ROBUST LEARNING CONTROL

by

Greg Heinzinger, Dan Fenwick, Brad Paden,
and Fumio Miyazaki

Memorandum No. UCB/ERL M89/115

25 September 1989

ROBUST LEARNING CONTROL

by

Greg Heinzinger, Dan Fenwick, Brad Paden,
and Fumio Miyazaki

Memorandum No. UCB/ERL M89/115

25 September 1989

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

ROBUST LEARNING CONTROL

by

Greg Heinzinger, Dan Fenwick, Brad Paden,
and Fumio Miyazaki

Memorandum No. UCB/ERL M89/115

25 September 1989

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Robust Learning Control

Greg Heinzinger,* Dan Fenwick,† Brad Paden,‡ Fumio Miyazaki§

Abstract

In this paper the robustness of a class of learning control algorithms to state disturbances, output noise, and errors in initial conditions is studied. We present a simple learning algorithm and exhibit, via a concise proof, bounds on the asymptotic trajectory errors for the *learned* input and the corresponding state and output trajectories. Furthermore, these bounds are continuous functions of the bounds on the initial condition errors, state disturbance, and output noise, and the bounds are zero in the absence of these disturbances.

*Research supported by the Berkeley Fellowship; by project MICRO of the University of California: funded by Semiconductor Research Corporation, Harris Corp., Intel, Phillips Research Labs, Fairchild, Hewlett Packard, Texas Instruments, AMD, Inference Corp., Siemens Corp. This author is at the University of California at Berkeley.

†Research supported by the National Science Foundation through Grant no. 8421415. This author is at the University of California at Santa Barbara.

‡Research supported by the National Science Foundation through Grant no. 8421415. This author is at the University of California at Santa Barbara.

§This author is at Osaka University, Osaka, JAPAN.

Notation

$\|\cdot\|$	Euclidean norm
$\|h(\cdot)\|_\lambda$	λ norm

Robust Learning Control

Greg Heinzinger, Dan Fenwick, Brad Paden, Fumio Miyazaki

1 Introduction

Learning control is a name attributed to a class of self-tuning processes whereby the system performance of a specified task improves based on the previous performances of identical tasks. This is an advantage when controlling systems that can not be modeled accurately. But the idea of a self-learning system is in itself aesthetically appealing in that it represents a significant step in the development of an intelligent, fully autonomous control system.

Consider the "learning" system depicted in Figure 1. $u_i(\cdot)$ denotes an input trajectory. P represents a plant whose desired output trajectory is $y_d(\cdot)$ and whose actual output due to $u_i(\cdot)$ is $y_i(\cdot)$. l is a learning operator which compares $y_d(\cdot)$ and $y_i(\cdot)$ and adds an update term to $u_i(\cdot)$ to produce $u_{i+1}(\cdot)$. These trajectories are taken to be functions of $t \in [0, T]$ and the updates occur sequentially in time. The trajectories are supported on finite intervals of the time axis and the iteration from i to $i+1$ occurs from one interval to the next. In a sense, $u_i(\cdot)$ is a parameter which is adaptively tuned. In contrast to typical adaptive control schemes, the parameter $u_i(\cdot)$ belongs to an infinite dimensional space.

As an example, a learning control scheme for a robot manipulator would record measurements as it moved an object from point A to point B; it would then use this data to improve its performance the next time it moved the same object from point A to point B. In some applications the need to repeat a trajectory multiple times is a distinct disadvantage of learning control. However, in many applications repetitive tasks are commonly performed making learning control a very natural solution. Another advantage of learning control is that it is easy to implement and allows simple models and control schemes to be used while compensating for unmodeled dynamics and complex phenomenon such as stiction. It is also appealing because it is similar to some of our own learning processes; we may practice a task (say throwing a ball) many times before we are able to find inputs to a complex system (our body) to accomplish the task.

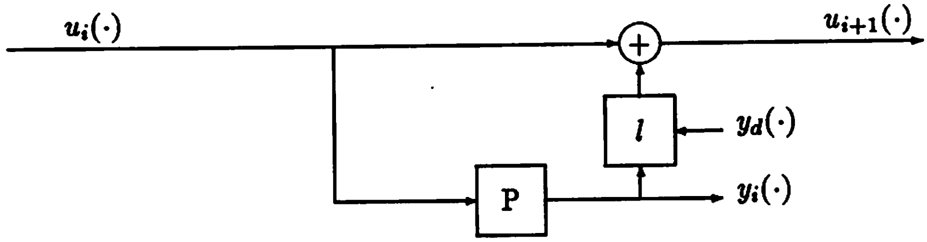


Figure 1: Basic learning system

Recently there have been a number of efforts toward defining and analyzing learning control schemes [1,2,3,4,5]. In model-based learning schemes (Atkeson [4]) the inputs corresponding to the desired and actual trajectories are computed from estimated system parameters and the resulting input errors fed to the learning operator. In this scheme, the performance of the algorithm depends on the quality of the parameter estimates. A more common approach is to operate on the output errors directly, and the model-based learning scheme in [4] is shown in [5] to be a special case of this more general approach. The basic strategy of these techniques is to use an iteration of the form $u_{i+1}(\cdot) = l\,(u_i(\cdot), y_d(\cdot) - y_i(\cdot))$, where the operator $l(\cdot,\cdot)$ remains to be specified.

For time-invariant mechanical systems Arimoto et al. [1] and Craig [2] present conditions on the learning operator which guarantee system convergence upon repeated application of the learning algorithm. One shortcoming of these analyses is that they are small signal analyses which require the assumption that the initial trajectory (and thus all subsequent ones) lies in a neighborhood of the desired trajectory. In addition, no investigation is presented on the size or existence of these neighborhoods. Both Hauser [5] and Bondi et al. [3] remove this assumption by developing global analyses, proving convergence of the input sequence $u_i(\cdot)$ with any initial trajectory. Another extension of Hauser [5] allows time-varying systems. This is important because we wish to improve the performance of the plant as much as possible using conventional feedback control methods. The learned input, $u_i(\cdot)$, is a feed forward term which further improves the performance for a specific task. Thus, for most applications we have the situation shown in Figure 2, and the learning algorithm operates on the system between $u_i(\cdot)$ and $y_i(\cdot)$ which is time-varying.

In this paper we consider the robustness of the learning algorithm. Specifically, does the performance of the algorithm continuously degrade as errors and disturbances are introduced? For a practical implementation we would like to know that the learning algorithm causes the input, state, and output trajectory errors to be asymptotically bounded when there are (1) errors in the initial state, (2) bounded state disturbances, and (3) bounded

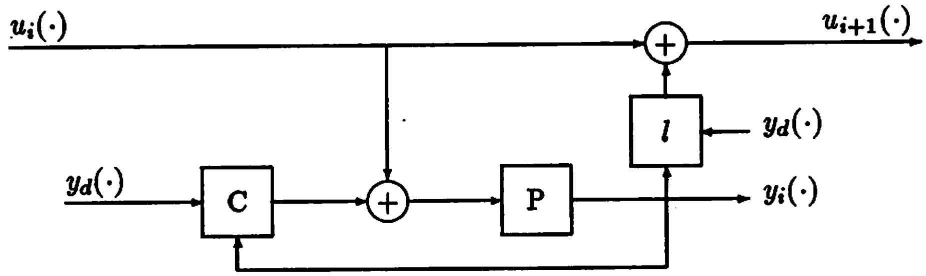


Figure 2: Learning control application for a plant with a feedback controller attached.

output disturbances. In addition, we would like to understand how these bounds depend on the disturbances, and they should decrease to zero as the disturbances do.

In the last few years researchers have begun to answer these questions. Arimoto et al. [6] deal with time-invariant mechanical systems and use a small signal analysis to demonstrate robustness to initial state errors and differentiable state disturbances; the proof once again assumes the initial trajectory is in some small neighborhood. In [3] Bondi et al. present a global analysis of robustness for time-invariant mechanical systems. However, neither of these papers answers all the questions posed above and both deal with time-invariant mechanical systems.

In this paper we consider the following class of nonlinear, time-varying systems described by the following state space equations:

$$\begin{aligned} \dot{x}(t) &= f(x(t),t) + B(x(t),t)u(t) \\ y(t) &= g(x(t),t). \end{aligned} \tag{1.1}$$

As mentioned before, this is significant because we can apply our results to a plant and feedback configuration as shown in Figure 2. The form of the learning operator studied often and the one we study operates on the derivative of the previous output error in a memoryless linear fashion:

$$u_{i+1}(t) = u_i(t) + L\left(\dot{y}_d(t) - \dot{y}_i(t)\right) \quad , \tag{1.2}$$

where L is a memoryless linear map. We choose L to be linear and memoryless because this is sufficient for robust convergence and it allows for a very simple implementation. It is also shown in this paper that we can include terms in the update law which act on the previous output error and its integral.

The inclusion of this more general class of systems represents the main contribution of this paper. In addition, the global proof of robustness we present is simple, concise and complete, and it makes explicit the dependence of the bounds on the disturbances and errors.

The remainder of the paper is structured as follows: Section 2 presents a simple learning control example. Section 3 states the problem formally and presents our results. Section 4 examines additional examples and implications of our conditions. Section 5 discusses some implementation issues.

2 An Example

In this section we present a simple example to illustrate the concept of learning control. Consider the one degree-of-freedom spring-damper system given by the differential equation

$$m\ddot{q} + c\dot{q} + h(q) = v, \tag{2.1}$$

where q is the position, m is the mass, c is the damping coefficient, and the continuous function $h(\cdot)$ represents the (possibly nonlinear) spring force. We rewrite (2.1) as a first order dynamical system:

$$\begin{aligned} \dot{x} &= \begin{pmatrix} \dot{q} \\ -\frac{1}{m}\left(h(q) + c\dot{q}\right) \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{m} \end{pmatrix} v \\ y &= (0\ 1)x = \dot{q} \end{aligned} \tag{2.2}$$

where $x = \begin{pmatrix} q \\ \dot{q} \end{pmatrix}$. Our objective is to find an input, $v(t)$, such that a desired twice continuously differentiable trajectory, $q_d(t)$ for $t \in [0, T]$, is followed.

If the system parameters are known exactly and $q(0) = q_d(0)$ and $\dot{q}(0) = \dot{q}_d(0)$, we can write down the solution from (2.1) and use the control law $v(t) = m\ddot{q}_d(t) + c\dot{q}_d(t) + h(q_d(t)) + k_v(\dot{q}_d(t) - \dot{q}(t)) + k_p(q_d(t) - q(t)) + u_0(t)$ where $u_0(\cdot) \equiv 0$. However, in many situations we have only estimates of the parameters ($\hat{m}$, $\hat{c}$, $\hat{h}(\cdot)$), the initial state may vary ($q(0) \approx q_d(0)$, $\dot{q}(0) \approx \dot{q}_d(0)$), and there may be unmodeled dynamics. The basic idea of learning control in this example is to attempt the task (following $q_d(t)$ starting from a given initial state $(q_i(0), \dot{q}_i(0))$), compare the output with the desired output, and adjust the "learned" input $u_i(\cdot)$ for the next attempt. We use the following update law:

$$u_{i+1}(t) = u_i(t) + \hat{m}\left(\ddot{q}_d(t) - \ddot{q}_i(t)\right).$$

In this situation the learned input is updated by the difference between the desired and actual forces exerted on the mass. In the next section we show that $q_i(\cdot) \to q_d(\cdot)$ as i increases if $|1 - \hat{m}\frac{1}{m}| < 1$. This implies that to guarantee convergence $\hat{m}$ must be in $(0, 2m)$. As expected the better the estimate of m the better the convergence rate will be, as shown in the next section. Intuitively we expect this algorithm to work since we are adjusting the force exerted on the mass to be that required to follow the trajectory ($m\ddot{q}_d(t)$). Requiring $\hat{m} > 0$ implies that we are updating in the correct direction and $\hat{m} < 2m$ prevents the algorithm from "over-correcting".

3 A Robust Learning Control Algorithm

In this section we present a robust learning algorithm for a class of time-varying, nonlinear systems. By robust we mean that when state disturbances are present or there are errors in the initial conditions our learning algorithm generates a sequence of inputs such that the asymptotic trajectory errors for the input, state, and output are bounded. In addition, these bounds are continuous functions of the bounds on the initial condition errors and the disturbances, and we quantify the degradation due to each of these factors.

The description of the system, assumptions, notation, and update law are similar to those in [5]; and the proof technique is similar to many in that it proceeds in a straightforward manner showing that we have a "contraction" on the input sequence implying the convergence results.

As specified in Equation (1.1) the class of nonlinear, time-varying systems considered is described by the following state-space equations:

$$\begin{aligned} \dot{x}_i(t) &= f(x_i(t),t) + B(x_i(t),t)u_i(t) + w_i(t) \\ y_i(t) &= g(x_i(t),t) \end{aligned} \tag{3.1}$$

where, for all $t \in [0,T]$, $x_i(t) \in \mathbf{R}^n$, $u_i(t) \in \mathbf{R}^r$, $y_i(t) \in \mathbf{R}^m$, and $w_i(t) \in \mathbf{R}^n$. The functions $f : \mathbf{R}^n \times [0,T] \to \mathbf{R}^n$ and $B : \mathbf{R}^n \times [0,T] \to \mathbf{R}^{n\times r}$ are piecewise continuous in t; and $g : \mathbf{R}^n \times [0,T] \to \mathbf{R}^m$ is differentiable in x and t, with partial derivatives $g_x(\cdot,\cdot)$ and $g_t(\cdot,\cdot)$. We consider inputs $u_i : [0,T] \to \mathbf{R}^r$, not necessarily continuous. In addition, we assume the following properties:

(A1) For each fixed $x_i(0)$ with $w_i(\cdot) \equiv 0$ the output map $\mathcal{O} : C([0,T],\mathbf{R}^r) \times \mathbf{R}^n \to C([0,T],\mathbf{R}^m)$ and the state map $\mathcal{S} : C([0,T],\mathbf{R}^r) \times \mathbf{R}^n \to C([0,T],\mathbf{R}^n)$ are one-to-one. In this notation $y_i(\cdot) = \mathcal{O}(u_i(\cdot), x_i(0))$ and $x_i(\cdot) = \mathcal{S}(u_i(\cdot), x_i(0))$.

(A2) The disturbance $w_i(\cdot)$ is bounded by b_w on $[0,T]$ (i.e. $\|w_i(t)\| \leq b_w$ on the interval $[0,T]$).

(A3) The functions $f(\cdot,\cdot)$, $B(\cdot,\cdot)$, $g_x(\cdot,\cdot)$, and $g_t(\cdot,\cdot)$ are uniformly globally Lipschitz in x on the interval $[0,T]$. That is, $\|h(x_1,t) - h(x_2,t)\| \leq k_h\|x_1(t) - x_2(t)\| \quad \forall\, t \in [0,T]$ and some $k_h < \infty \in \mathbf{R}$ ($h \in \{f, B, g_x, g_t\}$).

(A4) The operators $B(\cdot,\cdot)$ and $g_x(\cdot,\cdot)$ are bounded on $\mathbf{R}^n \times [0,T]$.

(A5) All functions are assumed to be measurable and integrable.

Assumption (A1) implies that given an achievable, desired output trajectory (y_d) and initial state ($x_d(0)$), there exist unique input (u_d) and state (x_d) trajectories corresponding to this output trajectory. Assumption (A4) on $g_x(\cdot,\cdot)$ implies that g is uniformly globally Lipschitz in x on $[0,T]$.

The function $w_i(t)$ represents both deterministic and random disturbances of the system; it may be stiction, non-reproducible friction, modeling errors, etc. This is important to include since these are present in physical systems. Assumption (A2) restricts these disturbances to be bounded, but they may be discontinuous (e.g. stiction in mechanical systems).

We consider the update law given by

$$u_{i+1}(t) = (1-\gamma)u_i(t) + \gamma u_0(t) + L(y_i(t),t)\left[\dot{y}_d(t) - \dot{y}_i(t)\right] \qquad 0 \le \gamma < 1, \tag{3.2}$$

where $L : \mathbf{R}^m \times [0,T] \rightarrow \mathbf{R}^{r\times m}$ is bounded. This learning operator is similar to those used in [5,6,1,7,8] in that it updates the system input in an *affine* fashion.

We include γ to allow the influence of a bias term. This may prevent the input from wandering too much initially. In addition, γ may be allowed to vary with the iteration to further improve performance; we leave it fixed for simplicity of presentation. The use of γ will be discussed in Section 5, and for an initial reading γ is best ignored (i.e. $\gamma = 0$) since its inclusion does not alter the proof in any essential way.

The following norm is used to simplify the expression of our results.

Definition 3.3 *We define the λ norm for a function $h : [0,T] \rightarrow \mathbf{R}^k$ by*

$$\|h(\cdot)\|_\lambda \triangleq \sup_{t\in[0,T]} e^{-\lambda t}\|h(t)\|. \tag{3.3}$$

Remark: From this definition we can see that $\|h\|_\lambda \le \|h\|_\infty \le e^{\lambda T}\|h\|_\lambda$ for $\lambda > 0$ (where $\|h\|_\infty \triangleq \sup_{t\in[0,T]} \|h(t)\|$), implying that these two norms are equivalent. Thus convergence results can be proved using either norm.

For clarification of the remaining discussion, function parameters will be shown in subscript notation with the dependence on time implied unless otherwise stated. In particular,

$$\begin{array}{ccc}
g_{xi} \triangleq \frac{\partial}{\partial x} g(x,t)|_{x=x_i(t)}, & g_{xd} \triangleq \frac{\partial}{\partial x} g(x,t)|_{x=x_d(t)}, & g_{ti} \triangleq \frac{\partial}{\partial t} g(x,t)|_{x=x_i(t)}, \\
g_{td} \triangleq \frac{\partial}{\partial t} g(x,t)|_{x=x_d(t)}, & f_i \triangleq f(x_i(t),t), & f_d \triangleq f(x_d(t),t), \\
u_i \triangleq u_i(t), & u_d \triangleq u_d(t), & w_i \triangleq w_i(t), \\
B_i \triangleq B(x_i(t),t), & B_d \triangleq B(x_d(t),t), & L_i \triangleq L(y_i(t),t)
\end{array}$$

and k_{gx},k_{gt}, k_f, k_B, k_g are the Lipschitz constants for $g_x(\cdot,\cdot)$, $g_t(\cdot,\cdot)$, $f(\cdot,\cdot)$, $B(\cdot,\cdot)$, and $g(\cdot,\cdot)$ respectively. We now state the main result of this paper.

Theorem 3.4 *Let the system described by (3.1) satisfy assumptions (A1)–(A5) and use the update law (3.2). Given an attainable $y_d(\cdot)$, if*

$$\left\|(1-\gamma)I - L(g(x,t),t)g_x(x,t)B(x,t)\right\| \leq \rho < 1 \qquad \forall(x,t) \in \mathbf{R}^n \times [0,T]$$

and the initial state error is bounded ($\|x_d(0) - x_i(0)\| \leq b_{x0}$), then as $i \to \infty$ the error between u_i and u_d is bounded. In addition, the state and output asymptotic errors are bounded. These bounds depend continuously on the bound on the initial state error, bound on the state disturbance, and γ; as b_{x0}, b_w, and γ tend to zero, these bounds also tend to zero.

The main idea of the proof is to show that $\|\delta u_{i+1}\|_\lambda \leq \bar{\rho}\|\delta u_i\|_\lambda + \varepsilon$ where $0 \leq \bar{\rho} < 1$ ($\delta u_i \triangleq u_d - u_i$). This implies that $\limsup_{i\to\infty} \|\delta u_i\|_\lambda \leq \frac{1}{1-\bar{\rho}}\varepsilon$. Thus the main portion of the proof is a calculation to show that this relationship holds. The results follow quickly once we have established this. Intuitively, the condition on L says that if we push on the system (through u), we can observe a change in the output, and we take an appropriate action to reduce the error.

Proof. From (3.1) and the update law (3.2), the error for the iterate $i+1$ can be written as

$$\begin{aligned} u_d - u_{i+1} &= u_d - (1-\gamma)u_i - \gamma u_0 - L_i\left[\dot{y}_d - \dot{y}_i\right] &(3.5)\\ &= (1-\gamma)(u_d - u_i) + \gamma(u_d - u_0) \\ &\quad - L_i\left[g_{xd}(f_d + B_d u_d) + g_{td} - g_{xi}(f_i + B_i u_i + w_i) - g_{ti}\right] &(3.6)\\ &= \left[(1-\gamma)I - L_i g_{xi} B_i\right](u_d - u_i) + \gamma(u_d - u_0) \\ &\quad - L_i\left[g_{xd}(f_d + B_d u_d) + g_{td} - g_{xi}(f_i + B_i u_d + w_i) - g_{ti}\right]. &(3.7)\end{aligned}$$

Taking norms, using the bounds, and using the Lipschitz conditions yields

$$\begin{aligned} \|u_d - u_{i+1}\| &\leq \|(1-\gamma)I - L_i g_{xi} B_i\|\|u_d - u_i\| + \gamma\|u_d - u_0\| \\ &\quad + \|L_i\| \left\{ \begin{array}{l} \|g_{xd} - g_{xi}\|\|f_d + B_d u_d\| \\ + \|g_{xi}\|\left[\|f_d - f_i\| + \|B_d - B_i\|\|u_d\| + \|w_i\|\right] \\ + \|g_{td} - g_{ti}\| \end{array} \right\} &(3.8)\\ &\leq \rho\|u_d - u_i\| + \gamma\|u_d - u_0\| \\ &\quad + b_L \left\{ \begin{array}{l} k_{gx}\|x_d - x_i\|b_d \\ + b_{gx}\left[k_f\|x_d - x_i\| + k_B\|x_d - x_i\|b_{ud} + b_w\right] \\ + k_{gt}\|x_d - x_i\| \end{array} \right\}, &(3.9)\end{aligned}$$

where b_L, b_{gx} are the norm bounds for $L(\cdot,\cdot)$, $g_x(\cdot,\cdot)$ respectively and

$$b_d \triangleq \sup_{t\in[0,T]} \|f_d + B_d u_d\| \quad \text{and} \quad b_{ud} \triangleq \sup_{t\in[0,T]} \|u_d\|.$$

Defining $k_1 \triangleq b_L[k_{gx}b_d + b_{gx}(k_f + k_B b_{ud}) + k_{gt}]$ equation (3.9) simplifies to

$$\|u_d - u_{i+1}\| \leq \rho\|u_d - u_i\| + k_1\|x_d - x_i\| + b_L b_{gx} b_w + \gamma\|u_d - u_0\|. \tag{3.10}$$

Now writing the integral expression for $x(t)$ (with the quantities in the integral being functions of τ) and taking norms we obtain

$$\begin{aligned}
\|x_d - x_i\| &= \Big\|x_d(0) - x_i(0) \\
&\quad + \int_0^t \Big((f_d + B_d u_d) - (f_i + B_i u_i + w_i)\Big)d\tau\Big\| & (3.11)\\
&\leq \|x_d(0) - x_i(0)\| \\
&\quad + \int_0^t \left(\begin{array}{c} \|f_d - f_i\| + \|B_d - B_i\|\|u_d\| \\ + \|B_i\|\|u_d - u_i\| + \|w_i\| \end{array} \right) d\tau & (3.12)\\
&\leq \|x_d(0) - x_i(0)\| \\
&\quad + \int_0^t \Big((k_f + k_B b_{ud})\|x_d - x_i\| + b_B\|u_d - u_i\| + b_w\Big)d\tau, & (3.13)
\end{aligned}$$

where b_B is the norm bound on $B(\cdot,\cdot)$. Defining $k_3 \triangleq (k_f + k_B b_{ud})$, $\|\delta x_i\| \triangleq \|x_d - x_i\|$, and using a basic integral inequality (see [9] p. 96) we have that

$$\begin{aligned}
\|\delta x_i\| &\leq \|x_d(0) - x_i(0)\|e^{k_3 t} \\
&\quad + \int_0^t e^{k_3(t-\tau)}\Big(b_B\|u_d(\tau) - u_i(\tau)\| + b_w\Big)d\tau. & (3.14)
\end{aligned}$$

Combining equations (3.10) and (3.14) yields

$$\begin{aligned}
\|\delta u_{i+1}\| &\leq \rho\|\delta u_i\| + k_1\left\{\|\delta x_i(0)\|e^{k_3 t} + \int_0^t e^{k_3(t-\tau)}(b_B\|\delta u_i(\tau)\| + b_w)d\tau\right\} \\
&\quad + b_L b_{gx} b_w + \gamma\|\delta u_0\| & (3.15)\\
&\leq \rho\|\delta u_i\| + k_1 b_B \int_0^t e^{k_3(t-\tau)}\|\delta u_i(\tau)\|d\tau \\
&\quad + k_1\|\delta x_i(0)\|e^{k_3 t} + k_1 b_w \int_0^t e^{k_3(t-\tau)}d\tau \\
&\quad + b_L b_{gx} b_w + \gamma\|\delta u_0\|. & (3.16)
\end{aligned}$$

Multiplying (3.16) by $e^{-\lambda t}$, defining $k \triangleq \max\{k_1 b_B, k_3\}$, and assuming $\lambda > k$ we have that

$$\begin{aligned}
e^{-\lambda t}\|\delta u_{i+1}\| &\leq \rho e^{-\lambda t}\|\delta u_i\| + k\int_0^t e^{-\lambda\tau}\|\delta u_i(\tau)\|e^{(k-\lambda)(t-\tau)}d\tau \\
&\quad + k_1\|\delta x_i(0)\|e^{(k_3-\lambda)t} + k_1 b_w \int_0^t e^{-\lambda\tau}e^{(k_3-\lambda)(t-\tau)}d\tau \\
&\quad + b_L b_{gx} b_w e^{-\lambda t} + \gamma e^{-\lambda t}\|\delta u_0\|. & (3.17)
\end{aligned}$$

Noticing that the integrals are strictly increasing and that for a constant $\|k\|_\lambda = k$ we obtain

$$\begin{aligned} \|\delta u_{i+1}\|_\lambda &\leq \left[\rho + \frac{k}{\lambda - k}(1 - e^{(k-\lambda)T})\right] \|\delta u_i\|_\lambda \\ &\quad + k_1\|\delta x_i(0)\| + \frac{k_1 b_w}{\lambda - k_3}\left(1 - e^{(k_3-\lambda)T}\right) \\ &\quad + b_L b_{gx} b_w + \gamma\|\delta u_0\|_\lambda. \end{aligned} \tag{3.18}$$

Defining $\bar{\rho} = \rho + \frac{k}{\lambda - k}(1 - e^{(k-\lambda)T})$ and $k_4 \triangleq b_L b_{gx} + \frac{k_1}{\lambda - k_3}\left(1 - e^{(k_3-\lambda)T}\right)$, we have that

$$\|\delta u_{i+1}\|_\lambda \leq \bar{\rho}\|\delta u_i\|_\lambda + k_1\|\delta x_i(0)\| + k_4 b_w + \gamma\|\delta u_0\|_\lambda \tag{3.19}$$

$$\|\delta u_{i+1}\|_\lambda \leq \bar{\rho}\|\delta u_i\|_\lambda + \varepsilon. \tag{3.20}$$

Where ε combines the norm bounds of the initial state errors, state disturbances, and bias contribution. Since $\rho < 1$, we can find a $\lambda > k$ which makes $\bar{\rho} < 1$. By Lemma A.1, u_i converges to the neighborhood of u_d of radius $\left(\frac{1}{1-\bar{\rho}}\right)\varepsilon$ with respect to the λ norm. Thus

$$\limsup_{i\to\infty} \|\delta u_i\|_\lambda \leq \left(\frac{1}{1-\bar{\rho}}\right)\varepsilon. \tag{3.21}$$

Using equation (3.14), and similar manipulations we obtain

$$\begin{aligned} \|\delta x_i\|_\lambda &\leq \|\delta x_i(0)\| + \int_0^t e^{(k_3-\lambda)(t-\tau)}\|\delta u_i\|_\lambda d\tau && (3.22) \\ &\leq \|\delta x_i(0)\| + \frac{1}{\lambda - k_3}\left(1 - e^{(k_3-\lambda)T}\right)\|\delta u_i\|_\lambda. && (3.23) \end{aligned}$$

So $\limsup_{i\to\infty} \|\delta x_i\|_\lambda \leq \|\delta x_i(0)\| + \frac{1}{\lambda - k_3}\left(1 - e^{(k_3-\lambda)T}\right)\left(\frac{1}{1-\bar{\rho}}\right)\varepsilon$.

To obtain the result for y_i we use the fact that g is Lipschitz in x. Thus $\|\delta y_i\|_\lambda \leq k_g\|\delta x_i\|_\lambda$, with $\|\delta x_i\|_\lambda$ being bounded as above. ■

Equation (3.19) clearly illustrates the influence of the initial state error, state disturbance, and bias term in degrading our bound on the asymptotic errors. We see that this bound on the degradation is continuous in these factors. Furthermore, in the absence of these terms $\varepsilon = 0$, and we have convergence of the algorithm to the desired trajectories.

We now state and prove robust learning for a few useful extensions of our update law.

Corollary 3.24 *If we replace (3.2) with*

$$\begin{aligned} u_{i+1}(t) &= (1-\gamma)u_i(t) + \gamma u_0(t) + L(y_i(t), t)\left[\dot{y}_d(t) - \dot{y}_i(t)\right] \\ &\quad + K(y_i(t), t)\left[y_d(t) - y_i(t)\right] \end{aligned}$$

with $K(\cdot,\cdot)$ bounded, then Theorem 3.4 still holds.

Proof. Using the fact that g is Lipschitz in x and the condition on K it follows that $\|K_i(y_d - y_i)\| \leq b_K k_g \|x_d - x_i\|$. Thus we modify k_1 in (3.10) by adding $b_K k_g$ and proceed exactly as in the proof of Theorem 3.4. ∎

Remark: This update law is referred to as a "PD" scheme.

Corollary 3.25 *If in addition to the conditions in Corollary 3.24 we have measurement noise, $y_i(t) = g(x_i(t), t) + v_i(t)$, such that $\|v_i(t)\| \leq b_{v1}$ and $\|\dot{v}_i(t)\| \leq b_{v2}$, then we obtain the same results as in Theorem 3.4.*

Proof. The proof proceeds the same as in the proof of the Theorem 3.4; equation (3.19) is modified by adding the term $b_K b_{v1} + b_L b_{v2}$. ∎

Corollary 3.26 *If we modify the learning operator as follows*

$$\begin{aligned} u_{i+1}(t) &= (1-\gamma)u_i(t) + \gamma u_0(t) + L(y_i(t), t)\left[\dot{y}_d(t) - \dot{y}_i(t)\right] \\ &\quad + Q(y_i(t), t)\left[\int_0^t (y_d(\tau) - y_i(\tau))d\tau\right] \end{aligned}$$

with $Q(\cdot,\cdot)$ bounded, then we obtain the same results as in Theorem 3.4.

Proof. From the bound on Q we have that

$$\|Q_i \int_0^t (y_d(\tau) - y_i(\tau))d\tau\| \leq b_Q k_g \int_0^t \|x_d(\tau) - x_i(\tau)\| d\tau. \tag{3.27}$$

Letting $RHS(t)$ denote the right hand side of (3.14) evaluated at time t, we see that $RHS(s) \leq RHS(t)$ for $s \leq t$. This together with (3.27) and the fact that $t \leq T$, allows us to modify k_1 in (3.10) by adding $b_Q k_g T$. Then we proceed exactly as in the proof of the Theorem 3.4. ∎

These corollaries are proved independently so they could be combined to show that a "PID" update law, operating on a system satisfying the conditions of Theorem 3.4, is robust to initial state errors, state disturbances, and output noise.

4 Examples and Implications

In this section we apply the learning algorithm to the dynamics of a robot manipulator and examine the implications of our results. In addition, we present an example to demonstrate the need for differentiation in the update law.

4.1 Robot Example

If $\Theta \in \mathbf{R}^r$ is the vector of joint angles of a robot manipulator, then we can write the dynamics as

$$M(\Theta)\ddot{\Theta} + C(\Theta,\dot{\Theta})\dot{\Theta} + F(\dot{\Theta}) + G(\Theta) = \tau \tag{4.1}$$

or as

$$\begin{aligned} \dot{x} &= \begin{pmatrix} \dot{\Theta} \\ -M^{-1}(\Theta)\big(G(\Theta) + C(\Theta,\dot{\Theta})\dot{\Theta} + F(\dot{\Theta})\big) \end{pmatrix} + \begin{pmatrix} 0 \\ M^{-1}(\Theta) \end{pmatrix}\tau \\ y &= (C_1 \ C_2)\, x, \end{aligned} \tag{4.2}$$

where $x = \begin{pmatrix} \Theta \\ \dot{\Theta} \end{pmatrix}$ and τ is the vector of joint torques which we assume are bounded. $M(\Theta)$ is the bounded, positive definite inertia matrix for the manipulator, $C(\Theta, \dot{\Theta})\dot{\Theta}$ is the coriolis and centripetal forces, $G(\Theta)$ is the gravity forces, $F(\dot{\Theta})$ is the viscous friction forces, and $C_1, C_2 \in \mathbf{R}^{m\times r}$ (all are assumed to be smooth). The update law examined is

$$\tau_{i+1} = \tau_i + L(y_i(t),t)\Big[\dot{y}_d(t) - \dot{y}_i(t)\Big]. \tag{4.3}$$

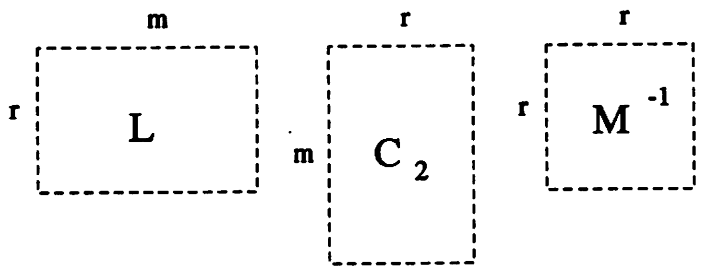
To apply our results we need to check that assumptions A1-A5 are satisfied. For open-kinematic-chain manipulators with bounded configuration space (i.e. $\theta_i \in [a_i, b_i]$ for prismatic joints) we know that A1 is satisfied, we assume A2 & A5, and we see that A3-A4 are satisfied provided we establish a bound on $\dot{\Theta}$. The bound on $\dot{\Theta}$ implies that the state space is bounded and contained within a compact set. This guarantees that the desired functions, which are continuously differentiable, are Lipschitz on the state space. Thus we are left to show that $\dot{\Theta}$ is bounded. This is done by noticing that the power dissipated by friction, $< F(\dot{\Theta}), \dot{\Theta} >$, increases as $\|\dot{\Theta}\|^2$, whereas, the power added to the system, $< M(\Theta)\tau, \dot{\Theta} >$, increases as $\|\dot{\Theta}\|$, implying a bound on $\|\dot{\Theta}\|$ since $\|\tau\|$ was assumed to be bounded. In addition, if a particular joint torque exceeds the torque bounds we project back into the allowable set of torques, this is discussed further in Section 5.

Theorem 3.4 implies that given a desired trajectory the torques will converge, even in the presences of disturbances, to a neighborhood of the desired torque trajectory providing that $\|I - Lg_xB\| \le \rho < 1$. Thus we ask, what does this condition imply?

We observe that $g_x = (C_1 \ C_2)$, and looking at the structure of B we have

$$Lg_xB = LC_2M^{-1}.$$

For $\|I - LC_2M^{-1}\|$ to be less than one we observe that it is necessary for C_2 to have full



column rank, which requires that m (the number of outputs) is greater than or equal to r (the number of joints). In essence we need measurements of each joint velocity, so for simplicity we let $C_2 = \begin{pmatrix} 0 \\ I \end{pmatrix}$, so the last r entries of y are $\dot{\Theta}$. Letting $L = (L_1 \ L_2)$, where $L_1 \in \mathbb{R}^{r\times(m-r)}$ and $L_2 \in \mathbb{R}^{r\times r}$, we see that the condition becomes $\|I - L_2 M^{-1}\| \le \rho < 1$ which gives a condition on the accuracy of the dynamical model of M that we must have. We are free to choose L_1 without destroying the convergence of the algorithm. Since the update law differentiates the output, we are using acceleration data for each of the joints to insure robust learning.

4.2 Another Example

Through the use of a simple example we hope to make plausible the claim that for robust learning with mechanical systems the update law must contain derivatives of the output.[1] We consider the one degree-of-freedom system described by (2.1) and (2.2). We specify the desired trajectory to be $q_d(\cdot) \equiv 0$, implying $x_d(\cdot) \equiv \begin{pmatrix} 0 \\ 0 \end{pmatrix}$. Consider the update law

$$u_{i+1}(t) \;=\; u_i(t) + K(q_i(t), \dot{q}_i(t))\left[\dot{q}_d(t) - \dot{q}_i(t)\right].$$

If $x_i(0) = \begin{pmatrix} 0 \\ v \end{pmatrix}$ and $u_0(\cdot) \equiv 0$ then

$$u_1(0) \;=\; 0 - K(0, v)v$$

[1]It is possible to consider systems for which the relative degree between the input and the output is not one (see [10] for definitions), however, since we are mainly interested in mechanical systems with force inputs and assume we can measure velocities we concentrate on relative degree one systems. For a SISO system, if the relative degree between the input and the output is ν, we could use the update law $u_{i+1}(t) = u_i(t) + L(y_i(t), t)\left[\frac{d^\nu}{dt^\nu} y_d(t) - \frac{d^\nu}{dt^\nu} y_i(t)\right]$ and the condition on the learning operator becomes $|1 - L(g(x,t),t) \cdot L_b L_f^{\nu-1} g| = |1 - L(g(x,t),t) \cdot L_f^{\nu-1} gb| \le \rho < 1$. To obtain the results in Theorem 3.4 we would need to add additional conditions on the derivatives of f, g, and B (e.g. $\frac{\partial^2}{\partial t^2} g(x,t)$ is Lipschitz in x). For MIMO systems the conditions are analogous. In these analyses we must also be concerned with the relative degree between the state disturbances and the output. If the state disturbances are discontinuous we can't differentiate y more than the relative degree between the state disturbances and the output.

$$
\begin{aligned}
u_2(0) &= -K(0,v)v - K(0,v)v \\
&\vdots \\
u_n(0) &= -nK(0,v)v,
\end{aligned}
$$

which is a divergent sum. In addition, adding terms depending on $x_d - x_i$ or on $\int x_d - x_i$ to the update law would not improve the situation since these would both be zero at $t = 0$ in the example considered. In the literature update laws such as the one above have been proposed, and convergence can be proven in the absence of disturbances. However, these update laws are not robust.

5 Discussion

The learning update laws presented in Section 3 guarantee robust learning providing certain conditions are met, and this is the main issue which we have researched. However, in practice there may be modifications which improve the performance of the algorithm.

If at any time the update law produces an input that we know to be out of range, and the allowable set of inputs is a convex set, then projecting back into this set will improve performance. This is evident from our analysis since this will further decrease $\|u_d - u_{i+1}\|$. For many robot manipulators the allowable set of joint torques is a hypercube and projection is easily implemented.

Theorem 3.4 implies that as the iteration number approaches ∞ the trajectory errors are less than certain bounds. However, in many applications we desire to stop the process in a finite time and we desire the error to be as small as possible at this time. For this situation the bias term may be helpful, and varying the update operator as the iterations progress may improve performance. The bias term is initially useful to keep the input from wandering excessively if we have a reasonable expected trajectory for the input, but with time we want to decrease its influence by decreasing γ. Once the input has converged fairly well we may want to begin decreasing the learning gain (the size of L) to cause the input to average out random disturbances, thus improving the accuracy of the final input that we choose after a finite time. It is easily seen that these modifications do not change the results of Theorem 3.4 provided that the condition on the update law is satisfied for all L_i and γ_i.

The class of systems considered is fairly general, and a closer examination of the results reveals that what is essential is that when we apply an input to the system, we can observe a corresponding output, and we act upon this output with the learning operator. The stability of the system may affect the convergence rate but not the actual convergence of the learning algorithm.

It is important to remember that learning control is not a form of dynamic feedback. It can't be used to stabilize a system nor to change its performance for a general trajectory. Thus in applications it is desirable to use a robust feedback controller to improve the system performance, and as explain earlier this is the motivation for considering time-varying systems. Learning control iteratively updates a feed forward term to provide a finer and finer "open loop" performance along a specific trajectory, it is not intended to make up for a poor feedback controller design.

In conclusion, we believe that the learning algorithm presented is applicable to a wide variety of problems. The proof of robust learning allows us to use the learning algorithm with confidence in applications. Moreover, for a particular system and task our results allow bounds to be calculated on the degradation in performance due to each disturbance. We further conjecture that these results can be extended to other update laws, allowing the differentiation to be replaced by say a lead filter with better noise response; this constitutes an interesting area for future research.

Acknowledgements

The authors would like to express their thanks to Robert Minnichelli, Richard Murray and Paul Jacobs for their help in the form of discussions and suggestions.

References

[1] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of robots by learning," *Journal of Robotic Systems*, vol. 1, no. 2, pp. 440–447, 1984.

[2] J. J. Craig, "Adaptive control of manipulators through repeated trials," in *American Control Conference*, (San Diego, CA.), pp. 1566–1573, June 1984.

[3] P. Bondi, G. Casalino, and L. Gambardella, "On the iterative learning control theory for robotic manipulators," *IEEE Journal of Robotics and Automation*, vol. 4, pp. 14–22, Feb. 1988.

[4] C. G. Atkeson and J. McIntyre, "Robot trajectory learning through practice," in *IEEE Conference on Robotics and Automation*, (San Francisco, CA), pp. 1737–1742, April 1986.

[5] J. Hauser, "Learning control for a class of nonlinear systems," in 26^{th} *IEEE Conference on Decision and Control*, (Los Angeles, CA), pp. 859–860, Dec. 1987.

[6] S. Arimoto, S. Kawamura, and F. Miyazaki, "Convergence, Stability, and Robustness of Learning Control Schemes for Robot Manipulators," in *Int. Sym. on Robot Manipulators: Modeling, Control and Education*, (Albuquerque, NM), Nov. 1986.

[7] S. Arimoto, S. Kawamura, and F. Miyazaki, "Mathematical theory of learning with applications to robot control," in *Adaptive and Learning Systems*, (K. S. Narendra, ed.), Plenum Publishing Corp., 1986.

[8] M. Togai and O. Yamano, "Learning control and its optimality: Analysis and its application to controlling industrial robots," in *24th IEEE Conference on Decision and Control*, (Fort Lauderdale, FL), pp. 248–253, Dec. 1985.

[9] T. M. Flett, *Differential Analysis*. Cambridge: Cambridge University Press, 1980.

[10] A. Isidori, *Nonlinear Control Systems: An Introduction*. Berlin: Springer-Verlag, 1989.

Appendix A

Lemma A.1 *If* $\{a_i\}_{i=0}^{\infty}$ *is a sequence of real numbers such that*

$$|a_{i+1}| \leq \rho |a_i| + \varepsilon \qquad 0 \leq \rho < 1 \tag{A.2}$$

then

$$\limsup_{i \to \infty} |a_i| \leq \left(\frac{1}{1-\rho}\right) \varepsilon. \tag{A.3}$$

Proof. Iterating equation (A.2) we obtain

$$\begin{aligned} |a_1| &\leq \rho |a_0| + \varepsilon \\ |a_2| &\leq \rho^2 |a_0| + (1+\rho)\varepsilon \\ &\vdots \\ |a_i| &\leq \rho^i |a_0| + \sum_{j=0}^{i-1} \rho^j \varepsilon \quad = \quad \rho^i |a_0| + \left(\frac{1-\rho^i}{1-\rho}\right) \varepsilon. \end{aligned} \tag{A.4}$$

So as $i \to \infty$, $\rho^i \to 0$ implying that $\limsup_{i \to \infty} |a_i| \leq \left(\frac{1}{1-\rho}\right) \varepsilon$. ■