Reference Tracking MPC Using Dynamic Terminal Set Transformation

Daniel Simon, Johan Löfberg and Torkel Glad

Linköping University Post Print

N.B.: When citing this work, cite the original article.

http://dx.doi.org/10.1109/TAC.2014.2313767

©2014 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Postprint available at: Linköping University Electronic Press
http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-112045
Reference Tracking MPC using Dynamic Terminal Set Transformation

Daniel Simon, Johan Löfberg, Torkel Glad

Published Oktober 2014

Abstract

Among the many different formulations of Model Predictive Control (MPC) with guaranteed stability, one that has attracted significant attention is the formulation with a terminal cost and terminal constraint set, the so called dual mode formulation.

In this paper our goal is to make minimal changes to the dual mode framework, for the linear polytopic case, in order to develop a flexible reference tracking algorithm with guaranteed stability and low complexity, which is intuitive and easily understood. The main idea is to introduce a scaling variable that dynamically scales the terminal constraint set and therefore allows it to be centered around an arbitrary setpoint without violating the stability conditions. The main benefit of the algorithm is reduced complexity of the resulting QP compared to other state of art methods without loosing performance.

1 Introduction

Input and state constrained optimal control problems are in most cases impossible to solve with an explicit feedback policy (Mayne et al., 2000). Model Predictive Control (MPC) offers a way to solve these problems by iteratively solving a constrained optimization problem online, using the current state of the system as initial condition.

Due to the iterative nature of MPC one must take special measures to ensure that the optimization problem remains feasible and that the controller stabilizes the system. The main goal of this paper is to develop a flexible reference tracking algorithm with guaranteed stability, which is intuitive and easily understood. Parts of the proposed algorithm have previously been published in Simon et al. (2012). In this paper we extend the theory and develop a dual form of the algorithm to avoid the, possibly complicated, vertex enumeration previously present in the algorithm, and compare the proposed algorithm to existing state of art MPC controllers.

In Section 2 an introduction to the MPC formulation, its stability properties and reference tracking formulation is outlined. We present some preliminary results to the proposed controller in Section 3 and then in Section 4 the proposed reference tracking algorithm is outlined, the main theorem of the paper and the proof of stability and convergence is presented. The paper finishes in Section 5 with an illustrative example from the aircraft industry to illustrate the properties of the proposed controller.

2 MPC, Stability and Tracking

For a discrete time system of the form

\[ x_{k+1} = f(x_k, u_k) \] 

with state constraints \( x_k \in \mathcal{X} \) and control constraints \( u_k \in \mathcal{U} \), the Model Predictive Control (MPC) law, \( \nu(x_k) \), is defined through the solution of a finite horizon optimization problem solved in a receding horizon fashion. Among the different formulations of the optimization problem, the one that is most widely used (at least in academia) is the formulation with a terminal state penalty \( \Psi(x_{k+N}) \) and a terminal state constraint \( x_{k+N} \in \mathcal{T} \), see Sznaier and Damborg (1987).

\[
\min_{u, x} \Psi(x_{k+N}) + \sum_{i=0}^{N-1} \ell(x_{k+i}, u_{k+i}) \tag{2a}
\]
\[
x_{k+i+1} = f(x_{k+i}, u_{k+i}) \quad i = 0, \ldots, N - 1 \tag{2b}
\]
\[
x_{k+i} \in \mathcal{X} \quad i = 0, \ldots, N - 1 \tag{2c}
\]
\[
u(x_k) = u_k^* \tag{2f}
\]

This formulation is also referred to as the dual mode formulation since, theoretically, the objective of the MPC controller is to steer the system state to a region around the origin (terminal set) where a local feedback policy takes over. This impose some specific requirements on the terminal set, \( \mathcal{T} \).

**Definition 1** (Blanchini, 1999)). The set \( \mathcal{T} \subset \mathbb{R}^n \) is said to be positively invariant for a system \( x_{k+1} = f(x_k) \) if for all \( x_k \in \mathcal{T} \) the solution \( x_{k+i} \in \mathcal{T} \) for \( i > 0 \).

Given the above MPC formulation we can now present the classical stability results for MPC which can be found, e.g., in Mayne et al. (2000).

**Theorem 1.** For the system (1) controlled with the MPC law defined through (2) the closed loop system is stable and asymptotically converges to the origin if the following conditions hold:

(a) \( \mathcal{T} \) is a positively invariant set of the system (1) controlled with the feedback \( u_k = \kappa(x_k) \) where \( \kappa(x_k) \in U \forall x_k \in \mathcal{T} \)

(b) \( \mathcal{T} \subseteq \mathcal{X} \) with \( 0 \in \mathcal{T} \)

(c) the stage cost \( \ell(x_k, u_k) \) is positive definite, and \( \Delta \Psi(x_k) + \ell(x_k, \kappa(x_k)) \leq 0 \), \( \forall x_k \in \mathcal{T} \), where \( \Delta \Psi(x_k) = \Psi(x_{k+1}) - \Psi(x_k) \).

The properties of \( \Psi \) can be obtained if \( \Psi \) is chosen to be a Lyapunov function upper bounding the infinite horizon cost when using the controller \( \kappa(x_k) \).

## 2.1 Reference tracking

When tracking a reference signal, i.e., the so called servo problem, the system shall not converge to the origin but settle at some steady state \((x_r, u_r)\) different from the origin, yielding the desired output. At this steady state it shall hold that \( y_k = r \), where \( r \) is the external reference to be followed. A reasonable choice of steady state is the minimal norm input, which can be formulated as a convex problem, (Muske and Rawlings, 1993)

\[
\begin{align*}
\min_{x_r, u_r} & \quad u_r^T R_s u_r \tag{3a} \\
\text{s.t.} & \quad \begin{bmatrix} A - I & B \\ C & D \end{bmatrix} \begin{bmatrix} x_r \\ u_r \end{bmatrix} = \begin{bmatrix} 0 \\ r \end{bmatrix} \tag{3b} \\
& \quad u_r \in U \tag{3c}
\end{align*}
\]

If the matrix on the left hand side of the equality constraint is square and nonsingular, then the problem (3) has the trivial solution

\[
\begin{bmatrix} x_r \\ u_r \end{bmatrix} = \begin{bmatrix} A - I & B \\ C & D \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ r \end{bmatrix} \tag{4}
\]

which means that there exist a unique linear mapping between the reference signal and the steady state of the system.

Instead of using the actual setpoint in the controller, a better way is to use a so called pseudo setpoint (Rossiter, 2006) which increases the state space region in which the optimization problem is feasible. Instead of using the true reference \( r \) in (3) one introduces a new variable \( r \) which gives a corresponding steady state and control, \( \bar{x} \) and \( \bar{u} \), and then penalize the deviation between the desired
reference $r$ and the pseudo reference $\bar{r}$ using a positive definite function $\phi(\bar{r} - r)$ in the objective function of (2).

Since a simple change of coordinates will shift the origin of the problem to the new setpoint, a pragmatic approach to implement the reference tracking case would be to apply the standard MPC scheme (2) on the translated system, i.e., in the original coordinate system, penalize deviations from the steady state setpoints. This approach has been analyzed for the infinite horizon case in, e.g., Muske and Rawlings (1993) and Rawlings et al. (1994). For nonlinear systems simulation results are presented in Meadows and Badgwell (1998), however without any detailed stability analysis.

In the dual mode formulation the terminal state constraint set now depends on the setpoint and it must hold that the terminal set fulfills the conditions (a) and (b) in Theorem 1 for all possible setpoints. A straightforward way would be to recalculate the terminal constraint set online based on the current setpoint, which is implicitly assumed in, e.g., Rao and Rawlings (1999). To recalculate the terminal constraint set can however require complex calculations to be performed online and might thus not be suitable for systems where the setpoint changes often.

A far better approach has been proposed in different forms by several authors Chisci and Zappa (2003), Limon et al. (2008), Ferramosca et al. (2009). In Chisci and Zappa (2003) the authors augment the system with a new constant state which corresponds to the reference signal giving a terminal constraint set in the higher dimension $(x, r)$ which contain the whole feasible equilibrium set for any given reference. Instead of using the reference as augmented state the authors of Limon et al. (2008), Ferramosca et al. (2009) introduce a new optimization variable, $\theta$, which spans the null space of steady state equation. The resulting controller guarantees feasibility and stability and has a larger domain of attraction than the standard MPC tracking controller.

3 Preliminaries To Main Result

Before we derive the new tracking algorithm let us first state some necessary assumptions, facts and definitions.

**Definition 2.** For a polytopic set $\mathcal{P}$ with the halfspace representation

$$\mathcal{P} = \{x \mid Ax \leq b\}$$

we define

- the scaling of the set with a positive scalar $\alpha$ as
  $$\alpha \mathcal{P} \triangleq \{x \mid Ax \leq \alpha b\}$$

- the translation of the set to an arbitrary point $y$ as
  $$\mathcal{P}(y) \triangleq \mathcal{P} \oplus y = \{x \mid A(x - y) \leq b\}$$

**Definition 3.** For a given constant $0 \leq \epsilon \leq 1$ and set $\mathcal{P}$ containing the origin, let $\text{int}_\epsilon$ denote the $\epsilon$-interior of $\mathcal{P}$, i.e.,

$$\text{int}_\epsilon(\mathcal{P}) \triangleq (1 - \epsilon)\mathcal{P} = \{x \mid Ax \leq (1 - \epsilon)b\}$$

**Definition 4.** $r_\perp$ is defined as the closest strictly feasible point to the reference $r$ in a distance measure given by $\phi(\cdot)$.

$$r_\perp = \arg \min_r \phi(\bar{r} - r)$$

subject to

$$\begin{bmatrix} A - I & B \\ C & D \end{bmatrix} \begin{bmatrix} \bar{x} \\ \bar{u} \end{bmatrix} = \begin{bmatrix} 0 \\ \bar{r} \end{bmatrix}$$

$$\bar{x} \in \text{int}_\epsilon(X)$$

$$\bar{u} \in \text{int}_\epsilon(U)$$

**Assumption 1.** The matrix

$$\begin{bmatrix} A - I & B \\ C & D \end{bmatrix}$$

is such that there exists a solution to (3) for any reference $r$. 

3
Assumption 2. $\mathcal{X}$, $\mathcal{U}$ and $\mathcal{T}$ contain the origin in their interior.

Lemma 1. Let $\mathcal{T}$ be a positively invariant set for a stable linear system, $x_{k+1} = Ax_k$. Then for any scalar $\lambda \geq 0$, the scaled set, $\lambda \mathcal{T}$ is also a positively invariant set for the system.

Proof. The lemma follows directly from the scaling invariance of linear systems, see, e.g., Blanchini (1999).

With these definitions and assumption in order, we are ready to formulate the proposed tracking algorithm.

4 The Proposed Controller

Instead of augmenting the system with a new state as in Chisci and Zappa (2003), Limon et al. (2008) the main concept in this paper is to introduce an extra optimization variable, $\lambda_k$, which scales the terminal set, $\mathcal{T}$. The scaling allows us to move the terminal constraint set to an arbitrary setpoint, since the terminal set can be scaled down to a point, i.e., $\mathcal{T} = \bar{x}_k$, eliminating the need for any online recalculation of $\mathcal{T}$ as in Rao and Rawlings (1999) and the terminal constraint set can possibly be far simpler than the terminal set for an augmented system.

The proposed terminal state constraint is a scaled and translated version of the terminal constraint set (2e)

$$x_{k+N} \in \lambda_k \mathcal{T}(\bar{x}_k)$$

(5)

The variable $\lambda_k$ is a non-negative scalar and the constraints on how it can be chosen are such that the conditions of Theorem 1 hold, i.e., the terminal set must be positively invariant (which it is due to Lemma 1) and the state and control constraints are satisfied in $\mathcal{T}$. To ensure this the following constraints must be added

$$\lambda_k \mathcal{T}(\bar{x}_k) \subseteq \mathcal{X}$$

(6a)

$$\kappa(x, \bar{x}_k, \bar{u}_k) \in \mathcal{U} \forall x \in \lambda_k \mathcal{T}(\bar{x}_k)$$

(6b)

The first constraint stating that the scaled and translated terminal set is a subset of the feasible state space, i.e., state constraints are satisfied in $\lambda_k \mathcal{T}(\bar{x}_k)$, and the second constraint stating that the predefined stabilizing controller $\kappa(\cdot)$ fulfills the control constraints for all $x$ in $\lambda_k \mathcal{T}(\bar{x}_k)$.

Modifying the standard MPC tracking algorithm with the above constraints we arrive at the following high-level representation of the optimization problem, which we soon will show can be written as a quadratic program

$$\min_{u,x,\lambda_k,\bar{x}_k,\bar{u}_k,\bar{r}_k} \Psi(x_{k+N} - \bar{x}_k) + \phi(\bar{r}_k - r)$$

(7a)

$$+ \sum_{i=0}^{N-1} \ell(x_{k+i} - \bar{x}_k, u_{k+i} - \bar{u}_k)$$

$$\text{s.t.}$$

$$x_{k+i+1} = Ax_{k+i} + Bu_{k+i}$$

(7b)

$$x_{k+i} \in \mathcal{X}$$

(7c)

$$u_{k+i} \in \mathcal{U}$$

(7d)

$$x_{k+N} \in \lambda_k \mathcal{T}(\bar{x}_k)$$

(7e)

$$\lambda_k \mathcal{T}(\bar{x}_k) \subseteq \mathcal{X}$$

(7f)

$$\bar{u}_k - K(x - \bar{x}_k) \in \mathcal{U} \forall x \in \lambda_k \mathcal{T}(\bar{x}_k)$$

(7g)

$$\begin{bmatrix} A - I & B \\ C & D \end{bmatrix} \begin{bmatrix} \bar{x}_k \\ \bar{u}_k \end{bmatrix} = \begin{bmatrix} 0 \\ \bar{r}_k \end{bmatrix}$$

(7h)

$$\bar{x}_k \in \text{int}_\epsilon(\mathcal{X})$$

(7i)

$$\bar{u}_k \in \text{int}_\epsilon(\mathcal{U})$$

(7j)

for all $i = 0, \ldots, N - 1$. 


In order to guarantee the convergence properties of the controller, our proof in section 4.3 requires the steady states and controls to be strictly feasible, i.e., the terminal set has to be placed in the strict interior of the feasible set. From a practical point of view we achieve this by constraining them to an \( \epsilon \)-interior of the feasible set, constraints (7i) and (7j). Simulations indicate that for practical purposes any arbitrarily small \( \epsilon \) can be chosen, even \( \epsilon = 0 \).

### 4.1 Quadratic Program Formulation

From an implementation perspective it is beneficial if the algorithm can be rewritten as a quadratic program. As the constraints (5) and (6a) are written now, it is not obvious that they are linear in \( \lambda \). Let us begin with (5) which models that the terminal state is inside the scaled and translated terminal set.

As the constraints (9) are derived now the main drawback is that they require the enumeration of all vertices in the terminal set. This can be formulated as the following optimization problem

\[
F_T(x_k + N - \bar{x}_k) \leq \lambda_k b_T
\]  

(8)

The constraints (6a), which ensure the scaled and translated terminal set to be state feasible, and (6b), which ensures that the nominal control law \( \bar{u}_k - K(x - \bar{x}_k) \) is feasible for any \( x \) in the scaled and translated terminal set, can be rewritten using the vertex form of the terminal set \( T \).

Let \( T \) have \( v_p \) vertices \( v_j \) and it follows that \( \lambda_k T(\bar{x}_k) \) can be represented as the convex hull \( \text{conv}(\bar{x}_k + \lambda_k v_j) \). By convexity, this polytope is a subset of \( X \) if and only if all vertices are in the set. With \( X = \{ x \mid F_X x \leq b_X \} \) we arrive at

\[
F_X(\bar{x}_k + \lambda_k v_j) \leq b_X \forall j = 1, \ldots, v_p
\]

(9a)

Similarly for the control constraints, we have to ensure that \( F_U(\bar{u}_k - K(x - \bar{x}_k)) \leq b_U \) for all \( x \in \text{conv}(\bar{x}_k + \lambda_k v_j) \). Checking the vertices leads to

\[
F_X(\bar{u}_k - \lambda_k v_j) \leq b_U \forall j = 1, \ldots, v_p
\]

(9b)

As the constraints (9a) and (9b) are derived now the main drawback is that they require the enumeration of all vertices in the terminal set. This can, even for simple polytopes, be extremely expensive (Grünaun, 2003), e.g., in the case when \( T \) is a simple box in \( \mathbb{R}^n \) it has \( 2^n \) facets but \( 2^n \) vertices and for higher dimensions this difference becomes crucial from a computational complexity point of view. Therefore we seek to avoid doing the vertex calculations and this can be done by instead rewriting the constraints (9a) and (9b) using duality theory.

### 4.2 Dual Formulation of Terminal Set Constraints

First note that the constraints (9a) and (9b) can be interpreted as uncertain constraints, i.e., they must hold for all \( x \in \lambda_k T(\bar{x}_k) \) and we can write them as

\[
F_U(\bar{u}_k - K(x - \bar{x}_k)) \leq b_U \quad F_X x \leq b_X
\]

(10a)

for all \( x \) such that

\[
\{ x \mid F_T(x - \bar{x}_k) \leq \lambda_k b_T \}
\]

(10b)

Now consider a general uncertain affine constraint of the form

\[
(e^T x + d) + (Ax + b)^T p \leq 0 \quad \forall F^T x \leq E
\]

(11)

where \( x \) is an uncertain, but bounded, variable and \( p \) is the optimization variable. This constraint must hold for all possible values of \( x \), i.e., it must hold for the worst case \( x \), in this case, the one that maximize the left hand side.

This can be formulated as the following optimization problem

\[
\max_x (A^T p + c)^T x + (b^T p + d) \quad \text{s.t. } F^T x \leq E
\]

(12)
Additionally we know that for a solution \( p \) to satisfy (11) it must hold that the left hand side is negative and hence also gives a negative optimal cost for the optimization problem (12).

From duality theory, see, e.g., Boyd and Vandenberghe (2004) or Nesterov and Nemirovskii (1994), the dual problem of (12) can be formulated in the dual variable \( \zeta \) as

\[
\min_{\zeta} E^T \zeta + (b^T p + d) \quad \text{s.t.} \quad \zeta \geq 0, \quad F \zeta = A^T p + c \tag{13}
\]

It then follows that the optimal value of the dual problem, which is an upper bound to the primal problem, must be negative. Thus we can conclude that given a solution \( p \) and the existence of a dual variable \( \zeta \geq 0 \) such that

\[
E^T \zeta + (b^T p + d) \leq 0, \quad F \zeta = A^T p + c \tag{14}
\]

is equivalent to that \( p \) satisfies the constraint (11) for all values of the uncertain variable \( x \). This derivation has previously been presented in, e.g., Ben-Tal and Nemirovski (2002).

Let us now use this result to rewrite the constraints (9a) and (9b) into the dual form. Denote the decision variable \( p_k = (\bar{x}_k, \bar{u}_k)^T \) and write each row of (10) generically as

\[
f^T_x x + f^T_p p_k + d \leq 0 \quad \forall \ x : F_T (x - \bar{x}_k) \leq \lambda_k b_T \tag{15}
\]

where \( f^T_x \) denotes the rows of the matrix

\[
F_x = \begin{bmatrix} -F_{xT} K \\ F_{xT} \end{bmatrix}
\]

and \( f^T_p \) denotes the rows of

\[
F_p = \begin{bmatrix} F_{xT} K & F_{xT} \\ 0 & 0 \end{bmatrix}
\]

and \( d \) are the corresponding row element of the vector \((b^T_{\bar{u}} b^T_{x})^T\).

Writing this in the same structure as (11) gives

\[
(f^T_x x + d) + (0x + f^T_p p_k) \leq 0 \quad \forall \ F_x x \leq \lambda_k b_T + F_T \bar{x}_k \tag{16}
\]

and the dual form then becomes

\[
(\lambda_k b_T + F_T \bar{x}_k)^T \zeta + f^T_p p_k + d \leq 0, \quad F_T^T \zeta = f_x, \quad \zeta \geq 0 \tag{17}
\]

Expand the parenthesis and use the fact that \( F_T^T \zeta = f_x \), this gives

\[
\lambda_k b^T_{f_T} \zeta + \bar{x}_k^T f_x + f^T_p p_k + d \leq 0, \quad F_T^T \zeta = f_x, \quad \zeta \geq 0 \tag{17}
\]

Hence, we should have a global solution \((\lambda_k, p_k)\) such that there exist a \( \zeta \) for every row satisfying this set of equations. Note that the term \( b^T_{f_T} \zeta \) is non-negative (both \( b^T_{f_T} \) and \( \zeta \) is non-negative) since the terminal set contains the origin in its interior. Hence, it is beneficial to make this term minimal for every row. Note also that this choice of \( b^T_{f_T} \zeta \) is independent of the variables \( \lambda_k, p_k \). Therefore let

\[
\gamma = \min_{\zeta} b^T_{f_T} \zeta \quad \text{s.t.} \quad F_T^T \zeta = f_x, \quad \zeta \geq 0 \tag{15}
\]

and replace the term \( b^T_{f_T} \zeta \) with the solution \( \gamma \), finally giving

\[
\lambda_k \gamma + \bar{x}_k^T f_x + f^T_p p_k + d \leq 0 \tag{16}
\]

We can now replace each row of the uncertain constraints (9a) and (9b) with (16) and if we write this into a matrix form we obtain

\[
\lambda_k \Gamma + \bar{x}_k^T \begin{bmatrix} -F_{xT} K \\ F_{xT} \end{bmatrix} + \begin{bmatrix} F_{xT} K & F_{xT} \\ 0 & 0 \end{bmatrix} \bar{u}_k \leq \begin{bmatrix} b_{\bar{x}} \\ b_{x} \end{bmatrix} \tag{17}
\]

where the vector \( \Gamma \) has the solution \( \gamma \) to (15), for each \( f_x \) of \( F_x \), as its elements.
We have now a complete description of the terminal state constraint from (8) and (17). The resulting quadratic program becomes

$$\begin{align*}
\min_{u, x, \lambda_k, \bar{x}_k, \bar{u}_k, r_k} & \Psi(x_{k+N} - \bar{x}_k) + \phi(\bar{r}_k - r) \\
+ & \sum_{i=0}^{N-1} \ell(x_{k+i} - \bar{x}_k, u_{k+i} - \bar{u}_k) \\
\text{s.t.} & \quad x_{k+i+1} = Ax_{k+i} + Bu_{k+i} \\
& \quad F_X x_{k+i} \leq b_X \\
& \quad F_u u_{k+i} \leq b_u \\
& \quad F_T(x_{k+N} - \bar{x}_k) \leq \lambda_k b_T \\
& \quad \lambda_k \Gamma + \bar{x}_T F_T^T + F_P \begin{bmatrix} \bar{x}_k \\ \bar{u}_k \end{bmatrix} \leq \begin{bmatrix} b_u \\ b_X \end{bmatrix} \\
& \quad [A - I B] \begin{bmatrix} \bar{x}_k \\ \bar{u}_k \end{bmatrix} = \begin{bmatrix} 0 \\ \bar{r}_k \end{bmatrix} \\
& \quad F_X(x_{k+i}) \leq (1 - \epsilon) b_X, \quad \epsilon > 0 \\
& \quad F_u(\bar{u}_k) \leq (1 - \epsilon) b_u, \quad \epsilon > 0
\end{align*}$$

for all $i = 0, \ldots, N - 1$ and where

$$\begin{align*}
\Psi(x_{k+N} - \bar{x}_k) &= ||x_{k+N} - \bar{x}_k||_P^2 \\
\phi(\bar{r}_k - r) &= \beta ||\bar{r}_k - r||_\infty \\
\ell(x_{k+i} - \bar{x}_k, u_{k+i} - \bar{u}_k) &= ||x_{k+i} - \bar{x}_k||_Q^2 + ||u_{k+i} - \bar{u}_k||_R^2
\end{align*}$$

Solving the above QP gives an optimal control input sequence $\{u_{k+i}^*\}_{i=0}^{N-1}$ and the applied MPC law is

$$\nu(x_k) = u_k^*$$

In (21), $Q$ and $R$ are positive definite weight matrices, used also to define the Lyapunov cost matrix $P$ in (19) and nominal state feedback $K$ through

$$(A - BK)^T P (A - BK) - P = -Q - K^T RK$$

In (20), $\beta$ is a positive scalar, $r$ is the desired reference to track and $\bar{r}_k$ is the pseudo reference variable.

### 4.3 Stability and Feasibility

In the following theorem we establish the stability and convergence properties of the proposed controller that was derived in the previous section.

The proof of the theorem is relatively straightforward and recursive feasibility and convergence of the state to a stationary point follow standard proofs found in the literature. In a second step, to show that the setpoint to which the state converges is the setpoint associated with the given reference $r$, if feasible, or the setpoint corresponding to the closest feasible reference sense has been proven, e.g., in Ferramosca et al. (2009). Therefore in this paper we only outline the key details.

**Theorem 2.** For any feasible initial state $x_0$, the MPC algorithm (18) and (22) remains feasible and stabilizes the system

$$x_{k+1} = Ax_k + Bu_k$$

Additionally, $x_k$ asymptotically converges to a setpoint given by a weighted infinity norm projection of the reference $r$ onto the ($\epsilon$-contracted) feasible set.
Proof. Assume there exist an optimal solution given by the sequence \{u^*_k\}_{i=0}^{N-1} and \lambda^*_k and \bar{r}^*_k at time \(k\).

Since \(\lambda^*_k\mathcal{T}\) is positively invariant w.r.t the system \(x_{k+1} = (A - BK)x_k\) according to Lemma 1, it is feasible, at the next time step, to use \(\lambda_{k+1} = \lambda^*_k\) and \(\bar{r}_{k+1} = \bar{r}^*_k\). Keeping \(\lambda_{k+1}\) and \(\bar{r}_{k+1}\) unchanged means that we keep the scaled and translated terminal set unchanged. Then recursive feasibility follows from the dual mode formulation.

Using the feasible, but possibly suboptimal, solution sequence \{u^*_k, u^*_{k+2}, \ldots, u^*_N, \bar{u} - K(\bar{x} - x_{k+N})\}, \(\lambda_{k+1} = \lambda^*_k\) and \(\bar{r}_{k+1} = \bar{r}^*_k\) at the next time step, \(k + 1\), then standard Lyapunov argument can be used to show that the cost, \(J_{k+1}\), equals

\[ J_{k+1} = J_k - ||x_k^* - \bar{x}^*_k||_Q^2 - ||u_k^* - \bar{u}^*_k||_R^2 \]

where \(J_k^*\) is the optimal cost at the previous iteration. This implies that the optimal cost at \(k + 1\) is

\[ J_k^* \leq J_{k+1} = J_k^* - ||x_k^* - \bar{x}^*_k||_Q^2 - ||u_k^* - \bar{u}^*_k||_R^2 \]

In other words, \(J_k^*\) is strictly decreasing as long as \(x_k^* \neq \bar{x}^*_k\) and \(u_k^* \neq \bar{u}^*_k\). Hence \(x_k^* \to \bar{x}^*_k\) and \(u_k^* \to \bar{u}^*_k\). Note that in the limit we have, since \(\bar{x}^*_k\) and \(\bar{u}^*_k\) represent a stationary pair, that \(\bar{x}^*_k = x^*_{k+1} = \bar{x}^*_k\), i.e., the pseudo setpoint converges too.

Now assume that the system has settled at a setpoint given by \(\bar{x}^*_k, \bar{u}^*_k\), defined by \(\bar{r}^*_k\) and assume \(\bar{r}^*_k \neq r_\perp\). Consider a perturbation \((0 \leq \gamma < 1)\) of the pseudo reference \(\bar{r}^*_k\) towards \(r_\perp\), given by

\[ \bar{r}_\gamma = \gamma \bar{r}^*_k + (1 - \gamma)r_\perp \]

By convexity, \(\bar{r}_\gamma\) is feasible with respect to \((18h)\) and \((18i)\). We use the constant control sequence corresponding to the steady state control given by \(\bar{r}_\gamma\), i.e., \(u_{k+i} = \bar{u}_\gamma\) (which also is feasible by convexity) and the predicted states evolve according to

\[ x_{k+i} = \bar{x}_\gamma + (1 - \gamma)A^T(\bar{x}^*_k - x_\perp) \]

For \(\gamma\) sufficiently close to 1 all predicted states are feasible w.r.t \(\mathcal{X}\), since \(\bar{x}^*_k\) and \(x_\perp\) (and thus \(\bar{x}_\gamma\)) are strictly inside \(\mathcal{X}\) and \((1 - \gamma)A^T(\bar{x}^*_k - x_\perp)\) approaches 0 as \(\gamma\) goes to 1.

The objective function for our proposed feasible solution, using the notation \(\Psi_i = A^T(\bar{x}^*_k - x_\perp)\), is

\[ J_k = (1 - \gamma)^2 ||\Psi_N||_p^2 + (1 - \gamma)^2 \sum_{i=0}^{N-1} ||\Psi_i||_Q^2 + \beta||\gamma(\bar{r}^*_k - r_\perp) + (r_\perp - r)||_\infty^2 \]

Differentiate \(J_k\) with respect to the step size \(\gamma\) and evaluate this at \(\gamma = 1\) gives

\[ \frac{\partial J_k}{\partial \gamma} \bigg|_{\gamma = 1} = c^T(\bar{r}^*_k - r_\perp) \]

where \(c\) is a subgradient to \(||\cdot||_\infty\) with respect to \(\gamma\).

If this inner product is positive it means that the cost function decrease as \(\gamma\) decreases which in turn implies that the cost can be reduced by moving \(\bar{r}^*_k\) closer to \(r_\perp\).

From the definition of the subgradient it follows that

\[ c^T(\bar{r}^*_k - r_\perp) \geq ||\bar{r}^*_k - r||_\infty - ||r_\perp - r||_\infty \]

Since \(r_\perp\) by definition is the closest feasible point to \(r\) in the chosen norm, the right hand side is strictly greater than zero unless \(\bar{r}^*_k = r_\perp\). This means that the cost, \(J_k\), can be improved by making an arbitrarily small move of \(\bar{r}^*_k\) towards \(r_\perp\) and hence the only stationary point for \(\bar{r}^*_k\) is \(r_\perp\). \(\square\)

Note that it is not self-evident that this tracking MPC algorithm, using pseudo setpoints, is locally optimal in the sense that it minimizes the infinite horizon LQR cost. However in the recent work Ferramosca et al. (2011) the authors prove that under certain conditions a similar type of MPC algorithm has the local optimality property. The same argumentation can be used to argue that also this algorithm possesses the local optimality property.
5 Aircraft Example

In this example we consider a linearized form of the unstable short period dynamics for a fighter aircraft, discretized using a sample-time of 16ms. The equations of motion describing the relevant aircraft dynamics can be approximated as a two state discrete time system with the angle of attack, $\alpha$, as the first state and the pitch rate, $q$, as the second, see Stevens and Lewis (2003).

$$
\begin{bmatrix}
\alpha_{k+1} \\
q_{k+1}
\end{bmatrix} =
\begin{bmatrix}
0.9719 & 0.0155 \\
0.2097 & 0.9705
\end{bmatrix}
\begin{bmatrix}
\alpha_k \\
q_k
\end{bmatrix} +
\begin{bmatrix}
0.0071 \\
0.3263
\end{bmatrix}
\delta_k
$$

$$
y =
\begin{bmatrix}
1 & 0
\end{bmatrix}
\begin{bmatrix}
\alpha_k \\
q_k
\end{bmatrix}
$$

The state constraints for angle of attack and pitch rate has been set to

$$
x_k \in \mathcal{X} = \left\{ (\alpha_k, q_k)^T \mid -15 \leq \alpha_k \leq 30, -100 \leq q_k \leq 100 \right\}
$$

The control surface deflection has been limited to $-25 \leq \delta_k \leq 25$.

The weighting matrices in the cost function have been chosen as $Q = \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix}$, $R = 1$, $\beta = 10^4$ and $P$ is the Lyapunov cost of the corresponding LQ controller. The prediction horizon is chosen to $N = 20$ and $\epsilon$ is chosen small enough to not have any noticeable effect, $\epsilon = 10^{-5}$.

In this example we compare the properties of the proposed MPC algorithm with the method proposed in Limon et al. (2008), with the difference that the penalty function for the pseudo setpoint which in Limon et al. (2008) is a quadratic function has been replaced with a linear penalty function, which also has been proposed as an extension of the algorithm in Ferramosca et al. (2009). We will refer to this method as the reference method. Both controllers have been tuned using the same weighting matrices.

Comparing the complexity, i.e., the size of the QP, of the proposed controller with the reference method, which has 66 variables and 229 constraints, the proposed controller has 67 variables, however only 164 constraints. The significant difference in number of constraints come from the terminal constraints set which is defined with 72 inequalities, in $\mathbb{R}^3$, for the reference method compared to only 4 inequalities, in $\mathbb{R}^2$, for the proposed controller. This difference will result in lower computational effort for online solution of the QP or a reduced complexity of an explicit implementation.

![Time Response](image1.png)

Figure 1: The upper figure shows the pilot input reference, $r$ (red) and the output, $\alpha$, for the two controllers. The proposed controller as the solid line and the reference controller as the dashed line. The lower figure shows the control signal for the both controllers.

When calculating the explicit MPC solution, (Bemporad et al., 2002), the resulting controller is a continuous piecewise affine function of the current state, $x_k$. Each affine function, $u_k^i = K_i x_k + g_i$, is
valid in a polytopic subset, $X_i$, of the state space where the active constraints does not change. The online complexity of the explicit solution is mainly affected by the task of finding the partition, $X_i$, in which the current state is located and hence dependent on the number of partitions, $N_X$, that is required to describe the solution. The runtime complexity is in worst case $O(N_X)$ but with advanced search algorithms it can be reduced to $O(\log_2 N_X)$ (Kvasnica, 2009).

The explicit solution for the reference method, with a prediction horizon of $N = 10$, has $N_X = 6850$ partitions, compared to the explicit solution for the proposed method which has only $N_X = 860$ partitions, i.e., a reduction of the number of partitions with 87%.

The step response for both controllers are shown in Figure 1. The response in angle of attack is very similar between the both methods, only a slightly faster convergence can be observed for the proposed controller as the setpoint approaches the border of the feasible set.

Since the terminal constraint set is a scaled version of the nominal invariant set, i.e., the one calculated for $r = 0$, it is clearly not the maximal invariant set and hence, it can result in a smaller domain of attraction. This drawback becomes more evident for shorter prediction horizons, e.g., for $N = 5$ the proposed controller has a smaller domain of attraction than the reference method, see Figure 2, while for $N = 10$ there is no difference between the two controllers.

Therefore one must make a tradeoff between the complexity, i.e., the prediction horizon, and the needed domain of attraction.

This example has been implemented in MATLAB with the use of the latest versions of the toolboxes YALMIP (Löfberg, 2004) and MPT (Herceg et al., 2013).

6 Conclusions

A simple extension to the standard dual mode MPC algoritm to allow for tracking arbitrary setpoints has been presented.

It has been proven that by scaling the terminal state constraint set with a positive scalar $\lambda$ and requiring that the scaled and translated terminal set is a subset of the feasible set, both stability and recursive feasibility can be guaranteed for setpoints that approach the boundary of the feasible set.

The performance and complexity of the proposed controller has been illustrated with an example from the aircraft industry. This example illustrates the possibility that reduced complexity can be achieved compared to existing state of art methods without loosing performance of the controller.
References


