A Robust Line Search for Learning Control*

Brian J. Driessen
Sandia National Laboratories, Albuquerque, NM 87185-0439, bjdries@sandia.gov
Nader Sadegh
Georgia Institute of Technology, Atlanta, GA 30332, nader.sadegh@me.gatech.edu
Kwan S. Kwok
Sandia National Laboratories, Albuquerque, NM 87185-1003, kskwok@sandia.gov

Abstract.

In this paper a new line search for a Newton Rhapson learning control algorithm is presented. Theorems and rigorous proofs of its increased robustness over existing line searches are provided, and numerical examples are used to further validate the theorems. Also, the previously posed open question of whether robust optimal trajectory learning is possible is also addressed. It is shown that the answer is generally no, at least for gradient-based learning control algorithms.

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 [7] 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. In Section 5 of the present paper, the question of whether this is possible will be addressed. Examples of work that have empirically investigated approaches to this problem include Gorinevsky ([4], [5], and [6]), who considered the use of the Levenberg-Marquardt optimization method for least squares, and Sadegh and Driessen [8] who considered the use of gradient-based algorithms for constrained optimization.

Herein we will present a new line search for a Newton Rhapson learning control algorithm in Section 2, discuss applications in Section 3, demonstrate its advantages over existing line searches on a numerical example in Section 4, discuss whether robust optimal trajectory learning is possible in Section 5, and finally end with conclusions in Section 6.

2. On the Robustness Against Erroneous Constraint Jacobean of a New Line Search for a Newton Rhapson Learning Control Algorithm

This section will prove that the modified-line-search Newton Rhapson algorithm, to be presented shortly, is robust to erroneous constraint gradients. We will show that, under a very weak requirement on the accuracy of the constraint Jacobean, the Newton Rhapson algorithm, with the newly proposed line search, is guaranteed to create a sequence of iterates whose constraint error goes to zero. Then we will show that the standard existing line search fails under the same requirements.

* 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, makes 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.

Theorem 2.1: Consider the problem to find a solution $\bar{x}$, $\bar{x} \in R^n$, that satisfies equation (2.1) below

$$f(x) = 0 \tag{2.1}$$

where *f* maps from R^n to R^n. Let *J*(*x*) denote the *n* by *n* Jacobean of *f* at *x*:

$$J(x) \equiv \frac{\partial f(x)}{\partial x} \tag{2.2}$$

Suppose, that for any *x*, the value of *f*(*x*) is available, but we have only an inaccurate estimate $\hat{J}(x)$ of the true Jacobean $J(x)$. Suppose that the merit function for the algorithm is *z*(*x*) given by equation (2.3) below

$$z(x) \equiv \frac{1}{2} f^T(x) f(x) \tag{2.3}$$

which is half the sum of the squares of the constraint violations. Let $z_x(x) \equiv \frac{\partial z(x)}{\partial x}$ $= J^T(x) f(x)$. Suppose that for all iterates x_k and x_{k+1} of the algorithm, the Hessian of *z* is norm-bounded so that there exists a *G*>0 such that

$$z(x_{k+1}) - z(x_k) \le$$
$$f^T(x_k) J(x_k)(x_{k+1} - x_k) + G\|x_{k+1} - x_k\|^2$$
$$\forall x_k, x_{k+1} \tag{2.4}$$

Suppose that for all iterates x_k, there exist a $\gamma > 0$ for which

$$\min(eigenvalues(\hat{J}^T \hat{J})) \ge \gamma > 0 \tag{2.5}$$

and that for all iterates x_k there exists a $\delta > 0$ such that

$$\min(eigenvalues(J\hat{J}^{-1}/2 +$$
$$\hat{J}^{-T} J^T / 2)) \ge \delta > 0 \tag{2.6}$$

Suppose that for every iterate x_k, the search direction p_k is calculated by:

$$p_k = -\hat{J}^{-1}(x_k) f(x_k) \tag{2.7}$$

and that x_{k+1} is chosen by:

$$x_{k+1} = \left(\frac{1}{2}\right)^m p_k + x_k \tag{2.8}$$

where *m* is the smallest nonnegative integer such that

$$z\left(x_k - \left(\frac{1}{2}\right)^m \hat{J}^{-1} f(x_k)\right) -$$
$$z(x_k) \le -\left(\frac{1}{2}\right)^{2m} f^T(x_k) f(x_k) \tag{2.9}$$

Then, the sequence $z(x_k) \to 0$ as $k \to \infty$, implying $f(x_k) \to 0$.

Proof of Theorem 2.1: We will prove that if $z(x_k)$ is lower bounded away from zero, i.e., $z(x_k) \ge \eta > 0 \ \forall k$, then $|z(x_{k+1}) - z(x_k)|$ is lower bounded away from zero. So, suppose $z(x_k) \ge \eta > 0 \ \forall k$. From (2.7) and (2.8), we have

$$x_{k+1} - x_k = -\left(\frac{1}{2}\right)^m \hat{J}^{-1}(x_k) f(x_k) \tag{2.10}$$

Substituting (2.10) into (2.4) gives

$$z\left(x_k - \left(\frac{1}{2}\right)^m \hat{J}^{-1}(x_k) f(x_k)\right) - z(x_k) \le$$
$$-\left(\frac{1}{2}\right)^m f^T(x_k) J(x_k) \hat{J}^{-1}(x_k) f(x_k) +$$
$$G\left(\frac{1}{2}\right)^{2m} f^T(x_k) \hat{J}^{-T}(x_k) \hat{J}^{-1}(x_k) f(x_k)$$
$$\forall x_k, x_{k+1} \tag{2.11}$$

If we choose *m* so that the right hand side of (2.11) is ≤ the right hand side of (2.9), then clearly this is sufficient for such an *m* to satisfy the line search (2.9). So, let us write out this sufficient condition. (Note: for conciseness we will omit the arguments x_k (e.g., $f(x_k)$ will be written f and $J(x_k)$ will be written J)). After a tiny amount of algebra, we have

$$m \ge fix(\overline{m}) + 1 \tag{2.12}$$

where $\overline{m}$ is given in (2.13) below and fix(*) is the smallest nonnegative integer≥ *.

$$\overline{m} \equiv \ln\left(\frac{f^T(G\hat{J}^{-T}\hat{J}^{-1} + I)f}{f^T J\hat{J}^{-1} f}\right) / \ln(2) \tag{2.13}$$

By hypothesis (2.5), the largest possible eigenvalue of $G\hat{J}^{-T}\hat{J}^{-1} + I$ is $G/\gamma + 1$ so that $f^T(G\hat{J}^{-T}\hat{J}^{-1} + I)f \le (G/\gamma + 1) f^T f$; and, by hypothesis (2.6), $f^T J\hat{J}^{-1} f \le \delta f^T f$. Therefore, $\overline{m} \le m_{\max}$ where $m_{\max}$ is given below

$$m_{\max} \equiv \ln((G/\gamma + 1)/\delta)/\ln(2) \tag{2.14}$$

If we substitute the associated value of *m*, ($m = m^* \equiv fix(m_{\max}) + 1$), into (2.9) and the current assumption that $z(x_k) \ge \eta > 0 \ \forall k$, we obtain

$$z(x_{k+1}) - z(x_k) \le -\left(\frac{1}{2}\right)^{2m^*} f^T(x_k) f(x_k) =$$
$$-(2)\left(\frac{1}{2}\right)^{2m^*} z(x_k) \le -(2)\left(\frac{1}{2}\right)^{2m^*} \eta \tag{2.15}$$

Thus, $z(x_k) \ge \eta > 0$ implies

$$z(x_{k+1})\text{-}z(x_k) \le -(2)\left(\frac{1}{2}\right)^{2m^*} \eta \tag{2.16}$$

Now, since the sequence $z(x_k)$ is monotonically decreasing and $z(x_k) \ge 0$, the sequence $z(x_k)$ must converge. We now prove that there cannot exist a $\mu > 0$ such that $z(x_k) \geq \mu \forall k$. If there did exist such a μ, then $z(x_{k+1}) - z(x_k) \leq -(2)\left(\frac{1}{2}\right)^{2m^*} \mu$ for all k, implying that $z(x_k)$ decreases indefinitely, which is impossible. Therefore, no such μ exists and $z(x_k) \to 0$ as $k \to \infty$. This completes the proof of Theorem 2.1. QED.

Theorem 2.2: Consider the hypotheses of Theorem 2.1, but with the following standard Armijo line search (see Bazarra [2], page 307) replacing the line search (2.9):

$$z\left(x_{k+1} - \left(\frac{1}{2}\right)^m \hat{J}^{-1}(x_k) f(x_k)\right) - z(x_k) \leq$$
$$-\alpha \left(\frac{1}{2}\right)^m f^T(x_k) f(x_k) \tag{2.17}$$

with $\alpha \in (0,1)$. Then, the Newton Rhaphson algorithm can fail to cause $z(x_k)$ to converge to zero.

Proof of Theorem 2.2: Again, choosing m so that the right hand side of (2.11) is $\leq$ the right hand side of (2.17) would be sufficient to guarantee that line search (2.17) is satisfied. Writing this sufficient condition gives, after a tiny amount of algebra

$$2^m (f^T J \hat{J}^{-1} f - \alpha f^T f) \geq G f^T \hat{J}^{-T} \hat{J}^{-1} f \tag{2.18}$$

We see that if α is chosen too large (i.e., $\alpha \geq \delta$), then the left hand side of (2.18) could be negative. Then, even if G=0, there would not exist a nonnegative integer m satisfying (2.18). Thus, the line search could fail. This completes the proof of Theorem 2.2. QED.

Remark 2.1: The advantages of line search (2.9) over the standard Armijo line search (2.17) are twofold. First, since δ is not known, one could easily choose $\alpha \geq \delta$ by accident, thus causing the standard Armijo line search to fail while line search (2.9) would not fail. Second, even if a conservative value of δ were known, this value is surely going to be over conservative, leading to the use of a much smaller value of α than would actually be required for convergence to occur. Such an over conservative Armijo parameter α could lead to unnecessarily slow convergence of the algorithm.

Remark 2.2: The reader may be wondering why we do not just calculate *J*(*x*) by numerically differentiating *f*(*x*). 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 *x*) is proportional to *N*. Such numerical differentiation would take Order(N^2) time. The number of variables is 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 Rhaphson search direction p_k in Order(*N*) time. Thus, while the model-based calculation of the search direction is practical, the numerical differentiation approach is not.

3. Overview of Learning Control

Learning control is a method of output tracking 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 [8]). 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 learning control problem considered herein can be viewed as the Newton Rhaphson problem illustrated in (3.1) below

$$\bar{U} \to (System) \to \bar{Y} \qquad (3.1)$$

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 Jacobean $J \equiv \partial Y / \partial U$ is a result of the fact that we never have a perfect model of the dynamic system.

4. Numerical Example

In this section we demonstrate the robustness advantage of the proposed line search over an existing Armijo line search. The proposed line search yields convergence of the output error to zero while the existing Armijo line search fails to do so. This numerically supports Theorems 2.1 and 2.2.

The dynamic system considered is a two-mass/two-spring system. The springs are nonlinear. The spring forces are given by

$$F_s = -k\delta - a\delta^3 \qquad (4.1)$$

where δ is the deflection of the spring, F_s the spring force, and k and a constants. If we let q_1 and q_2 denote the positions of the masses, measured from equilibrium positions, and relative to an inertial reference frame, and let each mass value be m and let both springs be equivalent and let there be control forces u_1 and u_2 on the first two masses, then the equations of motion are

$$m\ddot{q}_1 = k(q_2 - q_1) + a(q_2 - q_1)^3 - kq_1 - aq_1{}^3 + u_1 \qquad (4.2)$$

$$m\ddot{q}_2 = -k(q_2 - q_1) - a(q_2 - q_1)^3 + u_2 \qquad (4.3)$$

The inputs $u_1(t)$ and $u_2(t)$ will be approximated by 2-node linear splines. Thus, $\bar{U}$ in (3.1) is a 4-vector, i.e., $\bar{U}^T$=($u_1(0)$, $u_1(t_f)$, $u_2(0)$, $u_2(t_f)$). The output $\bar{Y}$ is the difference between the state $\bar{x} \equiv (q_1, q_2, q_3, \dot{q}_1, \dot{q}_2, \dot{q}_3)^T$ at the final time $t = t_f$ and the desired value of $\bar{x}(t_f)$, denoted by $(q_{1d}, q_{2d}, q_{3d}, 0, 0, 0)^T$, where q_{1d}=0.1 and q_{2d}=0.1. The true values of *m*, *k*, and *a* are 1.0, 0.1, and 0.001. The model values are 1.5, 1.42, and 0.001, (a percentage error of 50% in *m* and 1300% in *k*). A final time of t_f=2.0 seconds and 100 Euler integration steps were used. The initial guess was $\bar{U}^T$=(1.3179, -1.6680, -0.5748, 0.7917).

Using the proposed line search, the solutions for $u_1(t)$ and $u_2(t)$ were *successfully* found. On the other hand, the existing Armijo line search locked up and *failed* to converge to the solution. The reason was precisely the one stated in the proof of Theorem 2.2. Namely, $f^T J\hat{J}^{-1} f - \alpha f^T f$ (see equation (2.18)) was less than zero so that no integer *m* existed that would satisfy the standard Armijo line search criterion (2.17). Yet, all the requirements for convergence with the newly proposed line search were satisfied so that it yielded successful convergence.

5. Can Robust Optimal Trajectory Learning Be Accomplished?

We note that the title of this section is essentially the research question posed by Horowitz [7]. The discussion of this section will point out that in general the answer to the posed question is no, at least for gradient-based learning schemes. We will see that for such schemes, convergence to a truly optimal trajectory cannot be guaranteed. The reason is very simple. If we have an approximate (erroneous) gradient, denoted in this section by $\hat{\nabla} z$, which is "close" to the true gradient, denoted in this section by ∇z, then the search direction is *null* if $\hat{\nabla} z = 0$ (even if $\nabla z \neq 0$). Thus, while convergence of $\hat{\nabla} z$ to zero may be provable under mild assumptions, the convergence of the true gradient ∇z to zero clearly cannot be guaranteed. In summary, unlike robust learning control for output tracking, gradient-based optimal trajectory learning schemes cannot be guaranteed to converge to a stationary point.

It is worthwhile to point out that similar problems hold for the case when the cost function *z* is a sum of the squares of measurable outputs for the case of more outputs than inputs. Namely, the same problem of obtaining a null search direction at point(s) where the true gradient is not zero occurs in such over-determined least squares problems where the erroneous gradient is given by $\hat{J}^T f$ where *f* is the (measured) error vector and $\hat{J}$ its approximate (erroneous) Jacobean with respect to the input vector. Of course the true gradient is $J^T f$ where J is the exact Jacobean. With such least squares problems J^T has fewer rows than columns and thus has a null space of non-zero dimension, and a tiny difference between $\hat{J}$ and

J can render f in the null space of $\hat{J}$ even though f is *not* in the null space of J, thus, again, rendering $\hat{\nabla}z$ zero even though $\nabla z \neq 0$.

6. Conclusion

This work presented a new line search for a Newton Rhaphson based robust learning control method, where output measurements are available but the system model is known only approximately. The mild conditions that guarantee the convergence of the proposed method were given and the associated convergence proof provided. Existing Armijo line searches were proven to be susceptible to locking up under these mild conditions, which reveals that the newly proposed line search is more robust. A numerical example was presented that demonstrated these facts.

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. Bazarra, 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. Gorinevsky, D. 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.
5. Gorinevsky, D., "An Algorithm for On-Line Parametric Nonlinear Least Square Optimization," *33rd IEEE CDC*, Lake Buena Vista, Florida, December, 1994.
6. Gorinevsky, D., "An Application of On-Line Parametric Optimization to Task-Level Learning Control," American Control Conference, Seattle, Washington, June 1995, pp. 862-866.
7. Horowitz, Roberto, "Learning Control of Robot Manipulators," *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 115, June 1993, pp. 402-411.
8. Sadegh, N. and Driessen B., "Minimum Time Trajectory Learning," American Control Conference, Seattle, Washington, June, 1995.