Feedforward Learning Control with Application to Trajectory Tracking of a Flexible Beam

W. Cheng J.T.Wen
Department of Electrical, Computer, and Systems Engineering
Rensselaer Polytechnic Institute
Troy, NY 12180–3590, USA
cheng@ral.rpi.edu wen@ral.rpi.edu

Abstract

This paper presents a simple approach to the feedforward compensation in the tip tracking control of a flexible beam. The approach is based on probing the system with a finite number of independent input signals and then fit the corresponding output signals to a desired output trajectory. Since neither the dynamic model nor the full state of the beam is required, and the method applies to the tracking of a fixed, finite time output trajectory, we consider the scheme as a learning control strategy. Due to the assumed linearity, this problem is transformed into a standard least square problem. Controller saturation can be incorporated, resulting in a constrained least square problem. The basic method has also been generalized to linear time varying systems and nonlinear systems. Experimental and simulation results are presented to demonstrate the overall viability.

Key words: Learning Control, Flexible Beam, Feedforward Control, Tracking Control, Robot Tracking.

1 Introduction

The control of a planar flexible beam has long been used as a simple precursor of the general control problem of more complex flexible structures, such as articulated robots with link flexibility, disk drive arm, flexible space structures, etc. Most past approaches require the beam dynamic model for its implementation [1, 2, 3, 4]. Through analytic modeling and experimental identification, it is possible to obtain a model that is sufficiently accurate for the design of a stabilizing feedback controller. However, when output trajectory tracking is also sought, feedback controller alone or together with a model based feedforward usually still leaves a sizable tracking error. Adaptive controllers have been proposed a one solution [5, 6] but they suffer from parameter adaptation transient and complex real time control laws. Another alternative is off-line or iterative learning controllers. Most of the existing learning controllers only apply to the tracking control of rigid robots since they require all states of the plant [7, 8, 9, 10, 11]. In [12], a method is proposed which requires only the measured output of the plant, but a strong strict positive realness condition is needed. The dimension of the state space of a flexible beam is very high (infinite in the ideal case). It is therefore impractical to measure all the states. The output of interest we consider here is the position of the tip of the beam. With respect to the hub torque input, the transfer function is non–minimum–phase. These facts limit the application of the existing learning control methods to this problem.

Various model based feedforward compensation strategies have been proposed in the past [13, 3, 14, 4]. In all these cases, there is an underlying stabilizing feedback controller which has already been designed. When the input/output transfer function is non–minimum–phase, it is known that either a non-causal feedforward can be used for exact output tracking [13] or a causal feedforward can be used for asymptotic output tracking [4]. In this paper, we follow the latter approach and present a model independent method to find the required feedforward signal. The underlying controller is assumed to be hub proportional and derivative (PD) feedback, which globally asymptotically stabilizes the origin, but any other stabilizing controller (e.g., passivity based controller in [15]). The basic idea to find the feedforward is simple: 1. Choose a finite basis for the input space. 2. Apply each input basis to the actual system to generate an output basis. 3. Solve a least square problem that fits the desired output to the output basis. 4. Use the solution of the least square problem to form the required feedforward from the input basis functions. To ensure the outputs corresponding to the input basis functions

1050-4729/93 $3.00 © 1993 IEEE
are linearly independent, a mild rank condition needs to be satisfied.

The rest of the paper is organized as follows. Section 2 presents the problem statement. Section 3 introduces the proposed feedforward learning controller. Section 4 gives the experiment results of a flexible beam and simulation results of a rigid robot.

2 Problem Statement

Considering a linear time invariant system:

\[
\begin{align*}
\dot{x} &= Ax + Bu \\
y &= Cx
\end{align*}
\]  

(1)

where \( A \in \mathbb{R}^{n\times n} \), \( B \in \mathbb{R}^{n\times p} \) and \( C \in \mathbb{R}^{d\times n} \). Since we only consider feedforward control (feedback stabilization loop is already closed), we assume \( A \) is Hurwitz.

The objective is to find \( u \in \mathcal{U} \), \( \mathcal{U} \) is a subspace of \( L_2([0,T];\mathbb{R}^p) \) such that \( \|y - y_d\| \) is minimized (\( \|\cdot\| \) is the \( L_2([0,T];\mathbb{R}^p) \) norm) for a given \( y_d \) that is continuous in \([0,T]\).

For our experimental work, we consider a single flexible beam the hub of which is driven by a control torque about the vertical axis and the tip is free. The output of interest is the tip position of the beam.

The modal truncated model of the flexible beam is of the following form:

\[
\ddot{q} + D\dot{q} + \Omega^2 q = \dot{b} \tau
\]

(2)

where \( q \) is the mode amplitude, \( D \) is a positive semidefinite damping matrix, \( \tau \) is the control torque at the hub, \( \Omega^2 \) is a diagonal matrix consisting of the square of the resonant angular frequencies, and the input matrix \( \dot{b} \) is

\[
\dot{b} = \frac{1}{p} [\Psi_0(0), \Psi_1(0), \ldots]^T
\]

where \( \Psi_i \)'s are the mode shapes. Consider a control law of the following form:

\[
\tau = \tau_{fb} + u
\]

where \( \tau_{fb} \) is a feedback control that asymptotically stabilizes the origin and \( u \) is the feedforward control which we will select based on the output trajectory tracking criterion. A particularly simple \( \tau_{fb} \) that can be used is the PD feedback of the hub position (we have used this choice in all our experiments). More sophisticated feedback control can also be used without affecting the subsequent argument. Define the state as \( x = [q^T, \dot{q}^T]^T \) and select the output of interest as the tip position of the beam, the closed loop (with the hub PD feedback) system is of the form (1) with

\[
A = \begin{bmatrix} 0 & I \\ \Omega^2 - K_p \hat{b} \hat{b}^T & -D - K_v \hat{b} \hat{b}^T \end{bmatrix}
\]

\[
B = \begin{bmatrix} 0 \\ \dot{b} \end{bmatrix}
\]

\[
C = \begin{bmatrix} c \\ 0 \end{bmatrix}
\]

where \( c = [\Psi_0(L), \Psi_1(L), \ldots] \), and \( K_p \) and \( K_v \) are the hub PD feedback gains. As stated before, it can be shown that \( A \) is Hurwitz for any \( K_p > 0, K_v > 0 \) [4].

3 Feedforward Learning Control Strategies

Write the Equation (1) as follows:

\[
\begin{align*}
\dot{x} &= Ax + \sum_{i=1}^{p} B_i u_i \quad ; \quad x(0) = 0 \\
y &= Cx
\end{align*}
\]

(3)

where \( B_i \) is the \( i^{th} \) column of the \( B \) matrix and \( u_i \) is the \( i^{th} \) control variable. We have chosen the initial state to be the zero state since the origin is asymptotically stable when \( u_i = 0 \) for all \( i \).

We shall choose \( u_i \) as a linear combination of selected basis functions \( \{H_j\}_{j=1}^m \in L_2([0,T];\mathbb{R}^p) \). In other words,

\[
\mathcal{U}_i = \text{span}\{H_j : j = 1, \ldots, m\} \quad \mathcal{U} = \mathcal{U}_1 \times \mathcal{U}_2 \times \ldots \times \mathcal{U}_p
\]

(4)

and each \( u_i \) can be represented as

\[
u_i(t) = \sum_{j=1}^{m} W_{ij} H_j(t)
\]

(5)

where \( W_{ij}, i = 1, \ldots, p, j = 1, \ldots, m \) are constant scalar weights. The objective is now to find a set of these weights \( W_{ij} \) so that a norm of the output tracking error is minimized in some sense.

By substituting (5) into (1), we have

\[
\begin{align*}
\dot{x} &= Ax + \sum_{i=1}^{p} \sum_{j=1}^{m} W_{ij} B_i H_j \\
y &= Cx
\end{align*}
\]

(6)

By linearity, we have

\[
\begin{align*}
x &= \sum_{i=1}^{p} \sum_{j=1}^{m} W_{ij} \xi_{ij} \\
y &= \sum_{i=1}^{p} \sum_{j=1}^{m} W_{ij} C \xi_{ij} = \sum_{i=1}^{p} \sum_{j=1}^{m} W_{ij} \eta_{ij}
\end{align*}
\]
where $\xi_{ij}$'s and $\eta_{ij}$'s are the state and output trajectories corresponding to the $j$th basis function $H_j$ feeding into the $i$th input channel (with all other input channels set to zero). Therefore, $\xi_{ij}$ and $\eta_{ij}$ satisfy

$$
\dot{\xi}_{ij} = A\xi_{ij} + B_i H_j, \quad \xi_{ij}(0) = 0
$$
$$
\eta_{ij} = C\xi_{ij}
$$
(7)

Since $A$ is Hurwitz, $\eta_{ij}$ will be uniformly bounded for $t \in [0, T]$.

Denote $\eta_{ij}$ by $\eta_{ij} = G_i H_j$ where

$$
G_i H_j \triangleq \int_0^t C e^{A(t-s)} B_i H_j(s) \, ds.
$$
(8)

Then $\{\eta_{ij} : j = 1, \ldots, m\}$ is an independent set of signals if and only if

$$
\begin{bmatrix}
CB_i \\
CAB_i \\
\vdots \\
CAN^{i-1}B_i
\end{bmatrix}
$$

is full rank.

This is usually a very mild condition. Hence, if the rank condition above is satisfied for any input channel $i$, then among the $m \cdot p$ output signals $\eta_{ij}$, there are at least $m$ independent signals.

For the purpose of constraining the feedforward input based on the actuator saturation, we define $\tilde{u}_{ij}$ as the feedback control portion of the original input for the $i$th subsystem described by (7) (i.e., $B_i \tilde{u}_{ij}$ is the feedback control portion in $A\xi_{ij}$).

With a specified desired output trajectory $y_d$, the output tracking error with a general input

$$
u = \begin{bmatrix}
\sum_{j=1}^m W_{1j} H_j \\
\vdots \\
\sum_{j=1}^m W_{pj} H_j
\end{bmatrix}
$$

is

$$e = y - y_d = \sum_{i=1}^p \sum_{j=1}^m W_{ij} \eta_{ij} - y_d.
$$
(10)

Now the original output tracking problem can be posed as a least square problem:

Find $W_{ij}$, $i = 1, \ldots, p$, $j = 1, \ldots, m$, to minimize

$$J_1 = \int_0^T e^T(t) Q(t) e(t) \, dt
$$

where $Q(t)$ is some positive semidefinite weighting matrix.

If feedforward itself is costly, then one can modify the optimization index to

$$J_2 = \int_0^T \left( e^T(t) Q(t) e(t) + u^T(t) R(t) u(t) \right) \, dt.
$$

Both cases above are standard weighted least square problems and can be readily solved using pseudo-inverse operators.

If the total input cannot exceed a hard saturation bound, then the following set of inequalities can be added:

$$\left| \sum_{j=1}^m W_{ij}(H_j + \tilde{u}_{ij}) \right| \leq u_{max}
$$
(11)

for $i = 1, \ldots, p$. The ensuing constrained least square problem can then be solved by quadratic programming. In the actual implementation, all signals are actually uniformly sampled, so the above least square problems and quadratic programming problems only involve finite matrices and vectors.

In summary, the proposed strategy to generate the feedforward control involves the following steps:

1. First feedback stabilize the open loop system until the desired disturbance rejection and robustness properties are obtained.

2. Excite the closed loop system (1) at the $i$th input with the $j$th chosen basis functions $H_j(t)$ while holding other inputs zero. Repeat this for all input channels and all basis functions. Store the output trajectories and feedback control signals in $\eta_{ij}$ and $\tilde{u}$

3. Solve either the least square problem with objective function $J_1$ or $J_2$, or a quadratic programming problem with objective function $J_1$ or $J_2$ and constraint (11).

4. The final feedforward control law for the $i$th input is $u_i(t) = \sum_{j=1}^m W_{ij} H_j(t)$, where $W_{ij}$'s are obtained from step 3.

Remark:

1. Since $A$ is Hurwitz, even when the initial state is not zero either during excitation with input basis functions or playback using the final feedforward, the tracking error will converge to the expected level asymptotically. The system will of course also retain all the properties of the feedback stabilized system (such as disturbance rejection, insensitivity to noise, etc.).

2. The weights $W_{ij}$ for the learning controller encode the information of the desired trajectory, while $\eta_{ij}$ encode the information of the plant. Therefore, it is unnecessary to excite the plant again when only the desired trajectory changes — only the least square problem needs to be solved with the new trajectory. However, when the plant changes,
e.g., payload changes, the plant has to be excited again to obtain a new set of \( \eta_j \). Therefore, this method is not suitable for on-line adaptation of random parametric changes in the plant.

3. Only linearity is used in the derivation; the proposed method is equally applicable to infinite dimensional systems.

The learning scheme just presented can also be applied to linear time varying systems as long as the plant is internally asymptotically stable and the dynamics repeatable for each input basis excitation. We have also begun preliminary investigation in the extension to nonlinear time invariant systems. Consider the following nonlinear system:

\[
\begin{align*}
\dot{x} &= f(x) + g(x)u \\
y &= h(x)
\end{align*}
\]

where \( f, g \) are unknown nonlinear mappings and \( u \) is the feedforward. For simplicity of presentation, assume \( u \in \mathbb{R} \).

Let the feedforward control input in the \( k \)-th iteration be \( u^k(t) \); we have:

\[
\begin{align*}
\dot{x}^k &= f(x^k) + g(x^k)u^k \\
y^k &= h(x^k)
\end{align*}
\]

where \( u^k \) is the feedforward in the \( k \)-th iteration. Linearizing the nonlinear system around \( x^k, u^k \), we have the following time varying linear system:

\[
\begin{align*}
\delta x^k &= A_k \delta x^k + B_k \delta u^k \\
\delta y^k &= C_k \delta x^k
\end{align*}
\]

where \( \delta x^k = x^{k+1} - x^k, \delta u^k = u^{k+1} - u^k \), and \( \delta y^k = y^{k+1} - y^k \), and

\[
\begin{align*}
A_k &= \frac{\partial f(x)}{\partial x} \bigg|_{x^k} + \frac{\partial g(x)}{\partial x} \bigg|_{x^k} u^k \\
B_k &= g(x^k), \quad C_k = \frac{\partial h(x)}{\partial x} \bigg|_{x^k}
\end{align*}
\]

An important assumption that we make at this stage (this will need future research to further justify) is that the origin of this system is asymptotically stable. System (14) is a linear time varying system, but its dynamical property is the same if the same \( x^k \) and \( u^k \) are used. Now we can apply the off-line learning to drive \( \delta y^k \) to its desired trajectory. Since we would like to minimize \( \|y_d - y^{k+1}\| = \|y_d - y^k - y^{k+1} + y^k\| = \|y_d - y^k - \delta y^k\| \), the desired trajectory for \( \delta y^k \) can be chosen as \( y_d = y_d - y^k \).

Based on the algorithm of the off-line learning for linear systems, we can summarize the off-line learning algorithm for nonlinear systems as below:

Initialize \( u^0 \) and \( \eta_j^0 \) to zero functions in the interval \([0, T]\), \( j = 1, \ldots, m \).

For \( k = 1, \ldots, k_{\text{max}} \), do:

1. Using the feedforward \( u^{k+1} = H_j + u^k, j = 1, \ldots, m \), to excite the plant and record the corresponding output \( \eta_j^{k+1} \), where \( H_j \) is the \( j \)-th input basis function. Define \( \delta \eta_j^k(t) = \eta_j^{k+1}(t) - \eta_j^k(t) \).

2. Find the weight vector \( W \) to minimize the following objective function:

\[
J = \int_0^T e^T(t)Q(t)e(t)dt
\]

\[
\text{e} = \delta \eta_j^k - \sum_{j=1}^m W_j \delta \eta_j^k.
\]

3. Use the new feedforward \( u^{k+1} = \sum_{j=1}^m W_j H_j + u^k \) to drive the plant and observe the corresponding output \( y^{k+1} \). If \( \|y_d - y^{k+1}\| \) is within the given tolerance, stop, otherwise, continue.

4 Experiment and Simulation

In this section, we will present experimental results for a flexible beam and simulation results for one-link and two-link robot arms with frictions. The detail of the experimental setup can be found in [15].

Hub PD feedback is used as the underlying controller. A higher performance controller can also be designed as in [15], but we have chosen a simple controller to clearly demonstrate the effect of the feedforward. The feedback gains of the hub position and hub velocity are: \( K_p = 2.378 \) (NM/Rad) and \( K_v = 1.1651 \) (NM/Rad/sec). A minimal jerk trajectory and a sinusoid are chosen as the desired trajectories.

The duration of the desired trajectory is 6 seconds. Two different types of basis functions are used:

**Sinusoids:** There are 25 basis functions, constant function, \( \sin(nt) \) and \( \cos(nt) \), \( n = 1, \ldots, 12 \). The peak amplitudes of all signals are 1000 d/a units.

**Square Waves:** There are 25 basis functions, constant function, 12 square wave signals with frequencies \( \frac{\pi}{2^3}, n = 1, \ldots, 12 \), and the same signals shifted by 90°. The peak amplitudes of all signals are 1000 d/a units.

In order to verify the feasibility of the basic approach, we did not optimize the input basis functions in any way. However, the tracking results as shown in Figures 1-2 are quite good (certainly much better than
PD alone) with either type of basis functions. More experiment results can be found in [16].

We also present simulations results of a single rigid link with Coulomb friction (the values are taken from joint 1 of a PUMA 560 arm) and a two-link rigid arm to demonstrate the applicability to nonlinear systems. The simulation results are shown in Figures 3–4. In both cases, the learning control substantially reduces the tracking error of the PD feedback alone. The number of iterations for the single link case is 6 and 10 for the two-link case. The sinusoidal basis described before has been used for these simulations (so the number of excitation for each iteration is 25).

5 Conclusion and Future Work

In this paper, we have presented a simple off-line learning strategy to find a feedforward signal that improves the output tracking performance. The procedure requires exciting the system with a set of input functions and then fit the desired output to a linear combination of the basis output functions. Minimizing the output tracking error can be solved as a standard least square problem. If actuator constraints are incorporated, the problem becomes a quadratic programming problem. We have applied this approach in both simulation and experimentation to a single flexible beam with very good results. We have also begun preliminary investigation to generalization to nonlinear systems. Simulation results to multiple-link robots with frictions are encouraging. Our future work will focus on encoding the feedforward information obtained by learning in a neural network and extend the
Figure 4: Joint Position Tracking Error for a Two-Link Arm

learning paradigm to disturbance rejection.

Acknowledgment

This work was supported in part by the National Science Foundation under Grant No.MSS-9113633 and Army Research Office under grant DAAL03-92-G-012.

References


