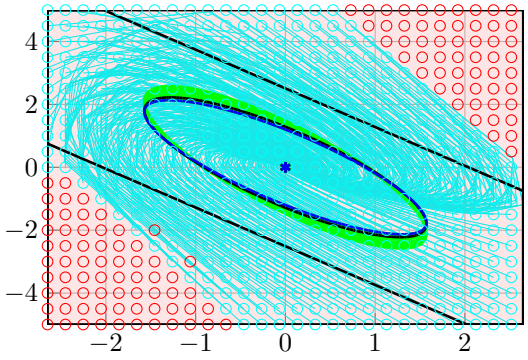


Figure 14: Nonlinear spring cart system, quasi-second order approximation using linear control law.

The figures for this system do not show an immediate increase of the domain of attraction, which can be caused by the coarse selection of initial conditions.

2DOF robotic manipulator The system is [30] describes a robotic manipulator system of order 4:

$$\phi_1(x,u) = x_1 + T_s x_3, \tag{27a}$$
$$\phi_2(x,u) = x_2 + T_s x_4, \tag{27b}$$
$$\begin{bmatrix}\phi_3(x,u)\\ \phi_4(x,u)\end{bmatrix} = \begin{bmatrix}x_3\\ x_4\end{bmatrix} - T_s\left[M^{-1}(c+g)\right] + T_s\begin{bmatrix}u_1\\ u_2\end{bmatrix}, \tag{27c}$$

with

$$M = \begin{bmatrix} b_1 + b_3\cos(x_1 - x_2) & b_2 + b_3\cos(x_1 - x_2) \\ b_3\cos(x_1 - x_2) - b_8 & b_7 \end{bmatrix}, \tag{28a}$$
$$c = \begin{bmatrix} -b_3\sin(x_1 - x_2)\ {x_3}^2 + b_6 x_3 + b_3\sin(x_1 - x_2)\ {x_4}^2 \\ -b_9\ (x_3 - x_4) - b_3 {x_3}^2\sin(x_1 - x_2) \end{bmatrix}, \tag{28b}$$
$$g = \begin{bmatrix} -b_4\sin(x_2) - b_5\sin(x_1) \\ -b_4\sin(x_2) \end{bmatrix}, \tag{28c}$$

and $b_1 = 0.0715, b_2 = 0.0058, b_3 = 0.0114, b_4 = 0.3264, b_5 = 0.3957, b_6 = 0.6254, b_7 = 0.0749, b_8 = 0.0705$ and $b_9 = 1.1261$ The system is discretized using a Euler discretization for $T_s = 0.01$. The stage cost is defined using $Q = diag(1, 1, 0.01, 0.01), R = 0.001 I_m$. The input constraints $|u| \leq 10$ and state constraints $|x| \leq \begin{bmatrix}90\pi/180 & 90\pi/180 & 10 & 10\end{bmatrix}^T$. A $M = 20$ set sequence is created

using $\kappa_j = 0.1$. Figure 15 and 16 show projections on the respective states of the terminal set sequence for visualisation. The largest gain in this example is how the set sequence in figures 15 and 16 increases the coverage of the state space compared to the standard method in figures 17 and 18.

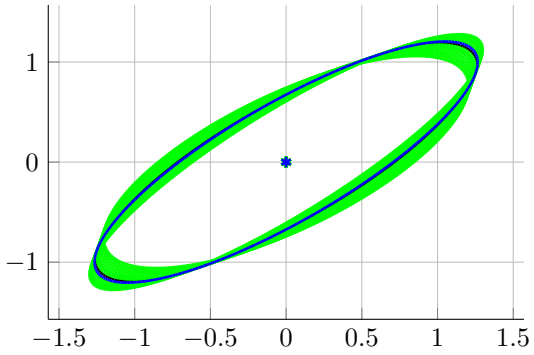


Figure 15: Robotic manipulator, State 1,2

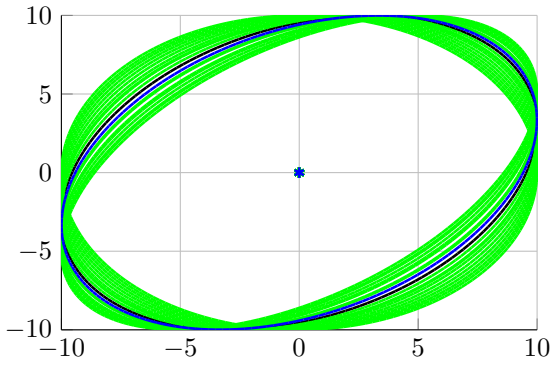


Figure 16: Robotic manipulator, State 3,4

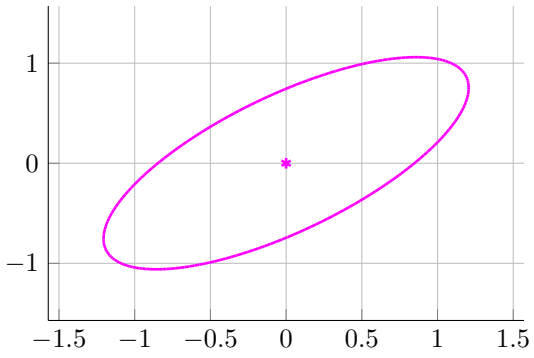


Figure 17: Robotic manipulator, State 1,2

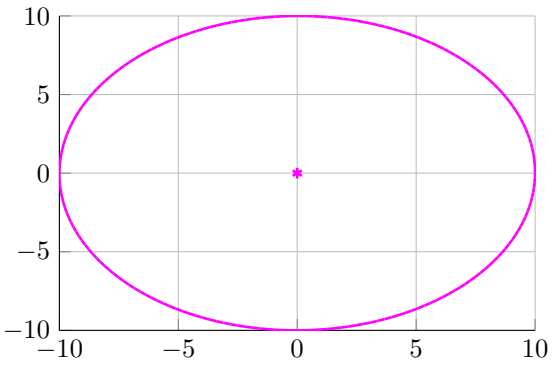


Figure 18: Robotic manipulator, State 3,4

6 Conclusions

In this thesis a new method for designing and computing time–varying terminal costs and terminal sets has been developed for discrete–time nonlinear MPC algorithms with guaranteed stability. First, a first or quasi–second order Taylor approximation of the nonlinear dynamics is employed and LMIs are solved to find a set of periodic quadratic control Lyapunov functions and corresponding ellipsoidal terminal sets. Then a global nonlinear convex optimisation problem is solved over each terminal set to correct for the approximation error and validate the terminal costs and sets for the original nonlinear dynamics. The method was tested on benchmark examples from the NMPC literature with encouraging results. Nonlinear reachable sets with coupled time–varying terminal cost to construct time–varying terminal sets were also researched. Sets constructed using Lagrange remainder remain small and are therefore less suited as terminal sets. The construction using random evaluation shows promising results, however some research should be done to the compromise between scalability and confidence of validity. The reachable set construction could possibly gain from implementation of nonlinear terminal controller(s), where the region of stability should be researched. Future work will deal with further improving the computational aspects of the algorithms, developing an automated software implementation and testing it on more complex examples.

Acknowledgements

I would like to thank my supervisor Dr. M. Lazar for guidance, help and encouragement, but especially for sacrificing part of not one but two holidays to finalize work within strict deadlines. Part of the results in this thesis were reported in a conference paper accepted at the 6th IFAC Conference on Nonlinear Model Predictive Control, Madison, USA, August 2018.

References

[1] H. Michalska and D. Mayne, "Robust receding horizon control of constrained nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 38, no. 11, pp. 1623–1633, 1993.

[2] D. Mayne, "An apologia for stabilising terminal conditions in model predictive control," *International Journal of Control*, vol. 86, no. 11, pp. 2090–2095, nov 2013.

[3] L. Grüne, J. Pannek, L. Grüne, and J. Pannek, *Nonlinear Model Predictive Control*. Springer London, 2011.

[4] D. Limon, T. Alamo, F. Salas, and E. Camacho, "On the stability of constrained MPC without terminal constraint," *IEEE Transactions on Automatic Control*, vol. 51, no. 5, pp. 832–836, may 2006.

[5] M. S. Darup and M. Cannon, "A missing link between nonlinear MPC schemes with guaranteed stability," in *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE, dec 2015.

[6] H. Chen and F. Allgöwer, "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability," *Automatica*, vol. 34, no. 10, pp. 1205–1217, oct 1998.

[7] W.-H. Chen, J. O'Reilly, and D. J. Ballance, "On the terminal region of model predictive control for non-linear systems with input/state constraints," *International Journal of Adaptive Control and Signal Processing*, vol. 17, no. 3, pp. 195–207, 2003.

[8] S. H. H. Wook Hyun Kwon, *Receding Horizon Control: Model Predictive Control for State Models*. SPRINGER VERLAG GMBH, 2005.

[9] S. Yu, C. Hou, T. Qu, and H. Chen, "A revisit to MPC of discrete-time nonlinear systems," in *2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*. IEEE, jun 2015.

[10] X. B. Hu and W. H. Chen, "Model predictive control for non-linear missiles," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 221, no. 8, pp. 1077–1089, nov 2007.

[11] J. Bravo, D. Limon, T. Alamo, and E. Camacho, "On the computation of invariant sets for constrained nonlinear systems: An interval arithmetic approach," *Automatica*, vol. 41, no. 9, pp. 1583–1589, sep 2005.

[12] M. Fiacchini, T. Alamo, and E. Camacho, "On the computation of convex robust control invariant sets for nonlinear systems," *Automatica*, vol. 46, no. 8, pp. 1334–1338, aug 2010.

[13] Y. Lee, M. Cannon, and B. Kouvaritakis, "Extended invariance and its use in model predictive control," *Automatica*, vol. 41, no. 12, pp. 2163–2169, dec 2005.

[14] M. Cannon, B. Kouvaritakis, and V. Deshmukh, "Enlargement of polytopic terminal region in NMPC by interpolation and partial invariance," *Automatica*, vol. 40, no. 2, pp. 311–317, feb 2004.

[15] D. M. Raimondo, "Nonlinear model predictive control: Stability, robustness and applications," Master's thesis, UNIVERSITÀ DEGLI STUDI DI PAVIA, 2009.

[16] S. Lucia, P. Rumschinski, A. J. Krener, and R. Findeisen, "Improved design of nonlinear model predictive controllers," *IFAC-PapersOnLine*, vol. 48, no. 23, pp. 254–259, 2015.

[17] M. Kögel and R. Findeisen, "Stability of NMPC with cyclic horizons," *IFAC Proceedings Volumes*, vol. 46, no. 23, pp. 809–814, 2013.

[18] M. Lazar and V. Spinu, "Finite-step terminal ingredients for stabilizing model predictive control," *IFAC-PapersOnLine*, vol. 48, no. 23, pp. 9–15, 2015.

[19] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, apr 2005.

[20] R. H. Byrd, J. Nocedal, and R. A. Waltz, "KNITRO: An integrated package for nonlinear optimization," *In G. di Pillo and M. Roma, editors, Large-Scale Nonlinear Optimization*, vol. Springer, pp. 35–59, 2006.

[21] M. Tawarmalani and N. V. Sahinidis, "A polyhedral branch-and-cut approach to global optimization," *Mathematical Programming*, vol. 103, no. 2, pp. 225–249, may 2005.

[22] J. Löfberg, "Yalmip : A toolbox for modeling and optimization in matlab," in *In Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[23] R. H. Tütüncü, K. C. Toh, and M. J. Todd, "Solving semidefinite-quadratic-linear programs using SDPT3," *Mathematical Programming Series B*, vol. 95, pp. 189–217, 2003.

[24] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Society for Industrial and Applied Mathematics, jan 1994.

[25] A. A. Kurzhanskiy and P. Varaiya, "Ellipsoidal techniques for reachability analysis of discrete-time linear systems," *IEEE Transactions on Automatic Control*, vol. 52, no. 1, pp. 26–38, jan 2007.

[26] L. Asselborn, D. Groß, and O. Stursberg, "Control of uncertain nonlinear systems using ellipsoidal reachability calculus," *IFAC Proceedings Volumes*, vol. 46, no. 23, pp. 50–55, 2013.

[27] A. A. Kurzhanskiy and P. Varaiya, "Ellipsoidal toolbox," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2006-46, May 2006.

[28] B. Houska, H. Ferreau, and M. Diehl, "ACADO Toolkit – An Open Source Framework for Automatic Control and Dynamic Optimization," *Optimal Control Applications and Methods*, vol. 32, no. 3, pp. 298–312, 2011.

[29] S. Iles, M. Lazar, and J. Matusko, "Stabilizing model predictive control based on flexible set-membership constraints," in *2015 European Control Conference (ECC)*. IEEE, jul 2015.

[30] P. G. Cisneros and H. Werner, "Fast nonlinear MPC for reference tracking subject to nonlinear constraints via quasi-LPV representations," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 11 601–11 606, jul 2017.