

Optimization-Based Drift Prevention for Learning Control of Underdetermined Linear and Weakly Nonlinear Time-Varying Systems*

Brian J. Driessen
Structural Dynamics Department
Sandia National Laboratories
Albuquerque, NM 87185-0847
bjdries@sandia.gov

Nader Sadegh
Department of Mechanical Engineering
Georgia Institute of Technology
Atlanta, GA 30332
nader.sadegh@me.gatech.edu

Kwan S. Kwok
Robotics Center
Sandia National Laboratories
Albuquerque, NM 87185-1003
kskwok@sandia.gov

Abstract

In this paper an optimization-based method of drift prevention is presented for learning control of underdetermined linear and weakly nonlinear time-varying dynamic systems. By defining a fictitious cost function and the associated model-based sub-optimality conditions, a new set of equations results, whose solution is unique, thus preventing large drifts from the initial input. Moreover, in the limiting case where the modeling error approaches zero, the input that the proposed method converges to is the unique feasible (zero error) input that minimizes the fictitious cost function, in the linear case, and locally minimizes it in the (weakly) nonlinear case. Otherwise, under mild restrictions on the modeling error, the method converges to a feasible sub-optimal input.

1. Introduction

Learning control is a method of control that feeds the system inputs for a specific task repetitively and uses the actual on-line measured response of the system to evaluate the quality or goodness of the input. The actual responses are used in a feedback loop in which the inputs are adjusted to reduce measured errors in the output.

Example applications include robotics and manufacturing where a certain output tracking task is to be performed repeatedly. Usually the output is the position or velocity history of the robot's joints although sometimes it also includes measured forces at the end effector (see Cheah and Wang [3]).

Learning control has a history dating back to 1984 (see Arimoto et al [1]) when it was first applied to robot motion control. Horowitz [10] gives a nice history of the development and usage of learning controllers for (rigid) robot manipulators. He compares and contrasts different learning algorithms and also provides an experimental demonstration of a robot that learns to make its end effector track a circular trajectory. He insightfully points out that an open area of research is in finding methods for robust *optimal* (e.g., minimum energy, minimum vibration, or minimum time) trajectory learning, as opposed to only finding a control history that meets output requirements. Examples of work that have empirically investigated approaches to this problem include Gorinevsky ([7], [8], and [9]), who considered the use of the Levenberg-Marquardt optimization method for least squares, and Sadegh and Driessen [11] who

* Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, make any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

considered the use of gradient-based algorithms for constrained optimization.

A method for underdetermined learning control for nonlinear systems was presented in [Driessen, et al, 2000]. A rigorous proof of the convergence to zero of the output error under the associated mild assumptions was provided. However, no method for drift prevention was given. And, at least theoretically, the input vector could drift large distances from the initial vector. The present work presents an extension to that work but is presently restricted to linear or weakly nonlinear time-varying systems; for such systems, the method presented herein prevents drift by imposing additional model-based sub-optimality constraints based upon a fictitious cost function. In the limiting case of no modeling error, the method produces a feasible input that minimizes the fictitious cost function. The method relies heavily upon the theory and methods in [Driessen, et al, 1998] and [Driessen, et al, 2000].

Herein we will present an overview and motivation in section 2 for the iterative learning control problem presented in section 3, present the method of solution in section 4, explain why the method is also applicable to weakly nonlinear systems in section 5, and give conclusions in section 6.

2. Applications/Overview of Learning Control

Learning control is a method of output tracking or point-to-point control that does not make any assumptions about whether the system is minimum-phase or non-minimum-phase nor any assumptions about knowledge of the system's order (number of state variables). The method uses the *actual* measured system's response to an input in a feedback loop, in order to obtain tracking robustness in the presence of large model mismatch.

Example applications include robotics and manufacturing where a certain output tracking task is to be performed repeatedly. An initial guess of the inputs to the system can be obtained off-line by using an approximate model of the system. Then, in the learning feedback loop, the actual output response errors are measured. The gradients of these output errors with respect to the inputs can be obtained from strictly the approximate model or can be improved in accuracy by calculating gradients based on the model but evaluated along the actual measured trajectory of the system (see Sadegh and Driessen [11]). The learning feedback loop is continued at least until the actual system's output error is zero.

However, the learning feedback loop is often continued after zero output error is achieved, in order to obtain output tracking robustness against slowly varying dynamics of the actual system. For example, the dynamic properties of the joints of a robot may be changing over time, and the learning feedback loop will maintain zero

output error in spite of these time-varying system properties.

Mathematically, the iterative learning control problem considered herein can be viewed as the Newton Raphson problem illustrated in below

$$\bar{U} \rightarrow (\text{System}) \rightarrow \bar{Y}$$

where \bar{U} denotes a vector of input values that define the input over the time interval of interest and \bar{Y} denotes an output vector which we desire to be zero. For example, \bar{U} may be parameters that define a spline of the input history over the interval. The inaccuracy of the Jacobian $J \equiv \partial Y / \partial U$ is a result of the fact that we never have a perfect model of the dynamic system.

Remark 2.1: The reader may be wondering why we do not just calculate $J(U)$ by numerically differentiating the *System*. The reason is as follows. Let N be the number of time steps in a digitally controlled system; then, the number of variables (length of the vector U) is proportional to N . Such numerical differentiation would take $\text{Order}(N^2)$ time. The number of variables can be on the order of 100's to 10000's. Thus, the numerical differentiation would not be practical. Banded matrix methods that use the model of the system, on the other hand, allow the user to calculate an approximate Newton Raphson search direction p_k in $\text{Order}(N)$ time. Thus, while the model-based calculation of the search direction is practical, the numerical differentiation approach is not.

Consider a problem of underdetermined learning control of a linear time-varying discrete-time system. Because the number of output requirements (constraints) is less than the number of inputs, iterative learning control (ILC) (see, for example [Driessen, et al, 2000] with model-based (inexact) Jacobians could, at least theoretically, lead to very large drifts of the input vector from its starting value.

This is simply due to the non uniqueness of the solution vector. A method for preventing such drift would be desirable. Moreover, it would be nice if the method converged to a truly optimal solution (e.g., minimum energy solution) in the limit as the modeling error (Jacobian error) goes to zero. The method should be implementable in $O(N)$ storage and $O(N)$ time, where N is the number of sampling periods or time steps. In the next section, section 3, we will present a precisely-stated problem statement whose solution, in section 4, satisfies the requirements stated above.

3. Problem Statement

We are given a linear underdetermined input-output relationship:

$$f = Jx - b \tag{3.1}$$

where $J \in R^{m \times n}$, $m \leq n$, and J has full rank. However, J is not known. Only an estimate, $\hat{J} \neq J$, of J is available, where \hat{J} also has full rank. The value of f in (3.1) is

RECEIVED

NOV 15 2000

OSTI

available, by measurement, at any value of x . The objective is to find an algorithm, which we will denote by M , that uses f and \hat{J} to drive f to zero while preventing drift in the vector x , i.e.,

$$\begin{array}{l} \hat{J} \rightarrow \\ f \rightarrow \end{array} \begin{array}{|c|} \hline M \\ \hline \end{array} \begin{array}{l} \rightarrow \lim_{k \rightarrow \infty} f_k = 0 \\ \rightarrow \text{bounded drift of } x \end{array} \quad (3.2)$$

The algorithm M must be implementable in $O(N)$ storage and $O(N)$ time for linear time-varying discrete-time (LTVD) dynamic systems, e.g., using banded matrix methods (see [Driessen, et al, 1999] and [Wright, 1991]). And, M must require only mildly stringent conditions on the modeling error $\hat{J} - J$.

4. Method of Solution

Let us propose a fictitious cost function q :

$$q \equiv \frac{1}{2} x^T Q x + c^T x \quad (4.1)$$

where $Q > 0$. Because J is not known, we cannot find a truly optimal solution to the problem of minimizing q subject to (3.1), at least not in $O(N)$ time for the case of LTVD problems. However, if we did know J , the conditions for optimality:

$$\nabla q + J^T \lambda = 0 \quad (4.2)$$

or,

$$Qx + c + J^T \lambda = 0 \quad (4.3)$$

and

$$f = Jx - b = 0 \quad (4.4)$$

could be solved for x and λ . The next best approach seems to be to find an x and λ that satisfy sub-optimality conditions:

$$Qx + c + \hat{J}^T \lambda = 0 \quad (4.5)$$

and

$$f = Jx - b = 0 \quad (4.6)$$

Since J is not available, we must use \hat{J} as the Jacobian of (4.6) in an iterative approach to solve (4.5) and (4.6). The model-based Jacobian of (4.5)/(4.6) is:

$$\hat{A} = \begin{bmatrix} Q & \hat{J}^T \\ \hat{J} & 0 \end{bmatrix} \quad (4.7)$$

and we have a measurement of the error in (4.5)/(4.6), i.e.,

$$e = \begin{pmatrix} Qx + c + \hat{J}^T \lambda \\ f \end{pmatrix} \quad (4.8)$$

The true Jacobian of (4.5)/(4.6) is:

$$A = \begin{bmatrix} Q & J^T \\ J & 0 \end{bmatrix} \quad (4.9)$$

Using the theory and methodology presented in [Driessen, et al, 1998] and [Driessen, et al, 2000] for nonlinear systems, we can iteratively find a solution to (4.5)/(4.6) if

A and \hat{A} satisfy the following conditions. There must exist a $\gamma > 0$ such that

$$\min(\text{eigenvalues}(\hat{A}^T \hat{A})) \geq \gamma > 0 \quad (4.10)$$

and there must exist a $\delta > 0$ such that

$$\min(\text{eigenvalues}(A \hat{A}^{-1} / 2 + \hat{A}^{-T} A^T / 2)) \geq \delta > 0 \quad (4.11)$$

Condition (4.10) is satisfied since $Q > 0$ and \hat{J} was stated to have full rank in the problem statement of section 3. We notice that condition (4.11) is satisfied with $\delta = 1.0$ if $\hat{A} = A$, i.e., if $\hat{J} = J$. Therefore, by the continuity of the eigenvalues' dependence upon $A \hat{A}^{-1}$, we see that (4.11) is a fairly mild requirement on the accuracy of \hat{J} . If \hat{J} is accurate enough that (4.11) is satisfied, then the algorithm of [Driessen et al, 1998 and 2000] is guaranteed to cause

$$\lim_{k \rightarrow \infty} e_k = 0 \quad (4.12)$$

so that

$$\lim_{k \rightarrow \infty} f_k = 0 \quad (4.13)$$

as required in the problem statement of section 3. Also, (4.12) implies that

$$\lim_{k \rightarrow \infty} (Qx_k + c + \hat{J}^T \lambda_k) = 0 \quad (4.14)$$

The solution pair (x, λ) to the equations (4.5)/(4.6) is unique due to the nonsingularity of these equations' Jacobian, A , in (4.9). Therefore, the sequence of iterates x_k cannot drift large distances from the starting point. Moreover, as $\hat{J} \rightarrow J$, the solution to (4.5)/(4.6) approaches the solution to (4.3)/(4.4), which implies that as $\hat{J} \rightarrow J$, the converged solution x approaches the unique optima of the quadratic program:

$$\text{Min} \frac{1}{2} x^T Q x + c^T x \quad (4.15)$$

subject to

$$f = Jx - b = 0 \quad (4.16)$$

Thus, if $\hat{J} \approx J$ and we choose $Q = I$ and $c = 0$, for example, then the sequence x_k converges to a feasible solution (satisfying (4.16)) that is nearly a minimum-energy solution.

In summary, the solution algorithm M in (3.2) is the application of the algorithm in [Driessen, et al, 1998] and [Driessen, et al, 2000] to the fictitious-cost base system of equations (4.5)/(4.6), in which the search variables are x and λ . Extension to the case of weakly nonlinear systems is given in the next section.

5. Extension to Weakly Nonlinear Systems

The method of section 4 is also applicable to nonlinear systems so long as the nonlinearity is small enough. In particular, for a nonlinear system, J and \hat{J} become dependent upon x , i.e., $J = \frac{\partial f}{\partial x}$. So, \hat{A} and A become dependent upon x . The true Jacobian A of (4.5)/(4.6) becomes

$$A = \begin{bmatrix} H & \hat{J}^T(x) \\ J(x) & 0 \end{bmatrix} \quad (5.1)$$

where H is non constant since \hat{J} depends upon x . However, if the partial derivatives of \hat{J} with x are small enough, i.e., $H \approx Q$, then \hat{A} in (4.7) can still be used and conditions (4.10) and (4.11) are still satisfied (and must be for all iterates x_k to meet the hypotheses in [Driessen, et al, 1998 and 2000]); then, $e_k \rightarrow 0 \Rightarrow f_k \rightarrow 0$ is still assured. As with the linear case, as the modeling error approaches zero, the converged solution approaches a local minima of the problem of minimizing q subject to the nonlinear constraint $f(x)=0$, since $e=0$ (in (4.8)) implies the pair (x, λ) is a Kuhn Tucker point (see [Bazaraa]).

6. Conclusion

This work presented a method for preventing drift in iterative learning control of underdetermined linear and weakly nonlinear time-varying systems. Through the insertion of a fictitious cost function and the associated model-based sub-optimality conditions, the problem is converted to one with a unique solution. Under the stated mild conditions on the modeling error, the method converges to an input with zero output error. And, if the modeling error were to approach zero, the point of convergence would approach the true optima of the fictitious cost subject to zero output error, in the linear case, and true local minima in the (weakly) nonlinear case.

References

- [1] Arimoto, S., et al, "Bettering Operation of Robots by Learning," *Journal of Robotic Systems*, Vol. 1, No. 2, 1984, pp. 123-140.
- [2] Bazaraa, M., et al, *Nonlinear Programming, Theory and Applications, Second Edition*, New York: John Wiley and Sons, Inc., 1979.
- [3] Cheah, C. and Wang, D., "Learning Control for a Class of Nonlinear Differential-Algebraic Systems With Application to Constrained Robots," *Journal of Robotic Systems*, Vol. 13, No. 3, 1996, pp. 141-151.
- [4] Driessen, B., Sadegh, N., and Kwok, K., "A Robust Line Search for Learning Control," Conference on Decision and Control, 1998.
- [5] Driessen, B., Sadegh, N., and Kwok, K., "A Robust Line Search for Learning Control of Underdetermined Systems," Conference on Control Applications, 2000.
- [6] Driessen, B., Sadegh, N., Parker, G., and Eisler, G., "A Fast and Robust Algorithm for General Inequality/Equality Constrained Minimum Time Problems," *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 121, September 1999, pp. 337-345.

- [7] Gorinevsky, D., "An Algorithm for On-Line Parametric Nonlinear Least Square Optimization," 33rd IEEE Conference on Decision and Control, Lake Buena Vista, Florida, December, 1994.
- [8] Gorinevsky, D., "An Application of On-Line Parametric Optimization to Task-Level Learning Control," American Control Conference, Seattle, Washington, June 1995, pp. 862-866.
- [9] Gorinevsky, G. et al, "Learning Approximation of Feedforward Dependence on the Task Parameters: Experiments in Direct-Drive Manipulator Tracking," American Control Conference, Seattle Washington, 1995, pp. 883-887.
- [10] Horowitz, Roberto, "Learning Control of Robot Manipulators," *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 115, June 1993, pp. 402-411.
- [11] Sadegh, N. and Driessen, B., "Minimum Time Trajectory Learning," *ASME Journal of Dynamic Systems, Measurement and Control*, Vol. 121, June 1999, pp. 213-217.
- [12] Strang, G., *Linear Algebra and Its Applications*, 3rd ed., Academic Press, New York, 1988.
- [13] Wright, S., "Structured Interior Point Methods for Optimal Control," Conference on Decision and Control, 1991, pp. 1711-1716.