Optimal Trajectory Design for Well-Conditioned Parameter Estimation

Andrew D. Wilson and Todd D. Murphey

Abstract— **When attempting to estimate parameters in a dynamical system, it is often beneficial to systematically design the experimental trajectory. This paper presents a method of generating trajectories using an extension of a nonlinear, infinite-dimensional, projection-based trajectory optimization algorithm. A reformulated objective function is derived for the algorithm to minimize the condition number of the Hessian of the batch-least squares identification method. The batch least-squares method is then used to estimate parameters of the nonlinear system. A simulation example is used to demonstrate that an arbitrarily designed trajectory can lead to an ill-conditioned Hessian matrix in the batch-least squares method, which in turn leads to a less precise set of identified parameters. An example using Monte-Carlo simulations of both trajectories shows a reduction in the variance of identified parameters for an example cart-pendulum system.**

I. Introduction

A widely used method to identify unknown or inexact parameters in a nonlinear dynamical system is batch least-squares parameter estimation [1]. Given a set of control inputs, the system's trajectory is experimentally measured, and the measured trajectory is compared to the expected trajectory in a least-squares sense. This comparison can be performed using the batch least-squares method where a set of parameters is found that minimizes the least-squares difference between the measured trajectory and the model. In a perfectly modeled system with zero process and measurement noise, the set of estimated parameters converges to the exact model parameters given an initial guess in the neighborhood of the actual parameter set.

One major choice in designing an experiment to obtain accurate estimates of the parameter values is the set of control inputs that will drive the experimental system. A change in the control inputs can have a significant impact on the ability to estimate the model parameters. In the extreme case, a trajectory could be chosen that is orthogonal to the sensitivity of a given parameter, thus making any estimate of that parameter impossible.

A rich and vast amount of work has been performed in the area of experiment design, input design, and identifiability of parameters [2-6]. Highlights of a relevant subset of this work include work by Armstrong on optimal "exciting" trajectories [7] with additional methods proposed by Gautier and Khalil [8]. These optimization methods synthesize trajectories for systems that have nonlinear dynamics with respect to the state but must be linear with respect to the parameters. Related work by Swevers also examines parameter estimation trajectories for the same class of systems but in the sense of the Fisher information matrix [9]. In both cases, the analysis is performed on a discrete number of measurements taken of the trajectory, and a finite discretization of the dynamics is used with time-discretized inputs. Many other methods in the area of experiment design also rely on discrete optimizations which tend to discretize the continuous dynamics [10], [11].

Additional studies have been performed in areas such as computational biology [12], [13] and aerospace [14], with an increasing emphasis being placed on trajectory generation using a variety of basis functions and other splines [15-18]. This method allows one to optimize over a cost, such as the condition number, using a finite dimensional optimization method over a fixed set of basis functions. While this allows the trajectory to be fully continuous, instead of discretized in time, the space of allowable control signals is still finite.

This paper contributes a new method of performing input optimization with respect to the conditioning of least-squares parameter estimation problems. Using an infinite-dimensional, projection-based optimization method originally designed for Bolza-type optimal trajectory tracking problems [19], the primary contribution of the paper is the reformulation of the cost function in a manner that could employ a non-Bolza cost function. This cost function is directly dependent on eigenvalues of the Hessian matrix of the parameter estimation method.

Additionally, the reformulation of the cost function requires additional states to be added to the trajectory optimization problem which are related to sensitivities of the trajectory with respect to the system parameters that will be identified. By using this optimization method, the continuous-time dynamics are not discretized; instead, variational curves on the input and trajectory are constructed locally and projected onto the trajectory manifold.

This paper is organized as follows: Section II covers the least-squares parameter optimization method; Section III presents the reformulated cost function and derives the equations necessary to perform an optimization over the condition number; and Section IV presents an example of condition number optimization for a cart-pendulum system.

II. Nonlinear Least-Squares Estimation

For general, nonlinear systems, the batch least-squares method of parameter optimization can be used [20], [21]. A

This material is based upon work supported by the United States Department of Defense through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program and by the National Science Foundation under Grant IIS 1018167. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

A. Wilson and T. Murphey are with the Department of Mechanical Engineering, Northwestern University, 2145 Sheridan Road, Evanston, IL 60208, USA. awilson@u.northwestern.edu, and t-murphey@northwestern.edu

brief description of the procedure and equations is included for completeness and with compatible notation. We begin by defining the nonlinear system model. Given a general, nonlinear system, the dynamics can be written as

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t), \mathbf{p}) \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^n$ defines the system states, $\mathbf{u} \in \mathbb{R}^r$ represents the control inputs, and $\mathbf{p} \in \mathbb{R}^h$ are the static parameters. For readability, the time arguments will be dropped from $\mathbf{x}$ and $\mathbf{u}$, though the input and trajectory remain time-varying.

The batch least-squares optimization is performed using a Newton-Raphson iterator. The cost function for the nonlinear least-squares method to be minimized is given by

$$J_p(\mathbf{x}, \mathbf{u}, \mathbf{p}) = \int_{t_0}^{t_f} \ell_p(\mathbf{x}, \hat{\mathbf{x}})\, dt \tag{2}$$

where

$$\ell_p(\mathbf{x}, \hat{\mathbf{x}}) = (\mathbf{x} - \hat{\mathbf{x}})^T \cdot Q_p \cdot (\mathbf{x} - \hat{\mathbf{x}})$$

and $\hat{\mathbf{x}}$ denotes the measured trajectory. The matrix Q_p is an $n \times n$ weighting matrix that can be designed according to the relative importance of tracking particular states. Although not covered in this paper, a proper noise analysis could influence the weighting of Q_p, which is analogous to work estimating the covariances in continuous time filtering [22].

The minimization problem is formulated as an unconstrained optimization since no bounds are placed on the values of the parameters. To use Newton's method for optimization, the first and second derivatives of J_p are needed w.r.t. $\mathbf{p}$.

A. First Derivative

The first derivative can be found by differentiating (2) w.r.t. $\mathbf{p}$. This results in

$$D_p J_p(\mathbf{x}, \mathbf{u}, \mathbf{p}) = \int_{t_0}^{t_f} D_x \ell_p(\mathbf{x}, \hat{\mathbf{x}}) \cdot D_p \mathbf{x}(\mathbf{x}, \mathbf{u}, \mathbf{p})\, dt. \tag{3}$$

To calculate $D_p\mathbf{x}$, we define the trajectory, $\mathbf{x}$, in its integral form,

$$\mathbf{x} = \mathbf{x}_0 + \int_{t_0}^{t_f} \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{p})\, dt.$$

Differentiating w.r.t. $\mathbf{p}$ yields

$$D_p \mathbf{x}(\mathbf{x}, \mathbf{u}, \mathbf{p}) = \int_{t_0}^{t_f} D_x \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{p}) \cdot D_p \mathbf{x}(\mathbf{x}, \mathbf{u}, \mathbf{p}) + D_p \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{p})\, dt. \tag{4}$$

The above equation provides the solution for the sensitivity of J_p with respect to changes in parameters. Since the expression for $D_p\mathbf{x}$ contains the integral of $D_p\mathbf{x}$, by differentiating with respect to time, it can be written as an independent differential equation, defined as

$$\mathbf{g}(\mathbf{x}, \mathbf{u}, \mathbf{p}) = D_p \mathbf{x}(\mathbf{x}, \mathbf{u}, \mathbf{p})$$
$$\dot{\mathbf{g}}(\mathbf{x}, \mathbf{u}, \mathbf{p}) = D_x \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{p}) \cdot \mathbf{g}(\mathbf{x}, \mathbf{u}, \mathbf{p}) + D_p \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{p}) \tag{5}$$

with the initial condition at $t = 0$, $\mathbf{g}(\mathbf{x}, \mathbf{u}, \mathbf{p}) = \{0\}^{n \times h}$.

B. Second Derivative

The second derivative of the cost is calculated in a similar manner. To further simplify notation, arguments of previously defined functions will be left out and replaced by $(\cdot)$. Taking the derivative of (3) results in the following:

$$D_p^2 J_p(\mathbf{x}, \mathbf{u}, \mathbf{p}) = \int_{t_0}^{t_f} \left[D_x^2 \ell_p(\cdot) \cdot \mathbf{g}(\cdot) \right]^{T(2,1)} \cdot \mathbf{g}(\cdot) + D_x \ell_p(\cdot) \cdot D_p^2 \mathbf{x}(\mathbf{g}, \mathbf{x}, \mathbf{u}, \mathbf{p})\, dt. \tag{6}$$

The quantity, $D_p^2\mathbf{x}$ must now be calculated, which is obtained by differentiating (4) with respect to $\mathbf{p}$. We again define this equation as follows

$$\mathbf{h}(\mathbf{g}, \mathbf{x}, \mathbf{u}, \mathbf{p}) = D_p^2 \mathbf{x}(\mathbf{g}, \mathbf{x}, \mathbf{u}, \mathbf{p})$$

$$\dot{\mathbf{h}}(\cdot) = \left[\left[D_x^2 \mathbf{f}(\cdot) \cdot \mathbf{g}(\cdot) + D_p D_x \mathbf{f}(\cdot) \right]^{T(1,3,2)} \cdot \mathbf{g}(\cdot) \right]^{T(1,3,2)} + D_x \mathbf{f}(\cdot) \cdot \mathbf{h}(\cdot) + D_x D_p \mathbf{f}(\cdot) \cdot \mathbf{g}(\cdot) + D_p^2 \mathbf{f}(\cdot) \tag{7}$$

with the initial condition that at $t = 0$, $\mathbf{h}(\cdot) = \{0\}^{n \times m \times m}$.

Notational Note: Since we are dealing with tensor calculations, it will be necessary to transpose different levels of the tensors for inner product calculations. The notation used in this paper will be $A^{T(i,j,k)}$ where i, j, k denote the new positions of the old levels. For example, $A^{T(2,1)}$ is the standard transpose of a matrix. $A^{T(3,2,1)}$ transposes the third and first levels of a 3-tensor. This notation will allow for standard matrix product calculations for all of the equations presented.

III. Least-Squares Conditioning

Given an initial trajectory and estimate of the parameters, Newton's method can be used to find an optimal set of parameters based on the trajectory. With $\mathbf{u}$ and $\mathbf{p}$, we can calculate the condition number of the cost function Hessian. Assuming that the Euclidean norm is used and the Hessian is symmetric, the condition number is equal to

$$\kappa(D_p^2 J_p(\mathbf{x}, \mathbf{u}, \mathbf{p})) = \left| \frac{\lambda_{max}}{\lambda_{min}} \right|,$$

where λ is the set of eigenvalues of the matrix $D_p^2 J_p(\cdot)$.

A. Trajectory Optimization Problem

The next step is to optimize the system's trajectory using the condition number of the Hessian, $D_p^2 J_p(\cdot)$, as the objective function. The theoretical framework for the optimization routine is largely based on an infinite dimensional LQR optimization with a nonlinear projection step. Details on the theoretical background can be found in [19], [23].

We define a cost function that is based on the condition number of $D_p^2 J_p(\cdot)$, defined in terms of its eigenvalues. The cost function is given by

$$J_\tau = \frac{1}{2} \left(\frac{\lambda_{max}}{\lambda_{min}} \right)^2. \tag{8}$$

Although the cost function appears to include only a terminal condition involving the eigenvalues of $D_p^2 J_p(\cdot)$, the eigenvalues are actually functions of the trajectory, which will result in a running cost. This property will allow the non-Bolza form of the optimal control problem to be cast into a Bolza form for the iterative LQR step described in the next section.

The overall optimization problem can therefore be written as

$$\boxed{\begin{array}{c}\min\limits_{\mathbf{u}}\ J_\tau \\ \\ s.t.\ (1),(5),(7) \text{ are satisfied.}\end{array}}$$

B. Steepest Descent Calculation

The nonlinear constrained optimization is performed using an iterative gradient descent technique. The descent direction ζ is given by the following LQR problem,

$$\zeta = \arg\min DJ_\tau(\xi)\circ\zeta + \frac{1}{2}\langle\zeta,\zeta\rangle. \tag{9}$$

For the method used in this paper, we will treat the sensitivity, $\mathbf{g}$, and Hessian, $\mathbf{h}$, equations as dynamics constraints. Therefore, the dynamics of the system are expanded to include $\mathbf{g}$ and $\mathbf{h}$. We denote this expanded state as

$$\bar{\mathbf{x}} = \{\mathbf{x},\mathbf{g},\mathbf{h}\},$$

and the state and control vector are given by

$$\xi = \{\bar{\mathbf{x}},\mathbf{u}\}.$$

Since the minimization involves a derivative of the cost function, a linearization of the cost with respect to the variation on ξ is necessary. Therefore, $DJ_\tau(\xi)\circ\zeta$ is defined as the following

$$DJ_\tau(\xi)\circ\zeta = \int_{t_0}^{t_f} \mathbf{a}^T\mathbf{z}\,dt, \tag{10}$$

where $\mathbf{a}$ is the linearization of the cost functional in (8), and $\mathbf{z}$ is the perturbation on the extended state vector. This linearization will be derived in the following section. We then need to define a Hamiltonian to construct the optimality condition for the LQR problem. For the LQR problem (9), the Hamiltonian is given by the following

$$H = \frac{1}{2}\mathbf{z}^T Q_n\mathbf{z} + \frac{1}{2}\mathbf{v}^T R_n\mathbf{v} + \mathbf{a}^T\mathbf{z} + \mathbf{p}^T(A\mathbf{z}+B\mathbf{v}). \tag{11}$$

A and B are the linearizations of the dynamics and will be calculated in a later section. Q_n and R_n are weight matrices for the local LQR optimization, $\mathbf{p}$ is the co-state variable, and $\mathbf{v}$ is the control perturbation.

C. Eigenvalue Derivatives

In order to take the derivative of the cost function, it must be possible to take an analytic derivative of an eigenvalue. Fortunately, eigenvalue perturbation theory permits such differentiation, and the resulting form is relatively compact.

The derivative method we will use was derived by Nelson [24]. Given an eigensystem of the form,

$$AX = X\Lambda$$

where Λ is a diagonal matrix of eigenvalues, X is a matrix of eigenvectors, and, in this case, $A = D_p^2 J_p(\cdot)$. We will take this derivative with respect to the extended system states, $\bar{\mathbf{x}}$ as well as the controls, $\mathbf{u}$.

The result of Nelson's work is that the derivative of one eigenvalue of the matrix A is given by

$$D_i\lambda_k = y_k^T \cdot D_i A \cdot x_k, \tag{12}$$

where λ_k is the k^{th} eigenvalue of A, x_k is the associated left eigenvector, y_k is the associated right eigenvector of A, and $D_i A$ and $D_i\lambda_k$ are the partial derivatives of A and λ_k with respect to some argument i.

Given this derivative, it is possible to calculate a closed form of the derivative of the cost function Lagrangian as necessary for the optimization algorithm.

D. Lagrangian Linearization

The first linearization that is required is the linearization of the cost function Lagrangian with respect to the expanded state, $\bar{\mathbf{x}}$, and the control vector, $\mathbf{u}$, as shown in (10). In this section, we detail the equations for the derivatives of the cost function with respect to the expanded dynamics.

To begin, we use the quotient rule on the optimization cost given in (8). Taking the first derivative results in the following equation for $\mathbf{a}$:

$$\mathbf{a} = \frac{\partial J_\tau}{\partial\bar{\mathbf{x}}} = \frac{\left(\lambda_{min}\frac{\delta\lambda_{max}}{\delta\bar{\mathbf{x}}} - \lambda_{max}\frac{\delta\lambda_{min}}{\delta\bar{\mathbf{x}}}\right)}{\lambda_{min}^2}.$$

The equation for the eigenvalue derivative is now used as defined in the previous section. Given (12), the derivative of λ_i, for $i \in \{min, max\}$, w.r.t. $\bar{\mathbf{x}}$ is written as

$$\frac{\delta\lambda_i}{\delta\bar{\mathbf{x}}} = \int_{t_0}^{t_f} \mathbf{v}^{*T}\frac{\delta}{\delta\bar{\mathbf{x}}}\left(\frac{d}{dt}D_p^2 J_p(\cdot)\right)\mathbf{v}^*\,dt,$$

where $\mathbf{v}^*$ is the eigenvector associated with λ_i.

The inside of this integral will become the Lagrangian linearization but now includes the derivative of $D_p^2 J_p(\cdot)$ w.r.t. $\bar{\mathbf{x}}$. This new term can be solved analytically using the constraint equations and equation for the Hessian.

The derivative of $\frac{d}{dt}D_p^2 J_p$ is computed first with respect to each component of the expanded state. Then, the derivative of each component can be combined into a complete vector. First, we will solve for the derivative of the Hessian with respect to the nominal state, $\mathbf{x}$. The partial derivative of (6) w.r.t. $\mathbf{x}$ is calculated. This results in a tensor given by the following equation:

$$\frac{\delta}{\delta\mathbf{x}}\left(\frac{d}{dt}D_p^2 J_p(\cdot)\right) = \left[Q_p \cdot D_p^2\mathbf{x}(\cdot)\right]^{T(2,3,1)} = \left[Q_p\cdot\mathbf{h}(\cdot)\right]^{T(2,3,1)} \tag{13}$$

Next, we take the partial derivative w.r.t. $\mathbf{g}$. This results in a tensor given by the following

$$\frac{\delta}{\delta\mathbf{g}}\left(\frac{d}{dt}D_p^2 J_p(\cdot)\right) = [Q_p\cdot\mathbf{g}(\cdot)]^{T(2,1)}\cdot E + \left[(Q_p\cdot E)^{T(2,3,1)}\cdot\mathbf{g}(\cdot)\right]^{T(2,3,1)} \tag{14}$$

where E is a sparse tensor given by

$$E_{i,j,k,l} = \delta_{i,k}\delta_{j,l},$$

where $\delta_{\cdot,\cdot}$ is the Kronecker delta function.

The last partial derivative is of the Hessian with respect to the last expanded state, $\mathbf{h}$. The result is a tensor given by the following equation

$$\frac{\delta}{\delta\mathbf{h}}\left(\frac{d}{dt}D_p^2 J_p(\cdot)\right) = Q_p\cdot(\mathbf{x}(\cdot)-\hat{\mathbf{x}})\cdot F \tag{15}$$

where F is a sparse tensor given by the following

$$F_{i,j,k,l,m,n} = \delta_{i,l}\delta_{j,m}\delta_{k,n}.$$

We now have the Lagrangian cost function linearization with (13), (14), and (15). The next step is to calculate the linearization of the dynamic constraints w.r.t. $\bar{\mathbf{x}}$.

E. Dynamics Linearizations

To perform the LQR optimization, we need the linearization of the dynamic constraints. Due to the structure of the expanded dynamics, A simplifies to a lower triangular matrix of the form

$$A = \begin{bmatrix} \frac{\partial\dot{\mathbf{x}}}{\partial\mathbf{x}} & 0 & 0 \\ \frac{\partial\dot{\mathbf{g}}}{\partial\mathbf{x}} & \frac{\partial\dot{\mathbf{g}}}{\partial\mathbf{g}} & 0 \\ \frac{\partial\dot{\mathbf{h}}}{\partial\mathbf{x}} & \frac{\partial\dot{\mathbf{h}}}{\partial\mathbf{g}} & \frac{\partial\dot{\mathbf{h}}}{\partial\mathbf{h}} \end{bmatrix}.$$

The derivatives are straightforward using (1), (5), and (7). The first derivative of the state dynamics w.r.t. $\mathbf{x}$ is simply

$$\frac{\partial\dot{\mathbf{x}}}{\partial\mathbf{x}} = D_x f(\cdot).$$

This derivative depends on the system dynamics and must be calculated with respect to the dynamics. The next quantity is the partial derivative of $\mathbf{g}$ w.r.t. $\mathbf{x}$. This derivative is given by

$$\frac{\partial\dot{\mathbf{g}}}{\partial\mathbf{x}} = \left[D_x^2 f(\cdot)^{T(2,3,1)}\cdot\mathbf{g}(\cdot)\right]^{T(1,3,2)} + D_x D_p f(\mathbf{x}(\cdot)).$$

Next, the derivative of the dynamics of $\mathbf{g}$ with respect to that state yields

$$\frac{\partial\dot{\mathbf{g}}}{\partial\mathbf{g}} = D_x f(\cdot)\cdot E,$$

where E is the same tensor that is used in the Lagrangian linearizations.

The last row of the A matrix involves the derivative of $\mathbf{h}$ with respect to each element of the expanded state. The derivative w.r.t. $\mathbf{x}$ is given by

$$\begin{aligned}\frac{\partial\dot{\mathbf{h}}}{\partial\mathbf{x}} = &\left[\left(D_x^3 f(\cdot)^{T(1,4,2,3)}\cdot\mathbf{g}(\cdot)\right)^{T(1,4,2,3)}\cdot\mathbf{g}(\cdot)\right]^{T(1,2,4,3)} \\ &+\left[D_x D_p D_x f(\cdot)^{T(1,3,4,2)}\cdot\mathbf{g}(\cdot)\right]^{T(1,2,4,3)} \\ &+\left[D_x^2 f(\cdot)^{T(2,1,3)}\cdot\mathbf{h}(\cdot)\right]^{T(1,3,4,2)} + D_x D_p^2 f(\cdot).\end{aligned}$$

The derivative w.r.t. $\mathbf{g}$ is given by

$$\begin{aligned}\frac{\partial\dot{\mathbf{h}}}{\partial\mathbf{g}} = &\left[\left(D_x^2 f(\cdot)\cdot E\right)^{T(1,3,4,2)}\cdot\mathbf{g}(\cdot)\right]^{T(1,2,4,3)} \\ &+\left[D_x^2 f(\cdot)\cdot\mathbf{g}(\cdot)\right]^{T(1,3,2)}\cdot E,\end{aligned}$$

and the derivative w.r.t. $\mathbf{h}$ is given by

$$\frac{\partial\dot{\mathbf{h}}}{\partial\mathbf{h}} = D_x f(\cdot)\cdot F.$$

With these equations, the A matrix can be constructed. For both the Lagrangian and dynamics linearizations, the tensor forms for $\mathbf{a}$ and A can be mapped to a single vector form. The result is an A matrix that is $n(h^2+h+1)\times n(h^2+h+1)$.

Additionally, the LQR Hamiltonian from (11) requires the linearization of the dynamics with respect to the control inputs, $\mathbf{u}$. This linearization is given by the matrix B in the Hamiltonian. In the same way, the derivatives are calculated with respect to the control inputs resulting in B of the form

$$B = \begin{bmatrix} \frac{\partial\dot{\mathbf{x}}}{\partial\mathbf{u}} \\ \frac{\partial\dot{\mathbf{g}}}{\partial\mathbf{u}} \\ \frac{\partial\dot{\mathbf{h}}}{\partial\mathbf{u}} \end{bmatrix}.$$

The three derivatives are as follows

$$\frac{\partial\dot{\mathbf{x}}}{\partial\mathbf{u}} = D_u f(\cdot)$$

$$\frac{\partial\dot{\mathbf{g}}}{\partial\mathbf{u}} = \left[D_u D_x f(\cdot)^{T(1,3,2)}\cdot\mathbf{g}(\cdot)\right]^{T(1,3,2)} + D_u D_p f(\cdot)$$

$$\begin{aligned}\frac{\partial\dot{\mathbf{h}}}{\partial\mathbf{u}} = &\left[\left(D_u D_x^2 f(\cdot)^{T(1,4,2,3)}\cdot\mathbf{g}(\cdot)\right)^{T(1,4,2,3)}\cdot\mathbf{g}(\cdot)\right]^{T(1,2,4,3)} \\ &+\left[\left(D_u D_p D_x f(\cdot)^{T(1,3,2,4)}\right)^{T(1,2,4,3)}\cdot\mathbf{g}(\cdot)\right]^{T(1,2,4,3)} \\ &+\left[D_u D_x f(\cdot)^{T(1,3,2)}\cdot\mathbf{h}(\cdot)\right]^{T(1,3,4,2)} + D_u D_p^2 f(\cdot).\end{aligned}$$

Again, these tensors can be mapped to a vector form of the expanded state, resulting in a $n(h^2+h+1)\times r$ matrix which is then multiplied by the variation of the control input, $\mathbf{v}$ in the optimization calculation.

F. Optimization Routine

Given the linearization calculations, it is possible to perform the iterative LQR optimization with projection step. At each iteration, the Lagrangian and dynamics linearizations are calculated, providing $\mathbf{a}$, A, and B.

Following these linearization calculations, a Riccati equation is solved as part of the LQR solution. This LQR solution provides a control input, $\mathbf{v}$, based on the linearization of the system at a given point in time. However, this solution is only valid at the linearization. To find a feasible control, this perturbation is projected back onto the space of feasible trajectories and constraints.

For the trajectory optimization, the projection operator uses a stabilizing feedback control law to satistfy the constraint equations for $\mathbf{x}, \mathbf{g}$, and $\mathbf{h}$. The form of the projection operator is given by

$$P(\bar{\mathbf{x}}):\begin{cases}\mathbf{u}=\mu+K(\bar{\alpha}-\bar{\mathbf{x}})\\ \dot{\mathbf{x}}=\mathbf{f}(\mathbf{x},\mathbf{u})\\ \dot{\mathbf{g}}=\text{Eq.}(5)\\ \dot{\mathbf{h}}=\text{Eq.}(7)\end{cases}\tag{16}$$

where $(\bar{\alpha},\mu)$ is the infeasible trajectory from the steepest descent calculation, and K is the feedback gain which can be optimized by solving an additional LQR problem. Details on the optimal gain selection can be found in [19].

A few design choices are available in the optimization routine such as the ability to choose local weighting matrices and projection weightings. In the following cart-pendulum example, all weighting matrices are chosen as the identity matrix.

IV. Example

To illustrate the use of conditioning optimization, we will optimize a simple cart-pendulum system. We will assume that the pendulum has a single rotational degree of freedom, and the cart provides horizontal forces on the base of the pendulum as seen in Fig. 1. The dynamics for the system are given by the following

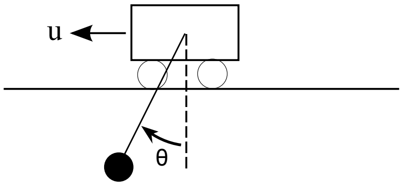


Fig. 1: Cart-pendulum system

$$\dot{\mathbf{x}}=\left[\begin{array}{c}\dot{\theta}\\ -\frac{u}{\alpha}\cos\theta-g\sin\theta-b\dot{\theta}\end{array}\right]$$

where u is the cart acceleration control input, b is the viscous damping coefficient, and α is an input scaling parameter.

A. Identification Procedure

To simulate an experimentally measured trajectory, we will generate a deterministic trajectory based on "actual" parameters which would be inherent to the system, but not known exactly if the trajectory was experimentally measured. We then use additive white noise to create a stochastic trajectory for this example. Given this uncertain trajectory, the procedure is as follows:

1) Perform the batch least-squares optimization using an initial guess of the parameters which yields the first estimate of the parameters.
2) Perform the conditioning optimization using the estimate of the parameters which yields a new set of control inputs for the system.

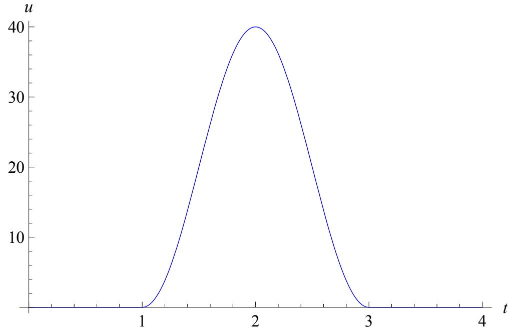


Fig. 2: Initial control input.

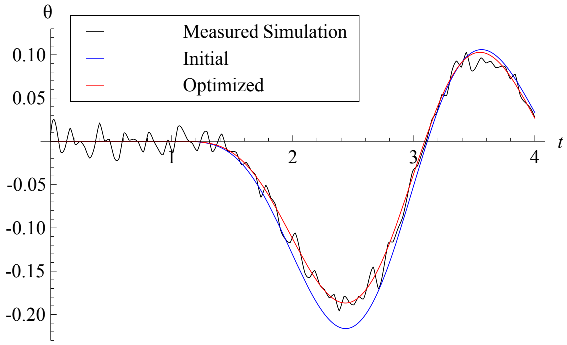


Fig. 3: System trajectories before and after parameter optimization simulated measurement noise.

3) Run the experiment using the newly generated inputs and measure the trajectory.
4) Perform a second parameter optimization using the new trajectory to determine a new estimate of the parameter set.

Note that the optimal trajectory generated by the conditioning optimization would be obtained using the actual model parameters; however, since in practice these parameters will not be known, the best alternative is to use the initial optimized set of parameters based on the chosen trajectory.

B. Parameter Optimization

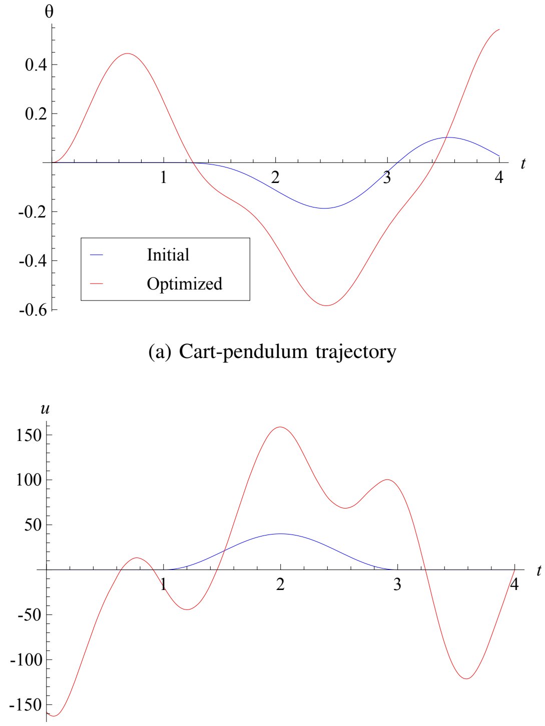
The two parameters that will be examined are the scaling parameter and damping coefficient. The parameters are as follows:

$$\mathit{Actual}:\alpha=30.0,\quad b=1.0\ \text{Ns/m}$$
$$\mathit{Initial\ Guess}:\alpha=25.1,\quad b=1.2\ \text{Ns/m}$$

The initial control input is arbitrarily chosen and shown in Fig. 2. The "measured" trajectory and optimized trajectory are shown in Fig. 3.

As a result of the parameter estimation, the parameters are identified as $b=0.983$ Ns/m and $\alpha=30.276$. The convergence criterion used for the parameter optimization was $|DJ_p(\cdot)|<10^{-8}$. The condition number of the Hessian of the estimation problem can also be calculated with these optimized parameters. For the initial trajectory, the Hessian is

$$D_p^2J_p=\left[\begin{array}{cc}0.034997 & 0.0022998\\ 0.0022998 & 0.00019817\end{array}\right].$$



(a) Cart-pendulum trajectory

(b) Input to cart-pendulum system

Fig. 4: Example trajectories and inputs before and after conditioning optimization.

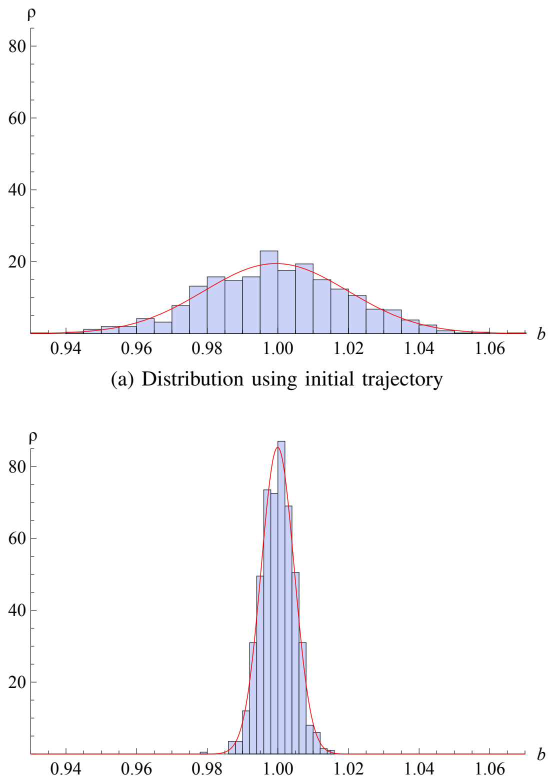
The condition number for this matrix can then be calculated and has a value of 629.1. The objective is to minimize the condition number as much as possible to improve the conditioning of the parameter identification optimization.

C. Condition Number Optimization

To begin the conditioning optimization, we take the control input, optimized parameter trajectory, and optimized set of parameters and then calculate the linearizations of the cost function Lagrangian and expanded state dynamics as detailed in Section III. The iterative optimization is performed, minimizing the LQR problem and projecting the perturbation into the feasible trajectory space.

The result is a new trajectory with a condition number of 23.8. The convergence criterion used for the condition number optimization is $|DJ_\tau(\xi)\circ\zeta| < 10^{-2}$. This criterion is set lower than the parameter optimization since only a first derivative method is used, leading to slower convergence rates. Plots of the initial and optimized trajectory are shown in Fig. 4. Following the conditioning optimization, the experiment would be run again to measure a new trajectory. For this simulation, we will again add noise to a deterministic trajectory calculated with the actual parameters.

With the new trajectory, the parameters can be optimized again. Performing the parameter optimization on this new trajectory yields a new optimal set of parameters, $b = 1.0012$ Ns/m and $\alpha = 29.993$.



(a) Distribution using initial trajectory

(b) Distribution using optimized trajectory

Fig. 5: PDF histograms of the damping coefficient, b.

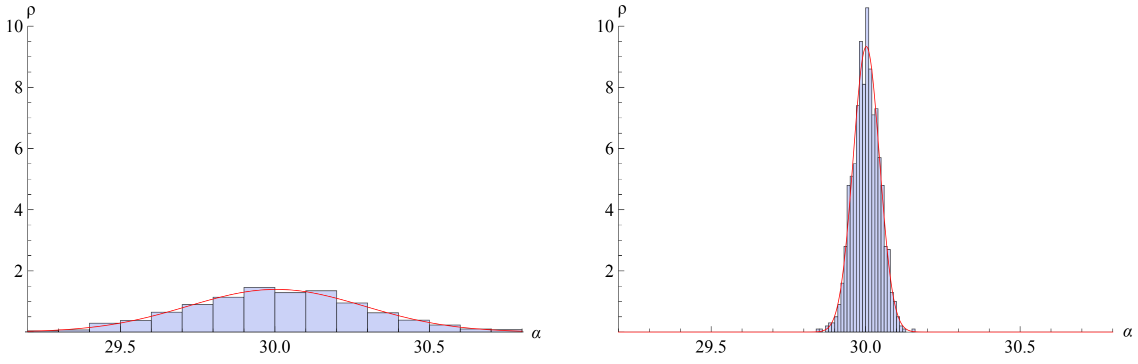
D. Variance Observations

This set of parameters is an improvement over the previous estimate using the initially chosen trajectory; however, since the trajectory is uncertain, we will perform a Monte-Carlo based simulation to observe that the variance of the optimized trajectory is lower than that of the ill-conditioned trajectory. The parameter optimization routine was run 1000 times using both the initial trajectory and the optimized trajectory. In each iteration, an additive white noise signal is interpolated from 100 random samples with a standard deviation of 0.01 and $dt = 0.04$ sec. This signal is regenerated for the "measured" trajectory, simulating the effect of measurement noise, or other additive, normally distributed noise.

A histogram of the distribution of the optimized parameters was computed for the position and velocity for both trajectories. The results are shown in Figs. 5 and 6. Additionally, a normal distribution curve was fit to the histogram data based on the parameter variance. Table I shows the variance computed for the set of 1000 estimates. As shown in the table and figures, the variance of both parameters identified by the optimized trajectory is reduced by approximately an order of magnitude over the initially selected trajectory.

TABLE I: Parameter Variance

	b (kg/s)	α
Initial chosen trajectory	0.000419	0.0818
Optimized trajectory	0.0000219	0.00183



(a) Distribution using the initial trajectory

(b) Distribution using the optimized trajectory

Fig. 6: PDF histogram of the scaling parameter, α.

V. Conclusion

By designing a different trajectory for a parameter estimation experiment, simulation results demonstrate a significant decrease in the variance of the estimated parameters. In the case of the cart-pendulum example, this decrease is an order of magnitude for each of the two parameters. This result highlights the ability to more precisely estimate a given set of parameters in the experimental setup by systematically designing the trajectory.

Depending on the experimental setup, additional factors may also be important, such as the magnitude of control inputs, or a trajectory that doesn't deviate too far from a prescribed motion. In both cases, a running cost on the trajectory states and cost on the control input can be added to the cost function to achieve a desired result.

A benefit of the continuous time formulation of the conditioning optimization is that differential equation solver packages can utilize adaptive time stepping methods throughout the trajectory and Riccati equation solutions instead of having to discretely sample at fixed time steps. This allows for faster computation, while preserving small time steps in regimes that require higher sampling to maintain accuracy.

Future goals related to the use of this algorithm include deriving an analytic link between the variance of the parameter identification method and the conditioning of the Hessian to support the Monte-Carlo analysis results. Additionally, links to quantities such as the Fisher information matrix, used in finite dimensional optimization methods, are sought in future research.

References

[1] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. New York: John Willey and Sons, 2001.

[2] L. Ljung. *System Identification: Theory for the User, 2nd ed*. Prentice Hall, Upper Saddle River, NJ, 1999.

[3] K. Hsu, K. Poolla, and T. L. Vincent, "Identification of Structured Nonlinear Systems," *IEEE Trans. on Automat. Control*, vol. 53, no. 11, pp. 2497-2513, Dec. 2008.

[4] H. Hjalmarsson and J. Martensson, "Optimal Input Design for Identification of Non-linear Systems: Learning From the Linear Case," *American Control Conference*, pp. 1572-1576, 2007.

[5] H. Jansson and H. Hjalmarsson, "Input design via LMIs admitting frequency-wise model specifications in confidence regions," *IEEE Trans. on Automat. Control*, vol. 50, no. 10, pp. 1534-1549, Oct. 2005.

[6] M. Casini, A. Garulli, and A. Vicino, "Time complexity and input design in worst-case identification using binary sensors," *46th IEEE Conference on Decision and Control*, no. 1, pp. 5528-5533, 2007.

[7] B. Armstrong, "On finding exciting trajectories for identification experiments involving systems with nonlinear dynamics," *Int. J. Robot. Res.*, vol. 8, no. 6, pp. 28-48, 1989.

[8] M. Gautier and W. Khalil, "Exciting trajectories for the identification of base inertial parameters of robots," *Int. J. Robot. Res.*, vol. 11, no. 4, pp. 362-375, 1992.

[9] J. Swevers, C. Ganseman, B. D. Tkel, J. De Schutter, and H. Van Brussel, "Optimal robot excitation and identification," *IEEE Trans. Robot. Automat.*, vol. 13, pp. 730-740, May 1997.

[10] R. Mehra, "Optimal input signals for parameter estimation in dynamic systems–Survey and new results," *IEEE Trans. on Automat. Control*, vol. 19, no. 6, pp. 753-768, Dec. 1974.

[11] T. L. Vincent, C. Novara, K. Hsu, and K. Poolla, "Input design for structured nonlinear system identification," *Automatica*, vol. 46, no. 6, pp. 990-998, Jun. 2010.

[12] M. Baltes, R. Schneider, C. Sturm, and M. Reuss, "Optimal experimental design for parameter estimation in unstructured growth models," *Biotechnology Progress*, vol. 10, no. 5, pp. 480-488, Sep. 1994.

[13] D. Faller, U. Klingmuller, and J. Timmer, "Simulation Methods for Optimal Experimental Design in Systems Biology," *SIMULATION*, vol. 79, no. 12, pp. 717-725, Dec. 2003.

[14] C. Jauberthie, L. Denis-Vidal, P. Coton, and G. Joly-Blanchard, "An optimal input design procedure," *Automatica*, vol. 42, no. 5, pp. 881-884, May 2006.

[15] W. Wu, S. Zhu, X. Wang, and H. Liu, "Closed-loop dynamic parameter identification of robot manipulators using modified Fourier series," *Int. J. Adv. Robot. Sys.*, vol. 9, no. 29, 2012.

[16] W. Rackl, R. Lampariello, and G. Hirzinger, "Robot excitation trajectories for dynamic parameter estimation using optimized B-splines," *2012 IEEE Conf. Robot. and Automat.*, pp. 2042-2047, May 2012.

[17] K. Radkhah, D. Kulic, and E. Croft, "Dynamic parameter identification for the CRS A460 robot," *2007 Conf. Int. Robots and Sys.*, pp. 3842-3847, Oct 2007.

[18] H. H. Jonas Mrtensson, "Robust input design using sum of squares constraints," *Proc. IFAC Symposium*, pp. 1352-1357, 2006.

[19] J. Hauser, "A projection operator approach to optimization of trajectory functionals," *IFAC world congress*, Barcelona, Spain, 2002.

[20] D. Ozyurt, B. Derya, and P. Barton, "Cheap second order directional derivatives of stiff ode embedded functionals," *SIAM Journal of Scientific Computing*, vol. 26, no. 5, pp. 1725-1743, May 2005.

[21] L. Miller and T. Murphey, "Simultaneous optimal parameter and mode transition time estimation," *IEEE Int. Conf. on Int. Robots and Sys.*, 2012.

[22] R. E. Kalman and R. S. Bucy, "New Results in Linear Filtering and Prediction Theory," *J. of Basic Eng.*, vol. 83, pp. 95-108, 1961.

[23] B.D.O. Anderson and J.B. Moore, *Optimal Control: Linear Quadratic Methods*, Prentice-Hall, 1989.

[24] R.B. Nelson. "Simplified calculation of eigenvector derivatives," *AIAA Journal*, vol. 14, no. 9, pp. 1201-1205, September 1976.