Explicit nonlinear predictive control for a magnetic levitation system

A. Ulbig, S. Olaru, D. Dumur and P. Boucher

ABSTRACT

The paper presents a methodology for the construction of an explicit nonlinear control law via approximation of the nonlinear constrained finite-time optimal control (CFTOC). This is achieved through an approximate mapping of a general nonlinear system in a set of linear piecewise affine (PWA) systems. The key advantages of this methodology are two-fold: firstly, the construction of an analytic solution of the CFTOC problem leads to an efficient explicit implementation. Secondly, by taking advantage of MPC’s systematic fashion to handle constraints, an improved performance can be obtained for the closed-loop system. The proposed theory is applied in real-time for a system with fast dynamics: a magnetic levitation benchmark.

Key Words: Predictive control, piecewise affine control law, magnetic levitation.

I. INTRODUCTION

Model predictive control (MPC) is a widely accepted control methodology for linear systems. Its extension to the nonlinear case (NMPC) attracts attention in industry and academics [2], [23] with several challenging issues. MPC traditionally involves the solution of a constrained finite-horizon optimal control problem at each sampling step [22]. Due to the ability to handle state and input constraints from the design stage, it represented a very active research area; with the practical success [11] comforting the theory.

However, MPC has in its traditional form a drawback originating from its time-consuming on-line solution of the optimisation problem. The restrictiveness of MPC’s computational burden manifests itself in practice through the use of control hardware with limited computational performance (often associated with exponential economical costs) for small sampling times. Unfortunately, small sampling times are necessary for the control of processes with fast dynamics and thus MPC’s applicability is restricted to processes with relatively slow dynamics. This limitation has motivated the search for new resolution methods for the optimisation problem that take advantage of pre-computing an optimal control law over all feasible states. In other words the system’s state vector is considered as the parameter in the optimisation. Hence, the computational burden is shifted off-line. Such methods have been presented for linear systems with linear and quadratic cost functions [3], [4], [5], [6]. They have further been extended by Borrelli et al. to linear hybrid systems in piecewise linear affine (PWA) form for linear and quadratic objectives respectively [7].

Johansen and Grancharova [13] proposed a construction technique for the case of nonlinear multiparametric optimisation. This allows to obtain an approximate explicit solution by partitioning the state-space into polyhedral regions and calculating a piecewise linear control function over these regions. The degree of approximation can be adapted by augmenting the density of the partition in the sensitive regions. Several other attempts of constructing approximate solutions have been reported recently for the linear case in [15, 16] but their construction exploits the structure of the optimization problem and cannot be extended directly to the nonlinear case. In [14], the linear MPC in presence of nonlinear constraints is analysed and approximate explicit solutions are
proposed upon a geometrical approach. However, the generalization of the presented methods to the case of nonlinear prediction models leads to computationally involving numerical procedures. Fotiou et al. presented a cylindrical algebraic decomposition (CAD) approach for nonlinear systems [9], leading to an exact explicit solution for an approximated problem. Thus the overall degree of approximation is given roughly by the precision of the initial decomposition.

The new methodology presented here uses an approximate map of an originally nonlinear system transformed into a piecewise linear affine (PWA) system that maintains the information of the nonlinear structure of the original system. Piecewise affine (PWA) systems have attracted much interest, being a powerful tool for approximating non-linear systems as already postulated by Sontag in 1981 [24] and their equivalence to other classes of hybrid systems as proven lately [17]. Bemporad et al. have shown that for constrained linear systems and further for PWA systems, a quadratic feedback controller may be obtained by applying multi-parametric programming techniques [4]. In a first stage, the solution found for a general constrained finite-time optimal control (CFTOC) problem is represented as a set of piecewise linear feedback control laws. By using multivariate interpolation for this set of feedback control laws, an explicit nonlinear control law in the form of a polynomial function can straightforwardly be constructed. Finally, this explicit nonlinear control law is applied to the given original nonlinear system, thus smoothing the dynamics and improving the online evaluation time. A look-up table and its tedious positioning mechanism is not needed any more.

The main contribution of the paper is the algorithmic construction of an explicit polynomial state feedback function approximating the constrained-finite-time control law. The main advantages of the construction is the continuity of the state feedback and the adaptation of the approximation with respect to the chosen partition of the state-space.

The paper is structured as follows: In Section II a formulation of the general nonlinear system and the corresponding general CFTOC problem is given. The PWA mapping algorithm for nonlinear systems is presented in Section III. The formulation of the explicit piecewise affine (PWA) control law with respect to the CFTOC problem is developed in Section IV. Its interpolation into a truly nonlinear control law is described in Section V. An experimental implementation for a magnetic levitation system is reported in Section VI. The conclusion is given in Section VII.

II. Constrained Finite-Time Optimal Control Problem

The general discrete-time nonlinear system can be described in state-space form as follows

\[ x(k+1) = f(x(k), u(k)) \quad (1) \]
\[ y(k) = h(x(k), u(k)) \]

subject to the state and input constraints

\[ u(k) \in U, x(k) \in X, \quad k = 0, ..., N, \quad (2) \]

where \( x \in \mathbb{R}^n \) is the discrete state vector with \( x(0) = [x_1(0), ..., x_n(0)]^T = x_0, \ u \in \mathbb{R}^m \) the discrete control input vector, \( y \) the system output, \( N \) the prediction horizon and \( f: \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n \) a nonlinear, continuously differentiable function in terms of \( x \) and \( u \). The constraints on the system are defined by convex sets \( X \subset \mathbb{R}^n, U \subset \mathbb{R}^m \).

Considering the task of regulating system (1) with respect to the imposed constraints (2) towards the origin and furthermore assuming that the origin is a stable equilibrium point, a cost function that penalizes the deviation of the state and the control input from zero is introduced

\[ J(u, x_0) = L_N(x(N), u(N)) + \sum_{k=0}^{N-1} L_k(x(k), u(k)), \quad (3) \]

where \( L_k \in \mathbb{R}^{n \times m} \) represents the so-called stage cost and \( L_N \in \mathbb{R}^{n \times m} \) the terminal state cost (both represented by linear combinations of \( \| . \|_2 \) terms on the tracking error and control energy). The optimisation vector is given by \( u^* := [u(0)^T, ..., u(N-1)^T]^T \in \mathbb{R}^{mN} \), which consists of all decision variables, e.g. control inputs, for \( k = 0, ..., N-1 \).

Obtaining the optimal control moves over the given prediction horizon and its application according to the receding horizon principle, is then equivalent to solving at each sampling instant \( k \) the constrained finite-time optimal control (CFTOC) problem

\[ u^*(x_0) = \arg \min_u J(u, x_0) \]

s.t. \[ x(k + 1) = f(x(k), u(k)) \quad (4) \]
\[ u(k) \in U, x(k) \in X, \quad k = 0, ..., N. \]
\[ X(N) \in X_f. \]

Here, \( X_f \) is a positive invariant set for the nonlinear system (1) in closed-loop with a stabilizing control law, which is satisfying the constraints and for which the terminal cost function is \( L_N(x(N), u(N)) \). These are classical ingredients in nonlinear model predictive control [22] that guarantee closed-loop stability upon a dual-mode design principle.
III. PWA Transformation Method

The task is to derive a generalisable approximate transformation from the general nonlinear system (1) into the form of a linear time-varying model
\[
\begin{align*}
x(k + 1) &= A(x(k))x(k) + B(x(k))u(k) \\
y(k) &= C(x(k))x(k) + D(x(k))u(k)
\end{align*}
\]
(5)
and further approximating it by a set of piecewise affine (PWA) systems [24], which are described as
\[
x(k + 1) = A_i x(k) + B_i u(k) + \alpha_i \\
y(k) = C_i x(k) + D_i u(k) + \beta_i \quad \text{for} \quad \begin{bmatrix} x \\ u \end{bmatrix} \in \Omega_i
\]
(6)
where \(\Omega_i\) are convex polyhedra that are defined by a finite number of linear inequalities in the control input and state-space. Note, that the convex polyhedra are a sub set of the complete state-space \(\Omega\), with \(\Omega_i \subset \Omega\).

PWA systems represent the simplest extension of linear systems that are able to model non-linear and non-smooth processes with arbitrary accuracy. Linearisation techniques using PWA systems (5) do approximately maintain — depending on the chosen accuracy — the information of the global nonlinear dynamics of the given nonlinear system. Standard Jacobi linearisation approximates nonlinear dynamics only locally around a given working point.

3.1. PWA Linearisation

The linearisation technique proposed here has the following principle: the state space is effectively partitioned into several regions, each representing a confined subspace of the entire state-space. There exists several viable options of how to partition the state-space such as uniform partitioning, logarithmic partitioning and in a very general sense case-tailored partitioning. For reasons of simplicity a partitioning that can be refined individually for each axis of the state-space has been chosen, as seen in Figure 1.*

For each region \(\Omega_i \subset \mathbb{R}^n\), a local Jacobi linearisation around the respective center \((x_i, u_i)\) of the confined

---

*Generating a mesh of points on the state-space is a common issue for several control problems, e.g. linearization, discretization and so on. A grid is the most intuitive construction, however the steady state behavior might be affected by errors near the equilibrium point. A possible alternative is to use a spherical distribution [10], which provides an equivalent grid distribution in polar coordinates. The advantage of changing from cartesian to polar coordinates is the concentration of partitions near the equilibrium point (or other sensitive regions with respect to the system’s nonlinearity).

---

Fig. 1. a) Case-sensitive partitioning of the state-space (9 × 2 partitions); b) A spherical mesh.

The extended (state+control) subspace is calculated
\[
\begin{align*}
A_i &= \frac{\partial f}{\partial x}(x_i, u_i) \\
B_i &= \frac{\partial f}{\partial u}(x_i, u_i) \\
C_i &= \frac{\partial h}{\partial x}(x_i, u_i) \\
D_i &= \frac{\partial h}{\partial u}(x_i, u_i)
\end{align*}
\]
(7)
The affine terms are found based on:
\[
\begin{align*}
\alpha_i &= f(x_i, u_i^*(x_i)) - A_i x_i - B_i u_i \\
\beta_i &= g(x_i, u_i^*(x_i)) - C_i x_i - D_i u_i
\end{align*}
\]
(8)
In this construction the choice of the control action \(u_i\) which generates the linearization point for the region \(i\) is a design parameter which influences the complexity of the partition and the quality of the approximation\(^1\). Using the optimal control input \(u^*\) for the CFTOC problem (5) leads to
\[
\begin{align*}
\alpha_i &= f(x_i, u_i^*(x_i)) - A_i x_i - B_i u_i^*(x_i) \\
\beta_i &= g(x_i, u_i^*(x_i)) - C_i x_i - D_i u_i^*(x_i)
\end{align*}
\]
(9)
The PWA model over a state-space partition will be:
\[
\begin{align*}
x(k + 1) &= A_i x(k) + B_i u_i^*(x_i) + \delta u(k) + \alpha_i \\
y(k) &= C_i x(k) + D_i u_i^*(x_i) + \delta u(k) + \beta_i
\end{align*}
\]
(10)
for \(x(k) \in \Omega_i\) with \(u_i^*(x_i)\) the input which maintains the system state at \(x_i\) and finally
\[
\begin{align*}
x(k + 1) &= A_i x(k) + B_i v(k) + \alpha_i \\
y(k) &= C_i x(k) + D_i v(k) + \beta_i
\end{align*}
\]
(11)
for \(x(k) \in \Omega_i\) with
\[
\alpha_i = \alpha_i + B_i u_i^*(x_i), \quad \beta_i = \beta_i + D_i u_i^*(x_i).
\]
\(^1\)The linearisation procedure was treated extensively in the literature related to control problems. The geometrical analysis performed in [26] can offer some answers. However, several questions arise for structural aspects of the Jacobi linearisation, for example regarding the controllability along the trajectories [1].
3.2. PWA Balancing

This section discusses the specific case of single input systems, where the control deviation \( v(k) \) is the effective control decision over a given region of the partition. Using this hypothesis for the construction of the system in the PWA form (5), the term \( \tilde{\alpha}_i \) (and \( \tilde{\beta}_i \) correspondingly) can be evaluated as a model parameter in order to reduce the discrepancies between the local linear approximations.

The calculation of each \( \tilde{\alpha}_i \) depends on the topology of the partitioning, e.g. the values of \( \tilde{\alpha}_i \) for all the neighboring regions. An exploration of the state-space, starting at the region that contains the equilibrium point towards those regions for which the regulation of the system shall be geared to, has to be done in order to correctly create the relationships with neighboring regions. In such a way, a necessary balancing of the set of PWA systems as a whole is accomplished.

For balancing the set of PWA systems, a data tree has been implemented to construct a data base of the relational structure for the regions of the state-space, as seen in Algorithm III.1 and depicted in Figures 2. Note that two regions are considered to be adjacent if they share a common frontier of dimension \( n - 1 \), where \( n \) is the dimension of the state-space.

Algorithm III.1: PWA EXPLORATION (\( \Omega \))

Start exploration of the set of PWA systems at the origin \( x_0 \) and create an index of adjacent regions.

\[
\text{Start exploration of the set of PWA systems at the origin} \ x_0 \ \text{and create an index of adjacent regions.}
\]

\[
\text{Move pointer to region } \Omega_k \ : \ x_0 \in \Omega_k
\]

\[
\text{pointer } \leftarrow x_0 = (x_0^1, \ldots, x_n^0)^T \in \Omega_k
\]

\[
\text{Include this region at the top of the search index.}
\]

\[
\text{SearchTree}(\text{level}) \leftarrow \Omega_k
\]

while (no. indexed regions < no. existing regions)

\[
\text{Explore all adjacent neighbours of the regions in } \text{SearchTree}(\text{level}).
\]

\[
\text{for } i \leftarrow 1 \text{ to } n
\]

\[
\text{do }
\]

\[
\text{Move pointer with } \Delta_i \text{ to adjacent region}
\]

\[
\text{\Omega_i in the } n\text{-th order space.}
\]

\[
x_i = (x_0^1, \ldots, x_n^0 + \Delta_i, \ldots, x_n^0)^T \ | \ x_i \in \Omega_i
\]

\[
\text{pointer } \leftarrow x_i
\]

\[
\text{Add region } \Omega_i \text{ into } \text{SearchTree}(\text{level} + 1).
\]

\[
\text{SearchTree}(\text{level} + 1) \leftarrow \Omega_i
\]

\[
\text{Go to next level of } \text{SearchTree}. \ 
\]

\[
\text{level } \leftarrow \text{level} + 1
\]

\[
\text{return (SearchTree)}
\]

Two cases have to be distinguished for the calculation of the \( \tilde{\alpha}_i \): Either there exists only one adjacent region where the value of \( \tilde{\alpha}_i \) is already known (Figure 3), or there exist already several adjacent regions for which the values of \( \tilde{\alpha}_{i-1}, \ldots, \tilde{\alpha}_{i-k} \) have already been calculated (also shown Figure 3).

This corresponds to the case of an explicit calculation of \( \tilde{\alpha}_i \) as a function of the value of the single already calculated \( \tilde{\alpha}_{i-1} \), Equation 12.a, and the case of a quadratic approximation, for which the values of the already calculated \( \tilde{\alpha}_i \)'s of all the neighboring regions have to be considered, Equation 12.b.
By implementing the above exploration and balancing techniques, a set of PWA systems is created that maintains — depending on the quality of the chosen partitioning — the nonlinear behaviour of the original system (1).

Algorithm
III.2: PWA balancing ($\Omega$, $ST$)

Balancing the set of PWA systems in the correct order using the index of search-tree $ST$: starting with $\Omega_k : x_0 \in \Omega_k$.

\[
\begin{align*}
\triangleright & \text{ Start balancing with } \Omega_k \in ST(0). \\
& \text{ level } = 0 \\
& \Omega_k \leftarrow ST(\text{level}) \\
\triangleright & \text{ Set } f_{\Omega_k} \text{ equal to zero.} \\
& f_{\Omega_k} \leftarrow 0 \\
\text{ while } & (\text{no. indexed regions} < \text{no. existing regions}) \\
& \text{ do } \left\{ \\
& \quad \triangleright \text{ Go to next level of } ST. \\
& \quad \text{ level } \leftarrow \text{level} + 1 \\
& \quad \text{ n } \leftarrow \text{length}(ST(\text{level})) \\
& \quad \text{ for } i \leftarrow 1 \text{ to n } \\
& \quad \text{ do } \left\{ \\
& \quad \quad \triangleright \text{ Get } i\text{-th region out of set } ST(\text{level}). \\
& \quad \quad \Omega_i \leftarrow ST(\text{level}) \\
& \quad \quad \triangleright \text{ Calculate points } x_{\text{sect}_i} \text{ on border of } \Omega_i \\
& \quad \quad \triangleright \text{ and its adjacent region(s) } \Omega_{i-1}, \ldots, \Omega_{i-k} \\
& \quad \quad \triangleright \text{ from } ST(\text{level} - 1). \\
& \quad \quad \text{ if } (\not\exists x_{\text{sect}}) \\
& \quad \quad \text{ then } Eq. (12.a) \leftarrow x_{\text{sect}} \\
& \quad \quad \text{ else } Eq. (12.b) \leftarrow x_{\text{sect}_{i-1}} \ldots x_{\text{sect}_{i-k}} \\
& \text{ return } (\Omega_{\text{balanced}})
\end{align*}
\]

IV. Obtaining an Explicit PWA Control Law

Given the balanced and sufficiently accurate modeled set of PWA systems (6), techniques exist that allow for an off-line calculation of a respective set of linear explicit piecewise control laws in the form of

\[
u_i(x) = k_i x + c_i \quad \text{ with } x \in P_i \subset \Omega_i,
\]

where $x \in \mathbb{R}^m$ represents the system state, $u_i \in \mathbb{R}^m$ is the valid control input for the region $P_i$ of the partition $\Omega_i$ of the sub-space. The parameters $k_i$ and $c_i$ are the respective linear and constant terms of the explicit linear feedback law [19]. This calculation is done, such that the criteria given by a general CFTOC problem (4) including state and input constraints are globally satisfied.

A general formulation for such an algorithm has been presented by Bemporad et al. [4], who have derived an algorithm that expresses the solution $u^*(x)$ and the minimum value $V(x) = J(u^*(x))$ of a nonlinear problem (1) as a set of explicit functions of the system state $x \in \mathbb{R}^n$.

The parameters that play an important role with respect to the complexity of the controller solution and hence the corresponding off-line computational cost have been identified as follows. They are listed together with their observed impact on computational cost.

Partitioning Accuracy: Has a linear effect, with respect to each axis of the state-space. If a higher grid density ($\Delta x_1, \ldots, \Delta x_n$) is chosen, PWA complexity and hence the computational cost will increase linearly.

Prediction Horizon ($N$): Has an exponential influence on the complexity of the piecewise affine control law and hence on the real-time computational cost for the control action.

Choice of Constraints: They are important with respect to feasibility concerns and might lead to numerous extra iterations in the optimisation calculation.

V. Obtaining an explicit Nonlinear Control Law

In order to derive an explicit and truly nonlinear control law in the form of a polynomial function in terms of the system state, a multi-dimensional approximation of the control information contained in the set of linear explicit PWA control laws is necessary. Sophisticated multivariate polynomial interpolation techniques do exist [12] with whom this task can be accomplished. Given the following general
interpolation framework for a set of \((r + 1)\) data points
\[
\Phi(x_i, k_0, \ldots, k_r) = z_i \ \forall (x_i, z_i) \ i = 0, \ldots, r, \quad (14)
\]
where \(\Phi\) is the polynomial function that fits best with respect to the data points according to a given criteria. A nonlinear control law in the form of a sufficiently complex \(s\)-th order polynomial
\[
u(x) = k_p x_i^s + k_p - 1 x_i^{s-1} x_j^1 + k_p - 2 x_i^{s-2} x_j^2 + \ldots + k_0, \quad (15)
\]
where \(k_0, \ldots, k_p \in \mathbb{R}\) are the coefficients of this polynomial, can be constructed.

The quality of the approximation depends on the density of the chosen PWA partitioning, which is case-dependent, and the chosen order \(s\) of the interpolating polynomial. A sufficiently high degree \(s\) can be selected upon an iterative procedure in terms of the quality of the regression model parameters \(R^2\) (the coefficient of determination) and \(R^{MSE}\) (the root mean squared error).

Hence, a procedure has been developed, which allows to approximately map the hard-to-tackle general nonlinear control problem (1) into a set of PWA systems (6). A set of linear explicit control laws can be calculated and interpolated into a nonlinear control law of polynomial form (15).

The polynomial interpolation will bring a faster evaluation of the control action and hence allow for a more widespread application of MPC algorithms to very fast responding systems.

**Stability**

The closed-loop stability can be assessed using two arguments (we point the reader to the references [22], [20] and [21] for a study of the approximation implications in NMPC stability):

- Stable predictive control design for the original nonlinear model (4);
- The approximated explicit control law satisfies the input constraints, boundedness of the approximation error and the capability of making such a bound arbitrarily small by increasing the quality of the PWA approximation.

The recent work [20] and [21] showed that a MPC law computed off-line from a finite number of exact control moves can be used to derive guaranteed approximation error, closed-loop stability and performances. The key properties of the approximated control law have to be: the continuity and the decrease of the approximation bound with the increase of the number of exact control moves (and the density of the grid). The method presented in the previous sections offer the continuity guarantees and in addition offers a physical insight on the dynamics in the approximated prediction model (thus being an alternative to model-free polynomial curve fitting, interpolation or neural networks type of approximation). The main criticism is the complexity of the explicit control design for PWA systems, which restricts the application to systems with small state-space representation and prediction horizon.

**VI. Magnetic levitation control**

A task of practical relevance in the realm of controlling dynamically fast responding systems is the magnetic levitation system [25]. The purpose is to control the vertical displacement of the pendulum, under the action of the magnetic and gravitational forces, leading the pendulum to an equilibrium point.

The system dynamics are given by
\[
m \ddot{z} = -mg + ce \frac{i^2}{(z_0 - z)^2}, \quad (16)
\]
where \(z\) corresponds to the vertical disc’s position, \(m\) is the mass of the disc, \(g\) is the gravitational constant, \(i\) the electric current that determines the strength of the magnetic field and \(z_0\) the disc’s initial position at \(t = 0\). The magnetic suspension system has, due to the magnetic coil, a strongly nonlinear behavior. Its dynamics are unstable and fast, e.g. sampling rates of \(T_s = 10\ m_s\) are needed for a sufficient real-time control. Constraints exist on the control input, the magnetic coil current is limited to \(i_{max} \leq 1\ A\), and on the system states, the disc’s position is bounded by the metal bearings, \(|z| \leq 6.5\ mm\) and the disc’s velocity shall be limited to \(|\dot{z}| \leq 0.05\ \frac{m}{s}\). Frictional effects can be neglected for the modelling of the magnetic suspension, since the controlling force is transmitted through the magnetic field, i.e. there are no surface contacts. Furthermore, air friction is considered negligible when compared to the magnitude of the magnetic field force.

The magnetic suspension’s dynamics (16), can be expressed by the two-dimensional state-space system
\[
\begin{bmatrix}
x_1(t) \\
x_2(t)
\end{bmatrix} = \begin{bmatrix}
x_2 \\
-g + \frac{u^2}{m(x_1(0) - x_1)^2}
\end{bmatrix}, \quad (17)
\]
where the variables of the system state \(x_i\) correspond to \(x_1 = z\), \(x_2 = \dot{z}\) and the control input to \(u = \frac{i}{K_p}\) for which \(K_p\) is the actuator gain. The equilibrium point
\( z_0 \) is set to the disk’s mid-air elevation height, being \( z_0 = x_1(0) = 7.6 \text{ mm} \).

The specific control task is to maintain the disc at equilibrium point at a mid-position, \( x_1(0) \), and realize rejection and tracking objectives around this point as optimal as possible with respect to the chosen criteria, which are expressed as weights on the input and system states and through an optional system constraint. The CFTOC problem formulation for the magnetic suspension system uses a prediction horizon of \( N = 3 \) and weighting matrices
\[
Q = \begin{bmatrix} 30 & 0 \\ 0 & 1 \end{bmatrix}, \quad Q_N = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}, \quad R = \frac{1}{10}
\]
and an optional state constraint on the velocity, \( x_2 \), of the suspended disc. classical well-tuned PID controller.

In order to derive a set of PWA control laws and eventually a nonlinear control law, the nonlinear suspension model (17), is first linearly mapped into a set of 25 PWA systems for the sub space of interest: \( x_1 = [-4.5 \text{ mm} \ldots + 4.5 \text{ mm}], x_2 = [-0.5 \frac{\text{m}}{s} \ldots + 0.5 \frac{\text{m}}{s}] \).

In a second step, a set of altogether 770 PWA control laws, \( u_{PW A}^*(x) \), is calculated for the given CFTOC problem. These PWA control laws, in the form of \( u_i^* = k_i x + c_i \), are stored in a multi-dimensional look-up table. Locating the currently needed control law \( u_i^* \) in a timely manner, might at times become unfeasible for the process control in real-time. In a final step, the PWA control law set \( u_{PW A}^*(x) \) is interpolated into a nonlinear control law, \( u^*(x) \), using \( (r + 1) = 24000 \) data points. This explicit nonlinear control law of 15th order is the real-time implementable solution for the CFTOC problem in the form
\[
\begin{align*}
u^*(x) := & -5.81427 \cdot 10^{37} x_1^{15} + 6.47081 \cdot 10^{35} x_1^{14} x_2 + 2.84809 \cdot 10^{33} x_1^{13} x_2^2 + 1.12819 \cdot 10^{34} x_1^{13} x_2^2 + 1.0516 x_2^2 - 7.4599 x_2 - 5.343 \cdot 10^{-5}.
\end{align*}
\]

The peak on-line evaluation time needed for a control law \( u^*(x) \) for a given sample \( (t_k, x_k) \) is approximately 0.06 ms. This is 170 times smaller than the time frame of the sampling rate, in which the control law \( u^*(x) \) needs to be implemented. The evaluation over the state-space \( \Omega = \{ x_1 \times x_2 \} \) of the set of 770 PWA control laws, \( u_{PW A}^*(x) \), is shown in Figure 4.a. Each law, \( u_i^* \), is only valid in a very small to medium-sized PWA model region \( \Omega_i \). Note the discontinuities between some neighboring PWA control laws \( u_i^* \), which obviously will also cause unwanted discontinuities in the closed-loop response, e.g. degrade the control performance of the magnetic levitation system. The nonlinear polynomial control law, \( u_{NL}^*(x) \), can be studied in Figure 4.b. The control dynamics of \( u_{NL}^*(x) \) are well-interpolated with respect to the obvious “gaps” seen in Figure 4.a. No discontinuities exist and instead of a time-consuming search for the correct control law in a look-up table, the polynomial \( u_{NL}^*(x) \) is efficiently evaluated.

![Figure 4](image)

Fig. 4. a) PWA control (770 regions), b) Nonlinear control.

The state-space evolutions, as well as the individual evolutions over time for the system states, \( x_1 \), and the control input, \( u^*(x) \), are presented in Figures 5-6 for the CFTOC problem formulation with and without the state constraints. As can be seen, the closed-loop response for the state-constrained setup is well within the given lower bound of the disc’s velocity, i.e. \( x_2 \geq -0.05 \frac{\text{m}}{s} \).

VII. Conclusion

A new approach for calculating an explicit solution to the CFTOC problem of a given general nonlinear dynamical system has been presented. This task is achieved by means of an intermediary approximate mapping of the nonlinear system into a set of PWA
This hands-on practical approach allows to use advanced optimal control schemes such as MPC in connection with input and state constraints for the calculation of an optimal nonlinear control law. Classical (N)MPC schemes normally imply a heavy computational burden on-line. With the proposed approach, an almost complete shift of this burden to an off-line phase is achieved. Only an easy-to-implement nonlinear control law, given in the form of an explicit polynomial function, is needed for the actual control task.

The frame of future work on this approach is defined by a recursive form of the explicit control law construction algorithm. In this case, the resulting map of piece-wise control laws of one iteration would be analysed and used to refine the PWA partitioning for the next iteration in regions where the gradient of the piece-wise control laws has been large. This iterative process would be repeated until an a priori chosen quality is attained for the control law.

REFERENCES


© 2008 John Wiley and Sons Asia Pte Ltd and Chinese Automatic Control Society

Prepared using asjcauth.cls


