Predictive Metamorphic Control

He Kong, Graham Goodwin and Maria Seron

Abstract—Model Predictive Control (MPC) has become widely accepted in industry. The reason for its success are manifold including easy implementation, ability to handle constraints, capacity to deal with nonlinearities, etc. However, the method does have drawbacks including tuning difficulties. In this paper, we propose an embellishment to the basic MPC strategy by incorporating a tuning parameter such that one can move continuously from an existing controller to a new MPC strategy. The continuous change of this tuning parameter leads to a continuously varying stabilizing control law. Since the proposed strategy allows one to slowly move from an existing control law to a new and better one, we term the strategy Predictive Metamorphic Control. For the case of an infinite horizon problem without constraints and for the general case with state and input constraints, stability results are established. The merits of the proposed method are illustrated by examples.

I. INTRODUCTION

Since MPC originated in late 1970’s, it has found widespread application in many areas of systems and control, especially in industrial process control (see, for example, [1]-[8]). The method has many advantages, among which are its ability to deal with constraints and its capability for analysis and design for nonlinear systems (see, [9]-[12] and the references therein). Also, a comprehensive supporting theory is available and a standard procedure can be applied to guarantee optimality and stability (see, [9], [10],[12], [13]-[16] and the references therein). Moreover, there has been recent interest in the robustness issues of MPC, e.g., the impact of unmodeled dynamics and disturbances (see, for example, [17]-[21] and the references therein).

A common situation in applying MPC is that one has an existing controller which may somehow not be performing satisfactorily. In such cases, MPC can be seen as a desirable up-grade of the existing control law to improve performance, especially when there are constraints and nonlinearities. However, this raises the question of how to move smoothly from the existing controller to the new controller. It would be desirable if one could introduce a strategy rendering a new control law that, not only makes use of the existing controller, but also improves system performance in a smooth fashion.

The above idea is an evolution of ideas that have a long history in control engineering. For example, the idea of bumpless transfer control ([22]-[25]) can be viewed as a mechanism that allows one to switch backward and forward between multiple controllers (e.g., an existing controller and a new one) while not causing undesirable transient performance.

Similarly, in the recent MPC literature, there has been interest in the problem of tuning an MPC controller such that it behaves as an existing linear controller (see [26]-[28]). In [28], it is constructively, yet not computationally, proved that every continuous feedback law can be obtained by parametric convex programming. In [26] and [27], methods are provided to select the MPC weight matrices so that the resulting MPC controller exactly matches (or closely approximates) a given linear controller. Therefore, the resulting MPC controller inherits the small-signal properties of the linear control design, when the constraints are not active, and still optimally deals with constraints during transients.

Our proposed strategy here is related to the above ideas but different in major aspects. Instead of considering the controller matching problem as in [26]-[28], we introduce a confidence parameter $\lambda \in [0, 1]$ which reflects the designer’s willingness to switch from the existing control law to a new one. In practice, we envisage that designers would begin with $\lambda = 0$ and slowly increase $\lambda$ as they gain confidence in the performance. Since the proposed strategy allows one to smoothly move from an existing control law to a new and better one, we term the methodology Predictive Metamorphic Control (PMC). More importantly, when there are no constraints present in the system, the PMC strategy with $\lambda = 0$ exactly matches the existing controller. However, when constraints are present, the PMC strategy gives better control performance than the existing controller since feasibility is enforced.

The remainder of this paper is organized as follows. In Section II, some preliminaries are given. Section III contains the main results of this paper. First, our main idea about mixing the control strategy is given. Then stability result for the infinite horizon problem with no constraints is presented. Next, we consider a more general case with state and input constraints and establish exponential stability of the closed-loop system. Two examples are given in Section IV to illustrate the merits of the proposed scheme. Section V concludes the paper.

Notation: Notation used in this paper is standard. We use $A^T$ to denote the transpose of matrix $A$ and $\mathbb{R}^n$ to denote the $n$-dimensional Euclidean space. The notation $M > 0$

Since $\lambda$ is embedded in a cost function, the optimization of which gives the new controller, as it will be shown later, we borrow the term “Metamorphic” from geology. In geology, metamorphism is the solid-state recrystallization of pre-existing rocks due to changes in physical and chemical conditions, primarily heat, pressure, and the introduction of chemically active fluids.
\( (\geq 0) \) means that \( M \) is real symmetric and positive definite (semi-definite). \( I \) stands for identity matrices of appropriate dimensions. Matrices, if their dimensions are not explicitly stated, are assumed to be compatible for the associated algebraic operations.

II. PRELIMINARIES

For simplicity, in this paper we will use linear system models to explain the ideas. However, the proposed strategy can be extended to nonlinear cases. This is left for future work. We assume we have a discrete-time stabilizable plant model as follows:

\[
\begin{align*}
x_{k+1}^p &= A^p x_k^p + B^p u_k \\
y_k &= C^p x_k^p ,
\end{align*}
\]

(1)

where \( A^p \in \mathbb{R}^{n \times n}, B^p \in \mathbb{R}^{n \times r} \) and \( C^p \in \mathbb{R}^{t \times n} \) are constant system matrices, \( x_k^p, u_k \) and \( y_k^p \) are the state vector, input vector and output vector, respectively.

We also assume that we have an existing stabilizing controller described in the following state-space form:

\[
\begin{align*}
x_{k+1}^{c} &= A^c x_k^{c} - B^c y_k^c \\
u_k &= C^c x_k^{c} - D^c y_k^c \quad ,
\end{align*}
\]

(2)

where \( A^c \in \mathbb{R}^{m \times m}, B^c \in \mathbb{R}^{n \times t}, C^c \in \mathbb{R}^{r \times m} \) and \( D^c \in \mathbb{R}^{t \times t} \) are constant system matrices. In the absence of constraints, the closed-loop system satisfies

\[
\begin{bmatrix} x_{k+1}^p \\ x_{k+1}^{c} \end{bmatrix} = \begin{bmatrix} A^p - B^p D^c C^p & B^p C^c \\ -B^c D^c & A^c \end{bmatrix} \begin{bmatrix} x_k^p \\ x_k^{c} \end{bmatrix} .
\]

Moreover, the closed-loop system matrix

\[
\begin{bmatrix} A^p - B^p D^c C^p & B^p C^c \\ -B^c D^c & A^c \end{bmatrix}
\]

has its all eigenvalues inside the unit circle. We also assume that when we use MPC as the control strategy for system (1), our desired performance is captured by a cost function of the following form:

\[
V_k = (x_N^p)^T F (x_N^p) + \sum_{k=0}^{N-1} [(x_k^p)^T Q (x_k^p) + u_k^T R u_k]
\]

(4)

where

\[
\begin{align*}
F &\geq 0, Q \geq 0, R > 0 \\
x_0 &= x \\
u_k \in \mathbb{U}, &\text{ for } k = 0, \ldots, N - 1 \\
x_k \in \mathbb{X}, &\text{ for } k = 0, \ldots, N \\
x_N \in \mathbb{X}_f
\end{align*}
\]

(5)

The matrix \( F \) is typically selected to ensure closed-loop stability ([15]). The idea of MPC is to start with a fixed optimization horizon, \( N \) length, as in (4), using the current state of the plant as the initial state. We then optimize the objective function (4) over the fixed interval accounting for constraints to obtain an optimal sequence of \( N \) control moves. We then apply only the first move to the plant. Time advances one step and the same \( N \)-step optimization problem is repeated using the new state of the plant as the initial state. Thus one continuously revises the current control action based on the current state and accounting for the constraints over an optimization horizon of length \( N \). Since we will start with systems having no constraints, the constraint sets in (5) will initially be taken as \( U = \mathbb{R}^m, X_f = \mathbb{X} = \mathbb{R}^n \). More generally, the sets \( U, X \) and \( X_f \) are appropriate sets that capture the constraints that the system is required to satisfy (see Part C of Section III).

III. MAIN RESULTS

A. Mixing the Control Strategies

In this section, we describe the proposed PMC strategy that smoothly moves from the existing controller (2) to MPC. We introduce an embellishment to the existing controller as below:

\[
u_k = \overline{u}_k + \tilde{u}_k
\]

(6)

where \( \overline{u}_k \) is the input that is provided by the existing control law, i.e., \( \overline{u}_k = C^c x_k^{c} - D^c y_k^c \); \( \tilde{u}_k \) is the control increment to be designed. Denote

\[
X_k = \begin{bmatrix} x_k^p \\ x_k^{c} \end{bmatrix},
\]

then we have the following composite model

\[
X_{k+1} = AX_k + Bu_k
\]

(7)

where

\[
A = \begin{bmatrix} A^p - B^p D^c C^p & B^p C^c \\ -B^c D^c & A^c \end{bmatrix}, B = \begin{bmatrix} B^p \\ 0 \end{bmatrix}
\]

(8)

Next we design a cost function which combines (6) with the desired performance as captured by (4):

\[
\tilde{V}_k = \lambda X_N^T F_L X_N + \sum_{k=0}^{N-1} \{(1 - \lambda)u_k^T \overline{u}_k + \lambda[(x_k^p)^T Q (x_k^p) + u_k^T R u_k]\}
\]

\[
+ \lambda[(x_k^p)^T Q (x_k^p) + u_k^T R u_k]
\]

\[
+ (C^c x_k^{c} - D^c y_k^c)^T R (C^c x_k^{c} - D^c y_k^c + \tilde{u}_k)]
\]

(9)

where \( Q, R \) are as in (4), \( \lambda \in [0, 1] \). The final state weighting, \( F_L \), will later be designed to achieve closed-loop stability in the presence of constraints.

Remark 1: The parameter \( \lambda \in [0, 1] \) reflects the designers’ willingness to switch from the existing control law to the MPC law. The parameter \( \lambda \) reflects the designers’ confidence in the new control strategy. It can be clearly seen that when \( \lambda = 0 \) and \( \lambda = 1 \), the above cost function reduces to the existing controller \( \overline{u}_k \) and full MPC strategy, respectively. In the remainder of paper, we will only consider the case of \( \lambda \in (0, 1) \).

B. The Infinite Horizon Problem without Constraints

The PMC strategy is aimed at solving analysis and design problems for control systems with constraints, nonlinearities, etc. However, it is insightful to consider the case where there are no constraints. In the absence of constraints, we can exactly formulate the problem into a standard linear quadratic regulator (LQR) problem. Moreover, the stability
of the closed-loop system (7) can then be guaranteed by the standard LQR theory. When there are no constraints in the system and $N$ is infinite, we can write the cost function as
\[
\begin{align*}
\tilde{V}_k &= \sum_{k=0}^{\infty} \{(1-\lambda)\tilde{u}_k^T\tilde{u}_k + \lambda[(x_k^T)^TPQ(x_k^T) + u_k^T R u_k]\} \\
&= \sum_{k=0}^{\infty} \{(1-\lambda)\tilde{u}_k^T\tilde{u}_k + \lambda[(x_k^T)^TQ(x_k^T) + \\
&\quad (C^T x_k^T - D^T y_k^T + \tilde{u}_k)^T R(C^T x_k^T - D^T y_k^T + \tilde{u}_k)]\} \\
&= \sum_{k=0}^{\infty} [u_k^T R L \tilde{u}_k + X_k^T Q L X_k + 2X_k^T S L \tilde{u}_k]
\end{align*}
\] (10)

The algebraic Riccati equation ([29]) associated with the system (7) and cost function (10) is given by
\[
\begin{align*}
A^T P A - P - (A^T P B + S_L) R_L^{-1} (B^T P A + S_L^T) + Q &= 0 \\
\end{align*}
\] (12)

where
\[
\begin{align*}
R_L &= (1-\lambda)I + \lambda R \\
Q_L &= \lambda \begin{bmatrix} Q + (D^T C P)^T R D^T C P & - (D^T C P)^T R C^c \\
-D^T C P^T R C^c & (C^c)^T R C^c \\
\end{bmatrix} \\
S_L^T &= \lambda \begin{bmatrix} -R D^T C P & R C^c \\
\end{bmatrix}
\end{align*}
\] (11)

The optimal control for the system (7) resulting from the optimization of the cost function (10) is given by
\[
\tilde{u}_k^* = -R_L^{-1} (B^T P A + S_L^T) X_k.
\] (13)

After a cross-term with weight $S_L$ is present in the above infinite horizon cost function, we can utilize a standard procedure to eliminate this term ([30]) as described below. First, since $R > 0$, when $\lambda \in (0, 1)$, we know that $R_L > 0$. Set
\[
\tilde{Q} = Q_L - S_L R_L^{-1} S_L^T
\] (14)

then it follows that:
\[
\tilde{Q} = \begin{bmatrix} Q + (D^T C P)^T R_1 (D^T C P) & - (D^T C P)^T R_2 C^c \\
(C^c)^T (R_2)^T R D^T C P & (C^c)^T (R_1)^T C^c \\
\end{bmatrix},
\]

where
\[
\begin{align*}
R_1 &= R - \lambda R R_L^{-1} R, \\
R_2 &= \lambda R R_L^{-1} R - I.
\end{align*}
\]

Set
\[
v_k = \tilde{u}_k + R_L^{-1} S_L^T X_k,
\] (15)

then the system (7) and cost (10) can be reformulated as
\[
X_{k+1} = AX_k + B v_k
\] (16)
\[
\tilde{V}_k = \sum_{k=0}^{\infty} v_k^T R L v_k + X_k^T \tilde{Q} X_k
\] (17)

respectively, where
\[
\tilde{A} = A - BR_L^{-1} S_L^T.
\]

Given $R_L > 0$, clearly, the nonnegativeness of $\tilde{V}_k$ guarantees that $\tilde{Q} \geq 0$. The algebraic Riccati equation associated with the system (16) and cost function (17) is ([30]):
\[
\tilde{A}^T P \tilde{A} - P - \tilde{A}^T P B R_L^{-1} B^T P \tilde{A} + \tilde{Q} = 0.
\] (18)

The optimal control for system (16) resulting from the optimization of cost function (17) is given by
\[
v_k^* = -R_L^{-1} B^T P \tilde{A} X_k.
\] (19)

Given that $v_k$ and $\tilde{u}_k$ are related by (15), the trajectories of the two systems (7) and (16) are the same if they start from the same initial state. Furthermore, the values taken by the two performance indices, (10) and (17), which are associated with systems (7) and (16), respectively, are also the same. Besides, the two optimal control laws are related by
\[
\tilde{u}_k^* = \tilde{u}_k + R_L^{-1} S_L^T X_k
\] (20)

We are then in a position to state the first result of this paper.

**Theorem 1:** Consider the system (16) and assume $M$ is a symmetric square root of $Q$, i.e.,
\[
M M^T = \tilde{Q}.
\]

Then the matrix pairs $(\tilde{A}, B)$ and $(\tilde{A}, M)$ are stabilizable and detectable, respectively. Moreover, the nonnegative solution, $P$, to (18), is unique and the state feedback given by (19) stabilizes the system (16).

**Proof:** Since $M M^T = \tilde{Q}$, from standard LQR theory, we know that in order to prove the result, we need only establish the stabilizability of $(\tilde{A}, B)$ and detectability of $(\tilde{A}, M)$ (see [29] and [30]). We first show the stabilizability of $(\tilde{A}, B)$. Since the existing controller yields a stable closed-loop system (see Section II), i.e., $A$ has its all eigenvalues inside the unit circle, if we simply set
\[
K_1 = R_L^{-1} S_L^T,
\]
then it follows that
\[
\tilde{A} + B K_1 = A.
\]

Thus $\tilde{A} + B K_1$ has its all eigenvalues inside the unit circle, i.e., there exist a feedback gain $K_1$ such that $\tilde{A} + B K_1$ is stable. Hence, $(\tilde{A}, B)$ is a stabilizable pair.

Next we establish the detectability of $(\tilde{A}, M)$. Note that this is equivalent to showing that $(\tilde{A}^T, M^T)$ is stabilizable. In order to do this, we will find a feedback gain $K_2$ such that $\tilde{A}^T + M^T K_2$ is stable. Given $M^T$ is nonnegative definite (see [30]), we can define $(M^T)^\dagger$ as its pseudo inverse. We choose
\[
K_2 = (M^T)^\dagger S_L R_L^{-1} B^T,
\]
then it follows that
\[
\begin{align*}
\tilde{A}^T + M^T K_2 \\
&= (A - BR_L^{-1} S_L^T)^T + M^T K_2 \\
&= A^T - S_L R_L^{-1} B^T + M^T (M^T)^\dagger S_L R_L^{-1} B^T \\
&= A^T.
\end{align*}
\]
where $A^T$ is Schur stable by assumption. This completes the proof. 

C. The General Case with State and Input Constraints

In this case, following a similar procedure to that of the previous section, we can formulate the optimization problem as:

$$E_{N}^{OPT} : \bar{V}_k = \lambda X_N^T F_L X_N + \sum_{k=0}^{N-1} [v_k^T R_L v_k + X_k^T Q X_k]$$

(21)

where $R_L, Q, v_k$ are the same as those in (11), (14) and (15), respectively, and

$$\begin{cases}
X_0 = X \\
v_k - \bar{K} X_k \in U \subset \mathbb{R}^m, \text{ for } k = 0, \ldots, N - 1 \\
X_k \in \mathbb{R} \times \mathbb{R}^m, \text{ for } k = 0, \ldots, N \\
X_N \in F_L \subset \mathbb{R}^{n+m}
\end{cases}$$

(22)

where

$$\bar{K} = R_L^{-1} S_L^T - \left[ -D^c C \quad C^c \right]$$

and $U, X$ are the original sets given in (5). In the remainder of the paper, we assume that the sets $U, X$ and $F_L$ are convex and that $U$ and $X$ contain the origin of their respective spaces. Generally speaking, when there are both state and input constraints, in order to maintain closed-loop stability, $F_L$ is often chosen such that $X^T F_L X$ is the value function of the infinite horizon, unconstrained optimal control problem for the same system (see [31] and [9]). This problem, solved in the above subsection, is a standard LQR problem whose value function is $X_0^T P X_0$, where $P$ is the nonnegative solution to (18). The terminal state weighting used in this case is then

$$F_L = P.$$ 

The local controller is then chosen to be the optimal linear controller

$$K_f(X) = -\bar{K} X$$

where

$$\bar{K} = R_L^{-1} B^T P \bar{A}$$

and $R_L^{-1}$ is as in (13). The terminal set $X_f$ is usually taken to be the maximal output admissible set $\Theta$ (9) for the closed-loop system using the local controller $K_f(X)$, i.e.,

$$\Theta \triangleq \Theta_1 \cap \Theta_2,$$

where, for $i = 0, 1, \ldots,$

$$\Theta_1 = \{X : (K + \hat{K})(\bar{A} - B(K + \hat{K}))^T X \in U\},$$

$$\Theta_2 = \{X : (\bar{A} - B(K + \hat{K})^T X \in X\}.$$ 

The set $\Theta$ is the maximal positively invariant set for the system $X_{k+1} = (\bar{A} - B(K + \hat{K})) X_k$, for which the required input and state constraints are not violated. Denote the set $W$ of feasible initial states for problem $E_{N}^{OPT}$. With the above choice for the terminal triple $(X_f, K_f, F_L)$, we then have the following result.

Theorem 2: Consider the closed-loop system (16), controlled by the PMC strategy (21)-(22), then the origin is exponentially stable in $W$.

Proof: With the assumptions made above and choice for the terminal triple $(X_f, K_f, F_L)$, the proof is immediate using the results in Chapters 4 and 5 of [9].

Remark 2: The requirement $v_k - \bar{K} X_k \in U \subset \mathbb{R}^m$, for $k = 0, \ldots, N - 1$, is to keep the mixed control strategy feasible, i.e., $u_k$ in (6) still lies in $U$. Thus by implementing the PMC strategy, one can get a satisfying control strategy without revising the original control constraints. This point will be illustrated through examples later.

IV. ILLUSTRATIVE EXAMPLES

In this section, we use two examples to illustrate the proposed strategy.

A. The Scalar Case without Constraints

First, consider the following scalar system:

$$x_{k+1}^p = 0.99 x_k^p + u_k.$$ 

The existing controller is assumed to be

$$u_k = -0.09 x_k^p.$$ 

We follow steps (16)-(19) and add an increment, $\tilde{u}_k$, to the given control input, i.e., we define

$$u_k = -0.09 x_k^p + \tilde{u}_k.$$ 

For simplicity, we set $Q = R = 1$. Then for the infinite horizon case, the cost function (10) becomes

$$\bar{V}_k = \sum_{k=0}^{\infty} \tilde{u}_k^2 + 1.0081 \lambda (x_k^p)^2 - 0.18 \lambda \tilde{u}_k x_k^p.$$ 

We move from $\lambda = 0$ to $\lambda = 1$ with each step increasing by 0.02. The initial state is chosen be $x_0^p = 1$. At each step, we solve the problem of minimizing the above cost function and the trajectories of the state of the plant are plotted in Figure 1.

Fig. 1. The trajectory of the state of the plant under PMC

It can be seen from Figure 1 that when $\lambda = 0$ the PMC strategy matches the existing controller. As $\lambda$ increases the performance improves. Thus one can choose a certain $\lambda$ according to the confidence in the new controller.
B. Second Order Plant with Constraints

The second example uses the following second order plant
\[ G(z) = \frac{0.4z + 1}{(z + 1)(z - 0.95)}. \]

Clearly the plant is marginally stable and non-minimum phase. We next design a PI controller of the following structure
\[ D(z) = \frac{z - 0.2}{z - 1} \]
to stabilize the plant. Following the procedure in sections II and III, we obtain the following state space model for the closed-loop system:
\[
A = \begin{bmatrix}
-0.45 & -0.05 & 0.8 \\
1 & 0 & 0 \\
-0.4 & -1 & 1 \\
\end{bmatrix},
B = \begin{bmatrix}
1 \\
0 \\
0 \\
\end{bmatrix}.
\]

We take the initial state as \( X_0 = \begin{bmatrix} 10 & 4 & 4 \end{bmatrix}^T \) and set
\[ Q = \begin{bmatrix}
10 & 0 & 0 \\
0 & 0 & 5 \\
\end{bmatrix},
R = 0.1. \]

(I) When there are no input or state constraints, we first compare the output trajectories of the system under the PI controller and PMC strategy. At each step, the PMC strategy is given by applying the first element of optimizing sequence for the 10-step cost function (21), i.e., the optimisation horizon is chosen to be length 10. The comparison of trajectories is given in Figure 2. The blue dotted line with squares and the family of red dotted lines are output trajectories of the system under the PI controller and the PMC strategy with different values for \( \lambda \), respectively. We move from \( \lambda = 0 \) to \( \lambda = 0.9 \) with each step increasing by 0.1. Finally we take \( \lambda \to 1 \). From Figure 2, we can see that when \( \lambda = 0 \), the PMC strategy matches the unconstrained PI controller. Moreover, increasing \( \lambda \) in the PMC strategy leads to a faster transient response.

(II) Next we assume the same initial state and same optimisation horizon, but add a constraint on the input, i.e., in (4), \( u_k \in [-0.4, 0.4] \), for \( k = 0, \ldots, N - 1 \). That is to say, in the optimization problem (21), \( v_k - \hat{K}X_k \in [-0.4, 0.4] \), for \( k = 0, \ldots, N - 1 \). Again, at each step, the PMC strategy is obtained by applying the first element of the optimizing sequence for a 10-step cost function of the form (21). We compare three controllers, namely, (i) the PI controller followed by a saturation, (ii) the same PI controller in anti-windup form, (iii) the PMC controller with \( \lambda = 0 \), (iv) the PMC controller with \( \lambda \in (0, 1) \). The results are shown in Figure 3 and 4. It is well known that simply saturating the output of a PI controller gives poor performance. This is illustrated by the blue dotted line in Figure 3. The output trajectory is oscillatory. Indeed, the closed-loop system achieved by simply saturating the output of the PI controller can be unstable when we change constraints to be stricter. In the presence of constraints, a common choice, in practice, is to implement the PI controller in anti-windup form (see, for example, [22], [23], [32]-[34]). And this renders good performance, as shown by the black dotted line in Figure 3. Setting \( \lambda = 0 \) in the PMC strategy does not correspond to either of the two PI controller forms since feasibility is enforced in the PMC formulation. Actually, as can be seen from the red dotted line in Figure 3, the performance of the PMC strategy with \( \lambda = 0 \) lies between the performance of the two PI controller formulations.
When we increase $\lambda$, the PMC strategy progressively gives better performance. When $\lambda \to 1$, the PMC strategy outperforms the anti-windup PI controller. This is shown in Figure 4. The red dashed lines are the output trajectory of the plant with the PMC strategy for different $\lambda$.

V. CONCLUSION

This paper has presented a novel strategy, called PMC, which allows one to move continuously from an existing controller to a new MPC strategy. This is achieved by incorporating a tuning parameter that gives the designer freedom to adjust between the existing control law and a new one. Stability of the closed-loop system has been established for the infinite horizon problem with no constraints and for the general case with state and input constraints. The advantages of the proposed method have been illustrated through two examples. Extensions of the method to more complex cases are the subject of current and future work.

REFERENCES