Proceedings of the ASME 2015 Dynamic Systems and Control Conference
DSCC2015
October 28-30, 2015, Columbus, Ohio, USA

DSCC2015-9859

TRACKING CONTROL OF NON-MINIMUM PHASE SYSTEMS USING FILTERED BASIS FUNCTIONS: A NURBS-BASED APPROACH

Molong Duan
Department of Mechanical Engineering
University of Michigan, Ann Arbor, MI, 48109
molong@umich.edu

Keval S. Ramani
Department of Mechanical Engineering
University of Michigan, Ann Arbor, MI, 48109
ksramani@umich.edu

Chinedum E. Okwudire
Department of Mechanical Engineering
University of Michigan, Ann Arbor, MI, 48109
okwudire@umich.edu

ABSTRACT

This paper proposes an approach for minimizing tracking errors in systems with non-minimum phase (NMP) zeros by using filtered basis functions. The output of the tracking controller is represented as a linear combination of basis functions having unknown coefficients. The basis functions are forward filtered using the dynamics of the NMP system and their coefficients selected to minimize the errors in tracking a given trajectory. The control designer is free to choose any suitable set of basis functions but, in this paper, a set of basis functions derived from the widely-used non uniform rational B-spline (NURBS) curve is employed. Analyses and illustrative examples are presented to demonstrate the effectiveness of the proposed approach in comparison to popular approximate model inversion methods like zero phase error tracking control.

1. INTRODUCTION

The goal of tracking control is to force the output of a system to follow a desired trajectory as closely as possible. In linear time invariant (LTI) systems, perfect tracking control (PTC) can be achieved, in theory, by model inversion based control techniques [1]. PTC results in zero gain and phase errors between the desired and actual trajectories, if the model of the system is accurate and has a stable inverse. However, when applied to systems with non-minimum phase (NMP) zeros, whose inversions yield unstable poles, PTC gives rise to highly oscillatory or unstable trajectories which are unacceptable [1].

NMP zeros are prevalent in practice. For example, they occur in digital systems with fast sampling rates [2], as well as in systems with non-collocated placement of sensors and actuators [3]. Consequently, a lot of research effort has been poured into developing methods for accurately tracking NMP systems. One school of thought is to perform direct inversion with bounded reference trajectories [4–8] by modifying initial conditions [7], employing noncausal plant inputs [4–6] or computing output trajectories that cancel the effect of unstable zeros [8].

An alternative approach is to use approximate model inversion by ignoring the NMP zeros (aka NPZ-ignore) [9] or by employing zero phase error tracking control (ZPETC) [1,10–13] or zero magnitude error tracking control (ZMETC) [14,15]. Approximate model inversion techniques are more popular than their direct inversion counterparts because they are easier to understand and implement, and are applicable to a wider range of problems [9]. The NPZ-ignore technique disregards the NMP zeros and makes adjustments such that the DC gain of the controlled system is unity; but it results in magnitude and phase errors between the desired and actual trajectories that may be very detrimental to tracking performance [9]. ZPETC inverts the system to produce no errors in phase across all frequencies; however, it gives rise to errors in the gain of the controlled system which may adversely affect tracking performance. Conversely, ZMETC achieves approximate model inversion by making the net gain of the controlled system to be unity across all frequencies, at the expense of phase errors which amount to time delays and tracking errors.

This paper proposes a new approach for realizing tracking control for NMP systems by employing filtered basis functions (FBF). The desired trajectory to be tracked is assumed to be entirely known and the output of the tracking controller is represented as a linear combination of basis functions having unknown coefficients. The basis functions are forward filtered using the modeled dynamics of the NMP system and their coefficients selected to minimize the errors in tracking the

desired trajectory. The choice of a suitable set of basis functions is up to the control designer but a set of basis functions derived from the widely-used non-uniform rational B-spline (NURBS) curve is employed in this paper for elucidating the proposed technique. Analyses and illustrative simulations are used to demonstrate the superior tracking performance of the proposed FBF approach in comparison with NPZ-ignore, ZPETC and ZMETC. The rest of the paper is organized as follows. Section 2 gives a brief overview of approximate model inversion techniques for tracking control of NMP systems and contrasts them with the proposed FBF technique. Section 3 then introduces NURBS curves and demonstrates how they can be used to realize the proposed technique. Simulation-based examples are presented and discussed in Section 4, followed by remarks on the versatility and practicality of the FBF technique, ending with conclusions and future work.

2. TRACKING CONTROL FOR NMP SYSTEMS

Consider the discrete-time LTI SISO system $G(z^{-1})$ shown in Fig. 1, augmented with a tracking controller $C(z^{-1})$. G could represent the transfer function of a plant or a closed-loop controlled system [16]. Given a desired trajectory, $x_d(k)$, where $0 \le k \le E$, $k \in \mathbb{Z}$, the objective of tracking control is to design C such that its output $c(k)$, after passing through G, results in an actual trajectory $x(k)$ that is sufficiently close to $x_d(k)$. Ideally, $C = G^{-1}$ should be selected such that the overall dynamics $L = CG$, from $x_d(k)$ to $x(k)$, is unity, representing perfect tracking performance.

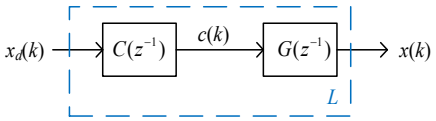

FIGURE 1: TRACKING CONTROL

However, if G contains NMP and/or poorly damped zeros (both of which are hereinafter referred to as un-cancellable zeros), its inverse is either unstable or gives rise to an overly oscillatory response, hence unacceptable. Under this condition, G can be written as

$$G(z^{-1}) = \frac{z^{-d} B_s(z^{-1}) B_u(z^{-1})}{A(z^{-1})}, \qquad (1)$$

where z^{-d} represents the d-step time delay of the system, and A, B_s and B_u are polynomials in z^{-1} with constant coefficients. B_s contains the stable and sufficiently damped zeros of G while B_u contains its un-cancellable zeros which prevent $C = G^{-1}$ from being realized.

2.1. TRACKING CONTROL USING APPROXIMATE MODEL INVERSION

A popular way of realizing C for NMP systems is through approximate model inversion. Approximate inversion techniques work by setting $C = \hat{G}^{-1}$, where $\hat{G}^{-1}$ represents a stable approximation of G^{-1} that is selected such that L is sufficiently close to unity (at least over a frequency range of interest). Table 1 provides a synopsis of $\hat{G}^{-1}$ and L for the three main approximate model inversion techniques, namely, NPZ-ignore, ZPETC and ZMETC. Notice that NPZ-ignore realizes $\hat{G}^{-1}$ by discounting B_u and normalizing the DC gain of L to unity, at the expense of gain and phase errors at higher frequencies. ZPETC invokes the fact that

$$\angle(B_u(z^{-1})B_u(z)) = 0 \qquad (2)$$

to achieve zero phase in L, but $|L|$ deviates from unity at frequencies beyond its DC value. On the other hand, ZMETC uses the fact that

$$\left|B_u(z^{-1})\right| = \left|B_u(z)\right| \qquad (3)$$

to create a stable approximate inverse of G that maintains unit magnitude across all frequencies in L, at the expense of non-zero phase behavior. Butterworth et al. [9] showed that the characteristics and severity of the magnitude and/or phase errors in NPZ-ignore, ZPETC and ZMETC are highly dependent on the location of the zeros of B_u. The implication being that the performance of any one of the methods relative to the others may vary quite significantly from system to system.

TABLE 1: SYNOPSIS OF APPROXIMATE INVERSION METHODS

Method	$C(z^{-1}) = \hat{G}^{-1}(z^{-1})$	$L(z^{-1})$
NPZ-ignore	$\frac{z^d A(z^{-1})}{B_s(z^{-1})B_u(1)}$	$\frac{B_u(z^{-1})}{B_u(1)}$
ZPETC	$\frac{z^d A(z^{-1})B_u(z)}{B_s(z^{-1})\left(B_u(1)\right)^2}$	$\frac{B_u(z^{-1})B_u(z)}{\left(B_u(1)\right)^2}$
ZMETC	$\frac{z^d A(z^{-1})}{B_s(z^{-1})B_u(z)}$	$\frac{B_u(z^{-1})}{B_u(z)}$

2.2. TRACKING CONTROL USING FILTERED BASIS FUNCTIONS (FBF)

The approximate model inversion techniques discussed in the previous section generate the exact representation of $c(k)$ using an approximate representation of G (or, more specifically, of G^{-1}). Here, we introduce a different approach for tracking control that generates an approximate representation of $c(k)$ using the exact representation of G. The method is built on two assumptions:

i) The desired trajectory, $x_d(k)$, is known a priori
ii) The output of the tracking controller, $c(k)$, is parameterized as

$$c(k) = \sum_{i=0}^{n} p_i u_i(k), \tag{4}$$

where $\{u_i(k)\}$ is a set of n+1 linearly independent basis functions and $\{p_i\}$ is a set of coefficients, $p_i \in \mathbb{R}$. Note that both assumptions stated above are satisfied by a wide range of tracking applications like those in manufacturing, robotics and aerospace, where desired trajectories are typically known ahead of time and are often parameterized using a finite set of basis curves [17,18]. Following from Eq.(4), $x(k)$ can be written as

$$x(k) = Gc(k) = \sum_{i=0}^{n} p_i \tilde{u}_i(k), \tag{5}$$

where

$$\tilde{u}_i(k) = Gu_i(k) \tag{6}$$

represents the basis functions, forward filtered by G. Let us define W and $\tilde{W}$, the input and output vector spaces, respectively, as

$$W = \text{span}\{u_i(k)\} \tag{7}$$

$$\tilde{W} = \text{span}\{\tilde{u}_i(k)\}. \tag{8}$$

The vector spaces W and $\tilde{W}$ are subspaces of $\mathbb{R}^{E+1}$ because any trajectory defined over E+1 discrete points belongs to $\mathbb{R}^{E+1}$. It is important to note that $\{u_i(k)\}$ should be chosen such that $\{\tilde{u}_i(k)\}$ is linearly independent (meaning that $\{\tilde{u}_i(k)\}$ should be a basis for $\tilde{W}$). Section 2.3 discusses conditions for the linear independence of $\{\tilde{u}_i(k)\}$.

For perfect tracking, $\{p_i\}$ should satisfy the equation

$$\underbrace{\begin{bmatrix} \tilde{u}_0(0) & \tilde{u}_1(0) & \dots & \tilde{u}_n(0) \\ \tilde{u}_0(1) & \tilde{u}_1(1) & \dots & \tilde{u}_n(1) \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{u}_0(E) & \tilde{u}_1(E) & \dots & \tilde{u}_n(E) \end{bmatrix}}_{\tilde{\mathbf{U}}} \underbrace{\begin{bmatrix} p_0 \\ p_1 \\ \vdots \\ p_n \end{bmatrix}}_{\mathbf{p}} = \underbrace{\begin{bmatrix} x_d(0) \\ x_d(1) \\ \vdots \\ x_d(E) \end{bmatrix}}_{\mathbf{x}_d}. \tag{9}$$

An exact solution to Eq.(9) exists, if and only if the rank of $\tilde{\mathbf{U}}$ is equal to the rank of $[\tilde{\mathbf{U}} | \mathbf{x}_d]$, i.e., $\mathbf{x}_d \in \tilde{W}$. However, generally $n < E$ and $\mathbf{x}_d \notin \tilde{W}$, therefore the solution to Eq.(9) is approximated by least squares according to the equation [19]

$$\mathbf{p} = \left(\tilde{\mathbf{U}}^{\mathrm{T}}\tilde{\mathbf{U}}\right)^{-1} \tilde{\mathbf{U}}^{\mathrm{T}} \mathbf{x}_d. \tag{10}$$

Consequently, the relationship between the vectors $\mathbf{x} = [x(0), x(1), \dots x(E)]^{\mathrm{T}}$ and $\mathbf{x}_d$ can be expressed as

$$\mathbf{x} = \underbrace{\tilde{\mathbf{U}}\left(\tilde{\mathbf{U}}^{\mathrm{T}}\tilde{\mathbf{U}}\right)^{-1} \tilde{\mathbf{U}}^{\mathrm{T}}}_{\mathbf{H}} \mathbf{x}_d. \tag{11}$$

The matrix $\mathbf{H}$ is a projection matrix from the desired trajectory to the actual trajectory [20]. It is a square matrix of order E+1, symmetric and idempotent (i.e., $\mathbf{H}^2 = \mathbf{H}$), with rank and trace both equal to n+1, i.e., the number of basis functions. In a perfect tracking scenario, $\mathbf{H}$ is the identity matrix such that $\mathbf{x} = \mathbf{x}_d$. However, because of the least squares solution, each row k of $\mathbf{H}$ represents a FIR filter which acts on $\mathbf{x}_d$ to generate the corresponding $x(k)$. Note that $\mathbf{H}$ is not a Toeplitz matrix (i.e., it is not a diagonal-constant matrix); therefore, in general, $\mathbf{H}$ behaves like a time varying FIR filter which acts on $\mathbf{x}_d$ to generate $\mathbf{x}$.

Notice that $\mathbf{H}$ is functionally equivalent to the transfer function L described in Section 2.1; i.e, it maps $\mathbf{x}_d$ to $\mathbf{x}$. From Table 1, one observes that for NPZ-ignore and ZPETC methods, L is a time invariant FIR filter, whereas for ZMETC, L is a time invariant IIR filter. In each case, the filter coefficients depend solely on $B_u(z^{-1})$; however, according to Eq.(11), the elements of $\mathbf{H}$ depend on the system dynamics G as well as the selected basis functions.

2.3. CONDITIONS FOR LINEAR INDEPENDENCE OF FILTERED BASIS FUNCTIONS

A unique solution to Eqs. (10) and (11) exists if $\left(\tilde{\mathbf{U}}^{\mathrm{T}}\tilde{\mathbf{U}}\right)^{-1}$ exists; i.e., if $\tilde{\mathbf{U}}^{\mathrm{T}}\tilde{\mathbf{U}}$ is invertible. The matrix $\tilde{\mathbf{U}}^{\mathrm{T}}\tilde{\mathbf{U}}$ is invertible provided $\tilde{\mathbf{U}}$ has linearly independent columns meaning that rank of $\tilde{\mathbf{U}}$ is n+1 [21].

The linear dependence of $\{\tilde{u}_i(k)\}$ implies the existence of non-zero $\{\gamma_i\}$ such that

$$x(k) = \sum_{i=0}^{n} \gamma_i \tilde{u}_i(k) = 0 \tag{12}$$

$$c(k) = \sum_{i=0}^{n} \gamma_i u_i(k) \neq 0, \tag{13}$$

given that $\{u_i(k)\}$ are linearly independent by definition (because they are basis functions). Hence $\{\tilde{u}_i(k)\}$ are linearly dependent if there exists a non-zero $c(k)$ such that

$$x(k) = Gc(k) = 0. \tag{14}$$

The implication of Eq.(14) is that $c(k)$ belongs to the null space of G.

The FBF approach is not limited to the system definition given by Eq.(1). Hence a more generalized (state-space) representation of the system G which accounts for non-zero initial conditions is used for the analysis. It is given by

$$\begin{aligned}\mathbf{f}(k+1) &= \mathbf{A}_d\mathbf{f}(k) + \mathbf{B}_d c(k),\\ x(k) &= \mathbf{C}_d\mathbf{f}(k) + D_d c(k),\end{aligned} \tag{15}$$

where $\mathbf{f}(k)$ is the state vector while $\mathbf{A}_d$, $\mathbf{B}_d$, $\mathbf{C}_d$ and D_d are the state, input, output and feedforward matrices, respectively. Note that the system transfer function can be recovered from Eq.(15) as

$$G(z^{-1}) = \mathbf{C}_d(\mathbf{I} - \mathbf{A}_d z^{-1})^{-1}\mathbf{B}_d z^{-1} + D_d, \tag{16}$$

where $\mathbf{I}$ is identity matrix. The system output $x(k)$ is composed of a zero input response (ZIR) and zero state response (ZSR) [19], i.e.,

$$x(k) = \underbrace{\mathbf{C}_d\mathbf{A}_d^k\mathbf{f}(0)}_{\text{Zero Input Response (ZIR)}} + \underbrace{\mathbf{C}_d\sum_{j=0}^{k-1}\mathbf{A}_d^{k-j-1}\mathbf{B}_d c(j) + D_d c(k)}_{\text{Zero State Response (ZSR)}}. \tag{17}$$

Judging from Eq. (17) one obvious (albeit trivial) solution to Eq.(14) is $\mathbf{f}(0) = \mathbf{0}$ and $c(k) = 0$, which contradicts Eq.(13). Non-trivial solutions can be obtained by solving the equations given by ($0 \le k \le E$)

$$0 = \mathbf{C}_d\mathbf{A}_d^k\mathbf{f}(0) + \mathbf{C}_d\sum_{j=0}^{k-1}\mathbf{A}_d^{k-j-1}\mathbf{B}_d c(j) + D_d c(k). \tag{18}$$

Let $c(k) = \hat{c}(k)$ and $\mathbf{f}(0) = \hat{\mathbf{f}}(0)$ be the solution of Eq.(18) and $\mathbf{f}_0(0)$, $\mathbf{f}_1(0)$, ..., $\mathbf{f}_n(0)$ be the initial states used to forward filter $u_0(k)$, $u_1(k)$, ..., $u_n(k)$, respectively, using Eq.(15). Then for linear dependence

$$\begin{aligned}\mathbf{U}\boldsymbol{\gamma} &= \hat{\mathbf{c}}\\ \mathbf{F}\boldsymbol{\gamma} &= \hat{\mathbf{f}}(0)\end{aligned} \tag{19}$$

where

$$\begin{aligned}\mathbf{U} &= \begin{bmatrix} u_0(0) & u_1(0) & \dots & u_n(0)\\ u_0(1) & u_1(1) & \dots & u_n(1)\\ \vdots & \vdots & \ddots & \vdots\\ u_0(E) & u_1(E) & \dots & u_n(E)\end{bmatrix},\\ \mathbf{F} &= \begin{bmatrix}\mathbf{f}_0(0) & \mathbf{f}_1(0) & \dots & \mathbf{f}_n(0)\end{bmatrix},\\ \boldsymbol{\gamma} &= \begin{bmatrix}\gamma_0 & \gamma_1 & \dots & \gamma_n\end{bmatrix}^{\mathrm{T}},\\ \hat{\mathbf{c}} &= \begin{bmatrix}\hat{c}(0) & \hat{c}(1) & \dots & \hat{c}(E)\end{bmatrix}^{\mathrm{T}}.\end{aligned} \tag{20}$$

Therefore, the filtered basis functions $\{\tilde{u}_i(k)\}$ are linearly dependent if and only if the basis functions $\{u_i(k)\}$ and filter initial states $\{\mathbf{f}_i(0)\}$ satisfy the condition

$$\operatorname{rank}\left(\begin{bmatrix}\mathbf{U}\\ \mathbf{F}\end{bmatrix}\right) = \operatorname{rank}\left(\begin{bmatrix}\mathbf{U} & \hat{\mathbf{c}}\\ \mathbf{F} & \hat{\mathbf{f}}(0)\end{bmatrix}\right). \tag{21}$$

Note that $\hat{c}(k)$ and $\hat{\mathbf{f}}(0)$ are properties of the system whereas $\mathbf{U}$ and $\mathbf{F}$ are defined by the user. Using a simple example, it is shown in Section 4 that the condition of Eq.(21) is very hard to satisfy in practice; therefore linear dependence of filtered basis functions is highly improbable. Nonetheless, should Eq.(21) be satisfied by a given $\mathbf{U}$ and $\mathbf{F}$, either $\mathbf{U}$ or $\mathbf{F}$ can be easily modified to establish linear independence of the filtered basis functions.

3. NURBS-BASED IMPLEMENTATION

3.1. NON UNIFORM RATIONAL B-SPLINE (NURBS)

The non-uniform rational B-spline (or NURBS for short) is widely used for parameterizing curves in applications which involve trajectory tracking like CNC manufacturing and robotics [17,18,22,23]. It is popular because of its intuitiveness and flexibility combined with its excellent mathematical and algorithmic properties [17,18]. Therefore, it is adopted in this paper for demonstrating the proposed technique.

A NURBS curve of degree m, defined by n+1 control points (or coefficients) p_0, p_1,…, p_n, with corresponding weights w_0, w_1,…, w_n and a knot vector $[g_0\ g_1\ \dots\ g_{m+n+1}]^{\mathrm{T}}$ is expressed as [17]

$$c(\xi) = \frac{\sum_{i=0}^{n} N_{i,m}(\xi) w_i p_i}{\sum_{i=0}^{n} N_{i,m}(\xi) w_i}, \tag{22}$$

where $\xi \in [0,1]$ is the spline parameter and $N_{i,m}(\xi)$ is the basis function of degree m given by

$$\begin{aligned}N_{i,m}(\xi) &= \frac{\xi - g_i}{g_{i+m} - g_i} N_{i,m-1}(\xi) + \frac{g_{i+m+1} - \xi}{g_{i+m+1} - g_{i+1}} N_{i+1,m-1}(\xi)\\ N_{i,0}(\xi) &= \begin{cases}1 & g_i \le \xi < g_{i+1}\\ 0 & \text{otherwise}\end{cases}\end{aligned} \tag{23}$$

The basis functions are piecewise polynomial functions forming a basis for the vector space of all piecewise polynomial functions of the desired degree and continuity (for a given knot vector) [17]. Selecting a uniform knot vector of the form [17]

$$g_j = \begin{cases}0 & 0 \le j \le m-1\\ \dfrac{j-m}{n-m+1} & m \le j \le n+1\\ 1 & n+2 \le j \le m+n+1\end{cases}, \tag{24}$$

Eq.(22) can be re-written as

$$c(\xi) = \sum_{i=0}^{n} R_{i,m}(\xi) p_i \,, \tag{25}$$

where $\{R_{i,m}(\xi)\}$ are rational basis functions given by

$$R_{i,m}(\xi) = \frac{N_{i,m}(\xi) w_i}{\sum_{j=0}^{n} N_{j,m}(\xi) w_j} \,. \tag{26}$$

Notice the equivalence between Eqs. (4) and (25). If $\xi \in [0,1]$ is discretized into E+1 uniformly spaced points such that $\xi = [\xi_0\ \xi_1\ \ldots\ \xi_E]$ represents normalized time, $u_i(k) = R_{i,m}(\xi_k)$ can be defined and used to implement the proposed technique as described in Section 2.2.

3.2. HANDLING OF NON-ZERO INITIAL CONDITIONS

In tracking control, the trajectory to be tracked could have non-zero initial conditions associated with the trajectory itself, as well as with its higher-order derivatives. The NURBS framework described in the preceding section can be used to enforce such constraints by noting that $c'(\xi)$, the derivative of Eq.(25) w.r.t. ξ, can be written as [24]

$$c'(\xi) = \sum_{i=0}^{n} R'_{i,m}(\xi) p_i \,, \tag{27}$$

where

$$R'_{i,m}(\xi) = \frac{w_i N'_{i,m}(\xi)}{\sum_{j=0}^{n} N_{j,m}(\xi) w_j} - \frac{w_i N_{i,m}(\xi) \sum_{i=0}^{n} N'_{j,m}(\xi) w_j}{\left(\sum_{j=0}^{n} N_{j,m}(\xi) w_j \right)^2} \,. \tag{28}$$

Evaluating Eqs. (27) and (28) at $\xi = \xi_0$ gives

$$\dot{c}(\xi_0) = \frac{c'(\xi_0)}{T} = m \frac{w_1}{w_0} (p_1 - p_0) \,, \tag{29}$$

where $\dot{c}(\xi)$ is derivative w.r.t. time and T is the time duration of the trajectory. Higher order derivatives of Eq.(25) can be obtained in a similar manner. Note that the value of the first derivative at $\xi = \xi_0$ depends on the first two control points. The value of second derivative will depend on the first three control points and so on. Consequently, an initial condition on the r^{th} derivative of a given trajectory constraints the first r+1 control points associated with the trajectory (where $r < m$). Hence, the initial conditions of $x_d(k)$ (which are the same as those of $c(k)$) can be expressed as

$$\mathbf{Qp} = \mathbf{z} \,, \tag{30}$$

where $\mathbf{z}$ is a vector of initial conditions and $\mathbf{Q}$ is a matrix comprising the coefficients obtained from Eqs. (27)-(29), and similar equations derived for higher order derivatives. Accordingly, Eqs. (9) and (30) can be combined to obtain [25]

$$\begin{bmatrix} \tilde{\mathbf{R}} & \mathbf{Q}^{\mathrm{T}} \\ \mathbf{Q} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{p} \\ \boldsymbol{\lambda} \end{Bmatrix} = \begin{Bmatrix} \mathbf{x}_d \\ \mathbf{z} \end{Bmatrix} , \tag{31}$$

where $\boldsymbol{\lambda}$ is a vector of Lagrange multipliers and $\tilde{\mathbf{R}}$ is the same as $\tilde{\mathbf{U}}$ defined in Eq.(9), with $\{\tilde{u}_i(k)\}$ replaced by $\tilde{R}_{i,m}(\xi_k)$. The solution of Eq.(31) (in a manner similar to Eq.(10)) results in minimum tracking error while satisfying the conditions given by Eq.(30).

3.3. ENHANCED TRACKING OF HIGHER ORDER DERIVATIVES

In certain tracking applications, e.g., position tracking in robots and CNC machines, it may be of interest to also maintain close tracking of higher order derivatives of the desired trajectory, e.g., velocity and acceleration. The NURBS framework is very useful in this regard. For simplicity, we assume that the weights $\{w_i\}$ are equal to 1, meaning that $u_i(k) = R_{i,m}(\xi_k) = N_{i,m}(\xi_k)$. In this case, $c'(\xi)$ is a NURBS curve with unity weights defined by n control points, degree $m-1$ and knot vector $[g_1\ g_2 \ldots\ g_{m+n}]^{\mathrm{T}}$ (obtained by dropping first and last knots from the knot vector of $c(\xi)$). The control points of $c'(\xi)$, denoted as $\{q_j\}$, are related to the control points of $c(\xi)$ by [17]

$$q_j = m \frac{p_{j+1} - p_j}{g_{j+m+1} - g_{j+1}} \,. \tag{32}$$

Defining $\dot{\mathbf{c}} = \begin{bmatrix} \dot{c}(\xi_0) & \dot{c}(\xi_1) & \ldots & \dot{c}(\xi_E) \end{bmatrix}^{\mathrm{T}}$ as the derivative of $\mathbf{c}$ w.r.t. time, we get

$$\dot{\mathbf{c}} = \frac{m}{T} \mathbf{N}_d \mathbf{D} \mathbf{S} \mathbf{p} = \mathbf{N}_1 \mathbf{p} \,. \tag{33}$$

The matrix $\mathbf{N}_d$ is obtained by discretizing $N_{i,m-1}(\xi)$, computed over knot vector $[g_1\ g_2 \ldots\ g_{m+n}]^{\mathrm{T}}$ using $\xi = [\xi_0\ \xi_1\ \ldots\ \xi_E]$, while $\mathbf{D}$ and $\mathbf{S}$ are defined as

$$\mathbf{S} = \begin{bmatrix} -1 & 1 & 0 & \ldots & 0 \\ 0 & -1 & 1 & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & -1 & 1 \end{bmatrix}_{n \times (n+1)} ,$$
$$\mathbf{D} = \left[\operatorname{diag}\left(\begin{bmatrix} g_{m+1} - g_1 & g_{m+2} - g_2 & \cdots & g_{m+n} - g_n \end{bmatrix} \right) \right]^{-1} . \tag{34}$$

Following the same procedure, the r^{th} time derivative of $\mathbf{c}$ can be expressed in the form

$$\frac{d^r \mathbf{c}}{dt^r} = \mathbf{N}_r \mathbf{p} \,. \tag{35}$$

Filtering $\mathbf{N}_1$, $\mathbf{N}_2 \ldots \mathbf{N}_r$ using G gives, $\tilde{\mathbf{N}}_1, \tilde{\mathbf{N}}_2 \ldots \tilde{\mathbf{N}}_r$ respectively, consequently, accurate tracking of $\mathbf{x}_d$ and its first r derivatives can be achieved by solving the composite equation given by

$$\underbrace{\begin{bmatrix} \tilde{\mathbf{N}} \\ \alpha_1 \tilde{\mathbf{N}}_1 \\ \vdots \\ \alpha_r \tilde{\mathbf{N}}_r \end{bmatrix}}_{\bar{\mathbf{N}}} \mathbf{p} = \underbrace{\begin{Bmatrix} \mathbf{x}_d \\ \alpha_1 \dot{\mathbf{x}}_d \\ \vdots \\ \alpha_r \dfrac{d^r \mathbf{x}_d}{dt^r} \end{Bmatrix}}_{\bar{\mathbf{x}}}, \tag{36}$$

where α_1, α_2, … α_r are user-defined weights attached to each derivative of $\mathbf{x}_d$ to indicate its importance relative to position tracking. The solution of Eq.(36) can be obtained using least squares in a manner similar to Eq.(10).

4. ILLUSTRATIVE EXAMPLES

4.1. TRACKING COMPARISON

This section compares the tracking performance of the proposed method with NPZ-ignore, ZPETC and ZMETC using the illustrative first order NMP system studied by Butterworth et al. [9]. It is given by the transfer function

$$G(z^{-1}) = K \frac{1 - az^{-1}}{1 - pz^{-1}}; \;\; K = \frac{1-p}{1-a}, \tag{37}$$

with $p = 0.5$ and $a = \pm 1.1$, sampled at a frequency of 10 kHz. When $a = 1.1$, the NMP zero is in the right half plane (RHP) of the z domain, and when $a = -1.1$ it is in the left half plane (LHP). Table 2 provides the state space parameters of $G(z^{-1})$. Note that $\mathbf{A}_d$, $\mathbf{B}_d$ and $\mathbf{C}_d$ are scalars because it is a first order system with only one state.

TABLE 2: STATE SPACE REPRESENTATION OF $G(z^{-1})$

	$\mathbf{A}_d$	$\mathbf{B}_d$	$\mathbf{C}_d$	D_d
$a = 1.1$ (RHP)	0.5	1	3	−5
$a = -1.1$ (LHP)	0.5	1	0.38	0.24

Figure 2 shows the x-y path of the desired trajectory to be tracked, consisting of motions along the x and y axes, sampled at 10 kHz with a total duration of 0.05 seconds (E = 500). Figure 3 shows the x and y axis position, velocity and acceleration profiles of the desired trajectory as functions of time. Notice that x and y positions have non-zero initial conditions. Each axis is treated as an independent SISO system having the NMP closed-loop transfer function given by Eq.(37). A NURBS curve of degree m = 4 and 201 control points (i.e., n = 200) is used to implement the proposed FBF approach for each axis.

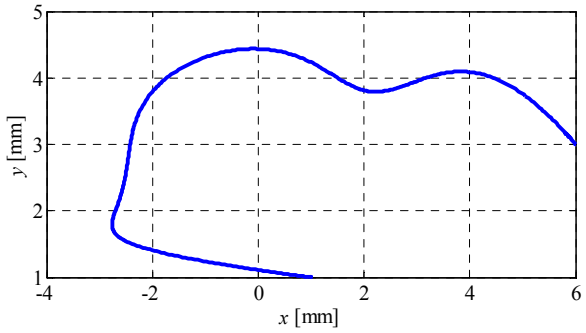

FIGURE 2: X-Y PATH OF DESIRED TRAJECTORY

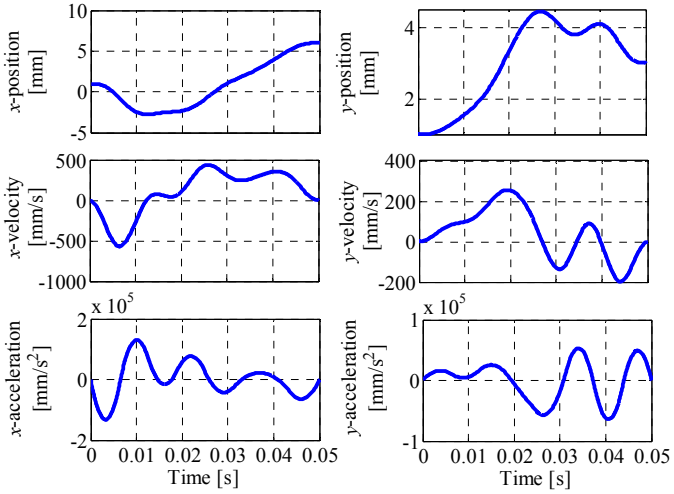

FIGURE 3: POSITION, VELOCITY AND ACCELERATION PROFILES OF DESIRED TRAJECTORY AS A FUNCTIONS OF TIME

For the system defined by Eq.(37), $\hat{c}(k)$ and $\hat{\mathbf{f}}(0)$ are obtained by solving Eq.(18) to get

$$\begin{aligned} &\hat{c}(k) = a^k \hat{c}(0), \\ &\mathbf{C}_d \hat{\mathbf{f}}(0) = -D_d \hat{c}(0). \end{aligned} \tag{38}$$

Note $\hat{c}(k)$ that is an exponential signal. To satisfy Eq.(21) NURBS basis functions must span this exponential signal which is highly unlikely for $n < E$ and a finite set of piecewise polynomial functions like NURBS basis functions. Furthermore, the filter initial states $\mathbf{f}_i(0)$ used to filter the basis functions should satisfy Eq.(38). In this work, the initial states for forward filtering $u_0(k)$, $u_1(k)$, ..., $u_n(k)$ are selected such that $u_i(0) = \tilde{u}_i(0)$ which ensures that the output trajectory can start from a non-zero initial condition. Hence based on Eq.(38), $\mathbf{U}$ and $\mathbf{F}$, for both the RHP and LHP systems, give

$$\text{rank}\left(\begin{bmatrix}\mathbf{U}\\ \mathbf{F}\end{bmatrix}\right)=201 \text{ and } \text{rank}\left(\begin{bmatrix}\mathbf{U} & \hat{\mathbf{c}}\\ \mathbf{F} & \hat{\mathbf{f}}(0)\end{bmatrix}\right)=202. \quad (39)$$

This implies that the filtered basis functions are linearly independent. Linear independence of the basis functions can also be verified based on the fact that $\mathbf{U}$ and $\tilde{\mathbf{U}}$ are rank n+1 = 201 matrices for the RHP as well as the LHP case.

Figure 4 compares the x and y axis tracking errors of NPZ-ignore, ZPETC, ZMETC and FBF as functions of time for the RHP zero system. The same comparison is made in Fig. 5 for the LHP zero system. Table 3 summarizes the x and y axis RMS tracking errors obtained from each plot. Notice that all the methods track better when the system has a LHP zero compared to a RHP zero. However, in both cases, the FBF method outperforms the approximate model inversion techniques in terms of tracking performance by at least one order of magnitude.

Beyond the time domain comparison of Figs. 4 and 5, it is of interest to compare the methods with regard to tracking bandwidth. Tracking performance depends on both magnitude and phase so we define the tracking bandwidth based on the frequency at which $|L|$ deviates from ±3 dB and/or $\angle L$ deviates from ±45°, as shown in Fig. 6. Note that ±45° corresponds to phase angle at ±3 dB of first-order continuous-time filter. The defined tracking bandwidth is therefore the frequency at which L crosses the boundaries of the shaded sector in Fig. 6.

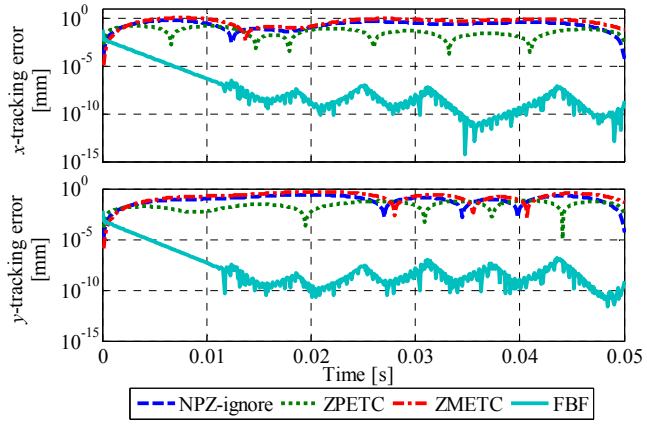

FIGURE 4: COMPARISON OF X AND Y AXIS TRACKING ERRORS FOR NPZ-IGNORE, ZPETC, ZMETC AND PROPOSED FBF METHOD APPLIED TO SYSTEM WITH RHP NMP ZERO (a = 1.1). THE FBF METHOD IS BASED ON A NURBS CURVE WITH 201 CONTROL POINTS

As discussed in Section 2.2, L is an LTI filter for the approximate model inversion techniques. However, each row of $\mathbf{H}$, the equivalent of L for the proposed FBF technique, acts like a separate FIR filter. It is therefore instructive to show the best and worst case bandwidths of all the FIR filters (i.e., all the rows of $\mathbf{H}$). Figure 7 compares the magnitude and phase plots of L (for the approximate model inversion methods) and the best/worst FIR filters of FBF's $\mathbf{H}$ matrix for the RHP zero case; Fig. 8 shows the same plots for the LHP zero case and Table 4 summarizes the tracking bandwidths.

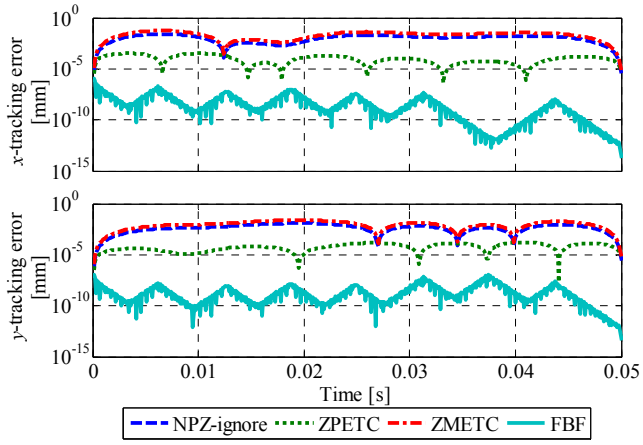

FIGURE 5: COMPARISON OF X AND Y AXIS TRACKING ERRORS FOR NPZ-IGNORE, ZPETC, ZMETC AND PROPOSED FBF METHOD APPLIED TO SYSTEM WITH LHP NMP ZERO (a = −1.1). THE FBF METHOD IS BASED ON A NURBS CURVE WITH 201 CONTROL POINTS

TABLE 3: RMS TRACKING ERROR COMPARISON

		RMS Tracking Error [mm]			
		NPZ-Ignore	ZPETC	ZMETC	FBF
a =1.1 (RHP)	x	2.93×10^{-1}	6.54×10^{-2}	5.74×10^{-1}	2.90×10^{-3}
	y	1.31×10^{-1}	3.58×10^{-2}	2.55×10^{-1}	3.19×10^{-4}
a = −1.1 (LHP)	x	1.39×10^{-2}	1.48×10^{-4}	2.79×10^{-2}	7.07×10^{-8}
	y	6.30×10^{-3}	8.13×10^{-5}	1.25×10^{-2}	1.42×10^{-8}

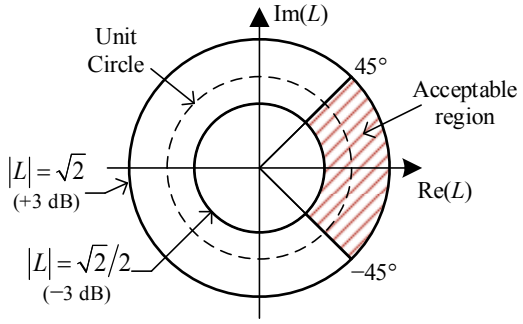

FIGURE 6: COMPOSITE TRACKING BANDWIDTH CRITERION INCLUDING MAGNITUDE AND PHASE DEVIATIONS

One observes that there is a marked difference in the tracking bandwidths of NPZ-ignore, ZPETC and ZMETC when applied to the RHP and LHP zero cases. In each case, the bandwidth is significantly lower for the RHP zero compared to the LHP zero. NPZ-ignore provides the best bandwidth of the three approximate inversion techniques for the RHP zero case as well as the LHP zero case. The worst-case bandwidth of FBF is better than those of the approximate model inversion methods for the RHP zero case. For the LHP zero case, its best-case bandwidth is superior to those of the approximate model

inversion methods while its worst-case bandwidth is better than ZMETC's. Hence its superior performance compared to NPZ-ignore, ZPETC and ZMETC.

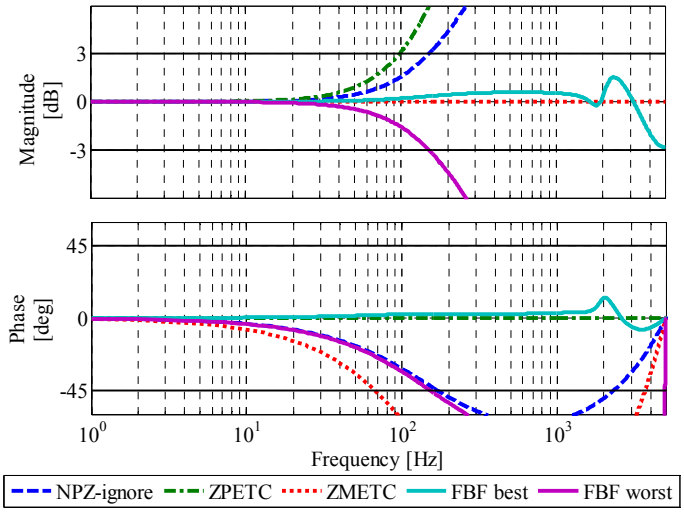

FIGURE 7: TRACKING FREQUENCY RESPONSE FUNCTION COMPARISON FOR a = 1.1. THE FBF METHOD IS BASED ON A NURBS CURVE WITH 201 CONTROL POINTS

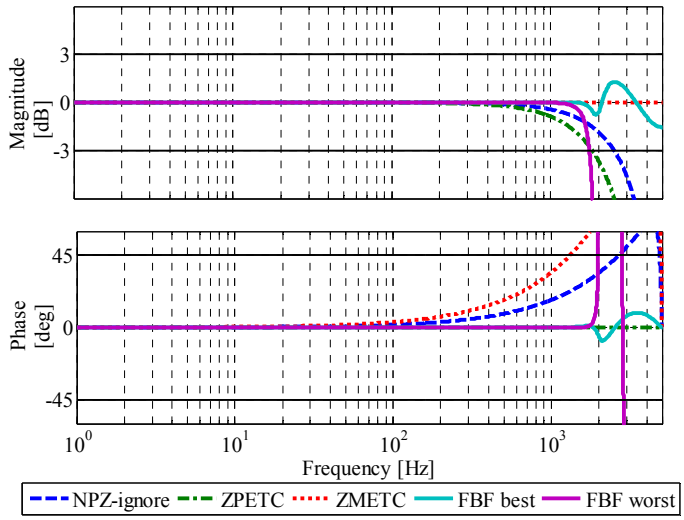

FIGURE 8: TRACKING FREQUENCY RESPONSE FUNCTION COMPARISON FOR a = −1.1. THE FBF METHOD IS BASED ON A NURBS CURVE WITH 201 CONTROL POINTS

TABLE 4: COMPARISON OF TRACKING BANDWIDTHS

		NPZ-Ignore	ZPETC	ZMETC	FBF (best)	FBF (worst)
Band-width [Hz]	a = 1.1 (RHP)	152	98	67	5000	152
	a = −1.1 (LHP)	2504	1823	1317	5000	1744

4.2. EFFECT OF NUMBER OF CONTROL POINTS

In the simulation results presented in Section 4.1 above, n = 200 was selected for the NURBS curve. The number of control points corresponds to the number of basis functions used to approximate $c(k)$. Keeping all other factors the same, the larger the number of basis functions the better the approximation. Therefore, n plays a significant role in the tracking performance of the FBF method. It was mentioned in Section 2.2 that for perfect tracking the matrix **H** should be the identity matrix. Figure 9 shows a plot of **H** for different values of n. Observe that, as n increases, the contributions of the diagonal terms increase and **H** approaches the identity matrix. Figure 10 shows the x-axis tracking error of the system with LHP zero for various values of n. One can see that there are very significant improvements in tracking performance from n = 10 to n = 100. However, the improvements significantly dwindle between n = 100, 200 and 300. Note that the trends shown in Figs. 9 and 10 for n based on the x-axis tracking error of the LHP zero case apply to the other scenarios studied in Section 4.1 above.

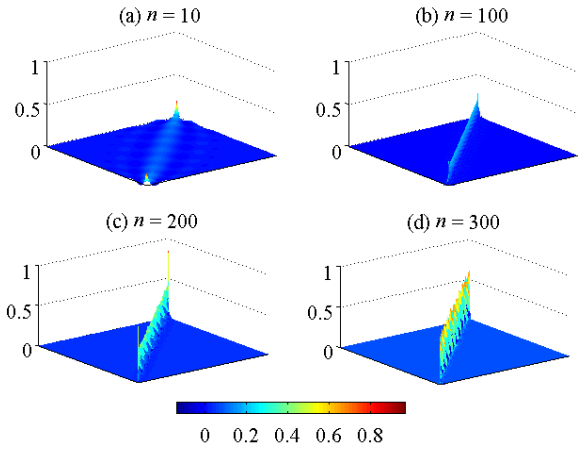

FIGURE 9: PLOTS OF ELEMENTS OF MATRIX **H** FOR DIFFERENT NUMBER OF CONTROL POINTS FOR SYSTEM WITH LHP ZERO

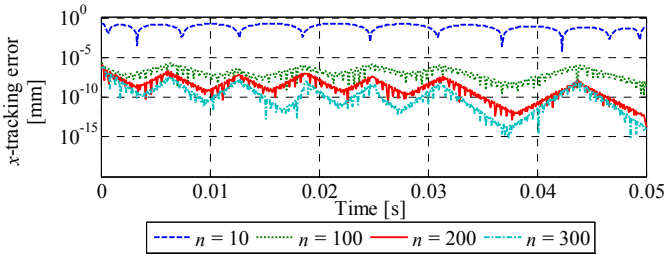

FIGURE 10: X AXIS TRACKING ERROR PROFILE FOR SYSTEM WITH LHP ZERO AS FUNCTION OF NUMBER OF CONTROL POINTS

4.3. EFFECT OF INITIAL CONDITIONS AND HIGHER ORDER DERIVATIVES

Sections 4.1 and 4.2 have focused on minimizing the tracking error. This section focuses on the handling of initial conditions (ICs) and higher order derivatives (HODs), as discussed in Sections 3.2 and 3.3, respectively. Figure 11 and Table 5 compare the results for x axis position, velocity and acceleration errors based on computation of $\mathbf{p}$ using Eqs. (10), (31) and (36) for the LHP system. Notice that the basic calculation of $\mathbf{p}$ using Eq.(10) and NURBS basis functions (with $E = 500$, $n = 15$, $m = 4$) results in large errors in velocity and acceleration because it does not consider ICs and HODs. The enhanced solution, incorporating ICs of Fig. 3, using Eq.(31), reduces velocity and acceleration errors, particularly at the beginning of the motion. The velocity and acceleration errors can be further reduced by incorporating velocity and acceleration information using Eq.(36) (with $\alpha_1 = 2\times10^{-4}$, $\alpha_2 = 2\times10^{-5}$); notice however that the reduction of velocity and acceleration errors have come to some extent at the expense of position errors.

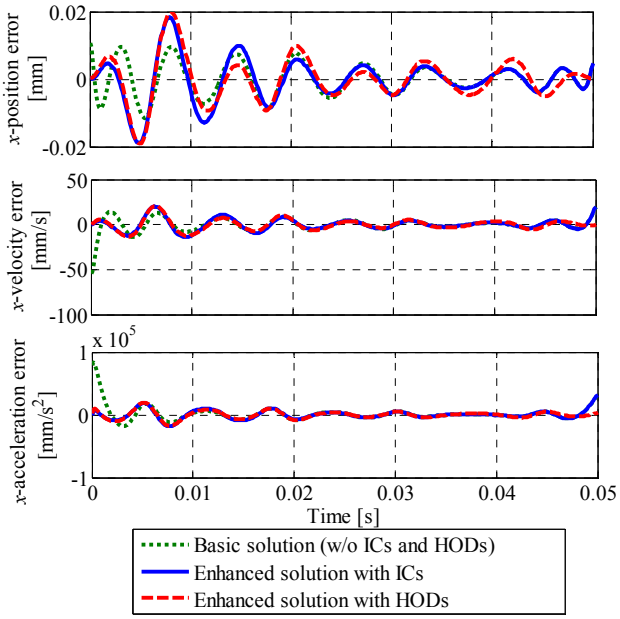

FIGURE 11: X AXIS POSITION, VELOCITY AND ACCELERATION TRACKING ERRORS AS FUNCTIONS OF TIME FOR BASIC AND ENHANCED SOLUTION TECHNIQUES

TABLE 5: TRACKING ERROR COMPARISON FOR BASIC AND ENHANCED SOLUTION TECHNIQUES

		Position error [mm]	Velocity error [mm/s]	Acceleration error [mm/s^2]
Basic	RMS	4.66×10^{-3}	7.24×10^{0}	1.20×10^{4}
	Max	1.15×10^{-2}	5.40×10^{1}	8.34×10^{4}
Enhanced with ICs	RMS	6.25×10^{-3}	6.18×10^{0}	6.97×10^{3}
	Max	1.89×10^{-2}	2.00×10^{1}	3.23×10^{4}
Enhanced with HODs	RMS	6.49×10^{-3}	5.98×10^{0}	6.08×10^{3}
	Max	1.99×10^{-2}	2.05×10^{1}	2.00×10^{4}

5. REMARKS ON VERSATILITY AND PRACTICALITY OF FBF

The FBF method has been presented in the context of discrete time systems but it is also applicable to continuous time systems. The key difference is that in continuous time domain, W and $\tilde{W}$ are function spaces rather than vector spaces. Consequently, the matrix-based least squares solution given in Eq.(10) has to be changed to the minimization of a norm of the form

$$\min_{\{p_i\}} \left\| x_d(t) - \sum_{i=0}^{n} p_i \tilde{u}_i(t) \right\| \tag{40}$$

The continuous time implementation can be achieved, for example, by using the Gram-Schmidt orthonormalization process [26] to construct a set of orthonormal basis in the function space.

It is worth mentioning a practical advantage that the FBF method has over the approximate model inversion techniques with regard to implementation in commercial systems. The control systems used in many commercial applications, e.g., CNC machines and robots, are configured to accept reference commands in the form of parametric curves like NURBS. Moreover, their controllers are closed and so cannot be modified. Therefore, for methods like ZPETC to be used in such systems, $C(z^{-1})$ in Fig. 1 would have to be used to filter $x_d(k)$ (offline) to yield $c(k)$ which would then be parameterized and injected as a reference input to the controller to be executed in real time. This would mean a two-step approximation – first by approximate model inversion and then by parameterization. The FBF approach, on the other hand, would only have the approximation related to parameterization. In such situations, the only drawback that FBF may have is with regards to its offline computation time. For example using Intel® Xeon® processor @ 2.67 GHz and 6 GB RAM, the computation time for generating $c(k)$ offline using ZPETC and parameterizing it using a NURBS curve (with $m = 4$ and $n = 200$) is 0.2496 s, for the trajectory given in Fig. 3. Performing the same task with FBF (using NURBS basis functions with $m = 4$ and $n = 200$) takes 0.5460 s; i.e., 0.2964 s longer.

6. CONCLUSION AND FUTURE WORK

This paper has presented a novel method for realizing accurate tracking control in LTI NMP systems by using filtered basis functions (FBF). The desired trajectory to be tracked is assumed to be entirely known and the output of the tracking controller is represented as a linear combination of user-defined basis functions having unknown coefficients. The basis functions are forward filtered using the modeled dynamics of the NMP system and their coefficients selected to minimize the errors in tracking the desired trajectory. As an example, basis functions derived from the popular NURBS curve are employed to demonstrate the method. Additional benefits of NURBS with regards to incorporating initial conditions and

tracking of higher order derivatives of the desired trajectory are also discussed. Numerical examples are presented, comparing the proposed FBF method with three popular tracking controllers for NMP systems, namely, NPZ-ignore, ZPETC and ZMETC. With a suitable set of basis functions, the FBF method is shown to outperform all three controllers by at least one order of magnitude under two different tracking scenarios. The applicability of the FBF method to continuous-time systems, as well as its practicality and ease of use in commercial control systems are also discussed. Future work will extend the FBF method to MIMO tracking control problems.

7. ACKNOWLEDGMENT

This work is partially funded by the National Science Foundation's CAREER Award #1350202 and the first author's Rackham Centennial Award. Both awards are gratefully acknowledged.

REFERENCES

[1] Tomizuka, M., 1987, "Zero Phase Error Tracking Algorithm for Digital Control," J. Dyn. Syst. Meas. Control, **109**(1), pp. 65–68.

[2] Astram, K., and Wittenmark, B., 1984, Computer Controlled Systems: Theory and Design, Prentice Hall.

[3] Miu, D. K., 1993, Mechatronics, Springer.

[4] Devasia, S., Chen, D., and Paden, B., 1996, "Nonlinear Inversion-Based Output Tracking," Autom. Control. IEEE Trans., **41**(7), pp. 930–942.

[5] Hunt, L., Meyer, G., and Su, R., 1996, "Noncausal Inverses for Linear Systems," Autom. Control. IEEE Trans., **41**(4), pp. 608–611.

[6] Zou, Q., and Devasia, S., 1999, "Preview-Based Stable-Inversion for Output Tracking of Linear Systems," J. Dyn. Syst. Meas. Control, **121**(4), pp. 625–630.

[7] Marconi, L., Marro, G., and Melchiorri, C., 2001, "A Solution Technique for Almost Perfect Tracking of Non-Minimum-Phase, Discrete-Time Linear Systems," Int. J. Control, **74**(5), pp. 496–506.

[8] Benosman, M., and Le Vey, G., 2003, "Stable Inversion of SISO Nonminimum Phase Linear Systems through Output Planning: An Experimental Application to the One-Link Flexible Manipulator," IEEE Trans. Control Syst. Technol., **11**(4), pp. 588–597.

[9] Butterworth, J. A., Pao, L. Y., and Abramovitch, D. Y., 2012, "Analysis and Comparison of Three Discrete-Time Feedforward Model-Inverse Control Techniques for Nonminimum-Phase Systems," Mechatronics, **22**(5), pp. 577–587.

[10] Haack, B., and Tomizuka, M., 1991, "The Effect of Adding Zeroes to Feedforward Controllers," J. Dyn. Syst. Meas. Control, **113**(1), pp. 6–10.

[11] Torfs, D., De Schutter, J., and Swevers, J., 1992, "Extended Bandwidth Zero Phase Error Tracking Control of Nonminimal Phase Systems," J. Dyn. Syst. Meas. Control, **114**(3), pp. 347–351.

[12] Gross, E., and Tomizuka, M., 1994, "Experimental Flexible Beam Tip Tracking Control with a Truncated Series Approximation to Uncancelable Inverse Dynamics," Control Syst. Technol. IEEE Trans., **2**(4), pp. 382–391.

[13] Weck, M., and Ye, G., 1990, "Sharp Corner Tracking Using the IKF Control Strategy," CIRP Ann. Technol., **39**(1), pp. 437–441.

[14] Wen, J. T., and Potsaid, B., 2004, "An Experimental Study of a High Performance Motion Control System," American Control Conference, 2004. Proceedings of the 2004, pp. 5158–5163.

[15] Rigney, B. P., Pao, L. Y., and Lawrence, D. A., 2009, "Nonminimum Phase Dynamic Inversion for Settle Time Applications," IEEE Trans. Control Syst. Technol., **17**(5), pp. 989–1005.

[16] Butterworth, J. A., Pao, L. Y., and Abramovitch, D. Y., 2009, "A Comparison of Control Architectures for Atomic Force Microscopes," Asian J. Control, **11**(2), pp. 175–181.

[17] Piegl, L., and Tiller, W., 1997, The NURBS Book, Springer Verlag, Berlin, Berlin.

[18] Piegl, L., 1991, "On NURBS: A Survey," IEEE Comput. Graph. Appl., **11**(1), pp. 55–71.

[19] Bay, J. S., 1999, Fundamentals of Linear State Space Systems, McGraw-Hill Science, Engineering & Mathematics.

[20] Hoaglin, D. C., and Welsch, R. E., 1978, "The Hat Matrix in Regression and ANOVA," Am. Stat., **32**(1), pp. 17–22.

[21] Strang, G., 1993, "The Fundamental Theorem of Linear Algebra," Am. Math. Mon., **100**(9), pp. 848–855.

[22] Cheng, C.-W., and Tsai, M.-C., 2004, "Real-Time Variable Feed Rate NURBS Curve Interpolator for CNC Machining," Int. J. Adv. Manuf. Technol., **23**(11-12), pp. 865–873.

[23] Shima, A., Sasaki, T., Ohtsuki, T., and Wakinotani, Y., 1996, "64-bit RISC-Based Series 15 NURBS Interpolation," FANUC Tech. Rev., **9**(1), pp. 23–28.

[24] Rogers, D. F., 2000, An Introduction to NURBS: With Historical Perspective, Elsevier.

[25] Ueng, W.-D., Lai, J.-Y., and Tsai, Y.-C., 2007, "Unconstrained and Constrained Curve Fitting for Reverse Engineering," Int. J. Adv. Manuf. Technol., **33**(11-12), pp. 1189–1203.

[26] Sasane, A., "Functional Analysis and its Applications," London Sch. Econ.