Optimal Parameter Identification for Discrete Mechanical Systems with Application to Flexible Object Manipulation

T. M. Caldwell and D. Coleman and N. Correll

***Abstract*—**

I. INTRODUCTION

This paper considers the problem of optimal parameter identification for mechanical systems described by variational integrators. The motivating example is a flexible loop clamped at one end and manipulated at the other. By stretching, twisting, and bending the loop, the manipulator displaces the point of contact. With the displacement measurement, along with the torques and forces that affected the displacement, the stiffness properties of the loop can be identified. The identification approach is an optimal one for which we provide an adjoint-based gradient calculation.

We model the loop as six links forming a hexagon. The links connect together through spherical joints attached with tosional springs. The goal of the example is to identify the torsional springs' spring constants. Alternatively, other parameters could be identified like lengths, masses, and damping coefficients. The presented approach also extends to identification of the coefficient of friction of a system's interaction with a surface similar to **cite Miller**.

Although we are concerned with flexible objects, we model them as connected rigid bodies for which modeling techniques are well understood **cite murray li sastry**. For this reason, the presented parameter identification approach makes use of standard rigid body modeling techniques **cite Murray Li Sastry**. Furthermore, we use variational integrators for simulation in order to provide a means to preserve—or nearly preserve—physical properites like conservation of energy and momentum **cite West Thesis, Johnson scalability, others... Johnson cites J. E. Marsden and M. West, Discrete mechanics and variational integrators, Acta Numer., vol. 10, pp. 357514, 2001**. This decision was made becasue simulation accuracy is a concern for lower order integrators which can significantly introduce energy errors. At worst these errors will destabilize the integration and at best compromise the model's dissipation **cite Johnson's scalability paper**.

Numerical issues are exacerbated for systems with closed chains—which is the case for the motivating example—since numerical error during Simulation can invalidate holonomic constraints. This problem arises due to the way holonomic constraints are included in continuous dynamics, which is with equivalent acceleration constraints. In comparison, variational integrators apply the holonomic constraints directly and do not have this issue **Johnson and Murphey scalable**.

This material is .....

Due to recent work by Johnson and Murphey **cite both Johnson and Murphey papers**, it is possible to efficiently simulate mechanical systems using variational integrators in generalized coordinates. They provide a framework using a tree representation and caching that not only makes for efficient simulation using variational integrators—especially for articulated rigid bodies—but also efficient model based calculations like linearizations about a trajectory. In optimal controls, linearizations are needed for gradient calculations like the gradient calculation presented in this paper for identifying parameters. As such, we review mechanical systems, variation integrators and give the system Jacobian as derivatives of the system's Lagrangian and forcing.

The parameter identification optimization problem is set up as a discrete-time Bolza type problem. For the loop example, the cost function is a summation of the error of the loop displacement at the manipulator compared to the simulated displacement for given spring constants. Alternatively, the cost can be given by a maximum likelihood estimate for which the cost is lower for parameters that coorespond to the simulation that is most consistent with the measurement (see **cite parameter estimation book and houska**). The objective is to calculate the spring constants that minimize the cost.

The paper's contributions are twofold. First, it provides a discrete-time adjoint-based gradient calculation for optimal parameter estimation. Second, it formulates parameter identification of mechanical systems with variational integrators for greater confidence in the accuracy of the model's simulation.

This paper is organized as follows: ....
XXXXXXXXXXXX

II. MECHANICAL SYSTEMS

A comparison of continuous time and discrete time mechanical systems is presented. The mechanical system depends on n_a system parameters from the parameter space $\mathcal{A} \in \mathbb{R}^m$. The mechanical system has n_q generalized coordinates $q \in \mathbb{R}^{n_q}$. For continuous time, q varies the continuous variable t—e.g. $q(t)$—for t in the interval $[0, t_f]$, $t_f > 0$. Likewise, for the discrete representation, q varies with the discrete variable k—e.g. q_k—for k in the set $\mathcal{K} := \{0, 1, \ldots, k_f\}$, $k_f > 0$. Each discrete time k pairs with a continuous time, labeled t_k, where $t_0 = 0$, $t_{k_f} = t_f$ and $t_k < t_{k+1}$. This section presents the continuous and discrete dynamics dependent on the parameters $a \in \mathcal{A}$. The continuous dynamics are provided for comparison with the discrete system, in which the paper results are given.

A. Continuous Mechanical System

We review continuous mechanical systems for reference with discrete mechanical systems.

A mechanical system's evolution is given by the path of least action. The system's action is

$$S = \int_0^{t_f} L(q(\tau), \dot{q}(\tau), a) d\tau$$

where $L(q, \dot{q}, a) := KE(q, \dot{q}, a) - V(q, a)$, the system Lagrangian, is the difference of the system's kinetic energy, KE, with its potential energy, V. For this paper, we assume that both kinetic and potential energies depend on system parameters $a \in \mathcal{A}$. Possible parameters of L are lengths, spring constants, and masses.

As is common in mechanical systems, we wish to include external forces, $F_c(q, \dot{q}, a, t)$. This term is the total external forcing in the generalized coordinates. We also assume dependence on parameters $a \in \mathcal{A}$. For example, likely parameter candidates showing in F_c are damping coefficients. By including the additional external force term, F_c, to the action, the Lagrange d'Alembert principle finds that the continuous dynamics of the mechanical system are given by the forced Euler-Lagrange equations **cite**:

$$\frac{d}{dt}\frac{\partial}{\partial \dot{q}} L(q, \dot{q}, a) - \frac{\partial}{\partial q} L(q, \dot{q}, a) = F_c(q, \dot{q}, a, t).$$

Holonomic constraints can be enforced as external forces using Lagrange multipliers. The n_h holonomic constraint equations are given in the form $h(q, a) = [h_1, \ldots, h_{n_h}]^T(q, a) = 0$. With the addition of the constraint, the forced Euler-Lagrange equations are **cite**:

$$\begin{gathered} \tfrac{d}{dt}\tfrac{\partial}{\partial \dot{q}} L(q, \dot{q}, a) - \tfrac{\partial}{\partial q} L(q, \dot{q}, a) = F_c(q, \dot{q}, a, t) + \tfrac{\partial}{\partial q} h^T(q, a)\lambda \\ \tfrac{\partial^2}{\partial q^2} h(q, a) \circ (\dot{q}, \dot{q}) + \tfrac{\partial}{\partial q} h(q, a)\ddot{q} = 0 \end{gathered} \tag{1}$$

where $\lambda(t) \in \mathbb{R}^{n_h}$ are Lagrange multipliers. For fixed parameters a, the system's evolution is integrated from Eq.(1) for q, $\dot{q}$ and λ. It is worth noting that holonomic constraints are not enforced directly but instead the acceleration is constrained. During integration, numerical error will violate the constraint and $h(q, a) \neq 0$.

Equation 1 can be transformed into first-order state space equations. Define the continuous state as $x = [q, \dot{q}]^T$. The state equations, dependent on the parameters $a \in \mathcal{A}$, are $\dot{x}(t) = f(x(t), \lambda(t), a, t)$ where $\ddot{q}$ is specified by the forced Euler-Lagrange equations. In the state space representation, gradient and Hessian calculations with respect to parameters are given for parameter optimization in **cite Lauren**.

B. Discrete Mechanical System

The discrete mechanical system is an approximation of its continuous counterpart. For an initial configuration $q(0)$ and velocity $\dot{q}(0)$, the continuous configuration $q([0, t_f])$ is integrated from the forced Euler-Lagrange equations, Eq.(1). The discrete analog to the forced Euler-Lagrange equations instead calculates the sequence $q_k \approx q(t_k)$ using a variational integrator approach **cite**.

The discrete Lagrangian, labelled L_d, is an approximation of the action over a short time interval. Instead of a velocity term, the discrete Lagrangian is defined by the current and next configurations, q_k and q_{k+1}:

$$L_d(q_k, q_{k+1}, a) \approx \int_{t_k}^{t_{k+1}} L(q(\tau), \dot{q}(\tau), a) d\tau. \tag{2}$$

The integration can be approximated with a quadrature like midpoint or trapezoidal rules. Refer to **cite Elliot** for details of using midpoint rule, which we use in the example.

Similarly, external forcing is included by approximating F_c by the discrete left and right forces $F_d^-(q_k, q_{k+1}, a, t_k, t_{k+1})$ and $F_d^+(q_k, q_{k+1}, a, t_k, t_{k+1})$ using a quadrature. In addition, the n_h holonomic constraints $h(q_k, a)$ can be enforced with the n_h Lagrange multipliers λ_k. With the discrete Lagrangian, discrete forces and holonomic constraints, **cite (Elliot?)**, finds that the forced discrete Euler-Lagrange equations are:

$$\begin{cases} D_2 L_d(q_{k-1}, q_k, a) + F_d^+(q_{k-1}, q_k, a, t_{k-1}, t_k) \\ \quad +D_1 L_d(q_k, q_{k+1}, a) + F_d^-(q_k, q_{k+1}, a, t_k, t_{k+1}) \\ \quad -D_1 h^T(q_k, a)\lambda_k = 0. \\ h(q_{k+1}, a) = 0 \end{cases} \tag{3}$$

These equations should be viewed as an implicit function on q_{k+1}. For example, given consecutive configurations q_0 and q_1, the next configuration q_2 is found with a root solving operation on Eq.(3). Following, q_3 is obtained from q_1 and q_2 and so forth. Whereas the continuous mechanical system is solved using integration, the discrete mechanical system is solved through recursive calls to root finding Eq.(3).

As in **cite johnson**, define p_k as

$$p_k := D_2 L_d(q_{k-1}, q_k, a) + F_d^+(q_{k-1}, q_k, a, t_{k-1}, t_k). \tag{4}$$

Without external forcing, p_k is the conserved momentum. With forcing, though, the physical interpretation for p_k is less clear and is defined simply for computational convenience. This convenience comes from defining the discrete state as $x_k := [q_k, p_k]^T$ which has a one-step mapping:

$$\begin{gathered} x_{k+1} = f(x_k, \lambda_k, a, t_k) := \\ \begin{cases} p_k + D_1 L_d(q_k, q_{k+1}, a) + F_d^-(q_k, q_{k+1}, a, t_k, t_{k+1}) \\ \quad -D_1 h^T(q_k, a)\lambda_k = 0 \\ h(q_{k+1}, a) = 0 \\ p_{k+1} = D_2 L_d(q_k, q_{k+1}, a) + F_d^+(q_k, q_{k+1}, a, t_k, t_{k+1}). \end{cases} \end{gathered} \tag{5}$$

This equation is the state equation for the discrete mechanical system. The function $f(x_k, a, t_k)$ is implicit, but, according to the Implicit Function Theorem, it exists when

$$M_{k+1} := D_2 D_1 L_d(q_k, q_{k+1}, a) + D_2 F_d^-(q_k, q_{k+1}, a, t_k, t_{k+1})$$

is nonsingular. By assuming nonsingularity, even though $f(x_k, a, t_k)$ is implicit, the linearization around x_{k+1}—i.e. $\frac{\partial x_{k+1}}{\partial x_k}$—is explicit. Letting $dx_k = [dq_k, dp_k]^T$, be the differential of x_k and da be the differential of a, the linearization of $f(x_k, a, t_k)$ is:

$$\begin{bmatrix} dq_{k+1} \\ dp_{k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{\partial q_{k+1}}{\partial q_k} & \frac{\partial q_{k+1}}{\partial p_k} \\ \frac{\partial p_{k+1}}{\partial q_k} & \frac{\partial p_{k+1}}{\partial p_k} \end{bmatrix}}_{A_k} \begin{bmatrix} dq_k \\ dp_k \end{bmatrix} + \underbrace{\begin{bmatrix} \frac{\partial q_{k+1}}{\partial a} \\ \frac{\partial p_{k+1}}{\partial a} \end{bmatrix}}_{B_k} da \tag{6}$$

The calculations for the linearization term A_k is given in **cite ELliot** and duplicated here for reference:

$$\frac{\partial q_{k+1}}{\partial q_k} = \\ -M_{k+1}^{-1}[D_1^2 L_d(q_k, q_{k+1}, a) + D_1 F_d^-(q_k, q_{k+1}, a, t_k, t_{k+1}) \\ -D_1^2 h^T(q_k, a)\lambda_k - D_1 h^T(q_k, a)\frac{\partial \lambda_k}{\partial q_k}] \tag{7a}$$

$$\frac{\partial q_{k+1}}{\partial p_k} = -M_{k+1}^{-1} \tag{7b}$$

$$\frac{\partial p_{k+1}}{\partial q_k} = \\ [D_2^2 L_d(q_k, q_{k+1}, a) + D_2 F_d^+(q_k, q_{k+1}, a, t_k, t_{k+1})]\frac{\partial q_{k+1}}{\partial q_k} \\ +D_1 D_2 L_d(q_k, q_{k+1}, a) + D_1 F_d^+(q_k, q_{k+1}, a, t_k, t_{k+1}) \tag{7c}$$

$$\frac{\partial p_{k+1}}{\partial p_k} = \\ [D_2^2 L_d(q_k, q_{k+1}, a) + D_2 F_d^+(q_k, q_{k+1}, a, t_k, t_{k+1})]\frac{\partial q_{k+1}}{\partial p_k} \tag{7d}$$

where $\frac{\partial \lambda}{\partial q_k}$ can be found in **cite elliot...**. Notice the calculatons for Eqs (7c) and (7d) rely on the calculations for Eqs (7a) and (7b) respectively. The B_k term is given by chain rule:

$$\frac{\partial q_{k+1}}{\partial a} = \frac{\partial q_{k+1}}{\partial p_k}\frac{\partial p_k}{\partial a} + \frac{\partial q_{k+1}}{\partial q_k}\frac{\partial q_k}{\partial a} + M_{k+1}^{-1}[D_1 D_2 h^T(q_k, a)\lambda_k \\ -D_3 D_1 L_d(q_k, q_{k+1}, a) - D_3 F_d^-(q_k, q_{k+1}, a, t_k, t_{k+1})] \tag{8a}$$

$$\frac{\partial p_{k+1}}{\partial a} = \\ [D_2^2 L_d(q_k, q_{k+1}, a) + D_2 F_d^+(q_k, q_{k+1}, a, t_k, t_{k+1})]\frac{\partial q_{k+1}}{\partial a} \\ +[D_1 D_2 L_d(q_k, q_{k+1}, a) + D_1 F_d^+(q_k, q_{k+1}, a, t_k, t_{k+1})]\frac{\partial q_k}{\partial a} \\ +D_3 D_2 L_d(q_k, q_{k+1}, a) + D_3 F_d^+(q_k, q_{k+1}, a, t_k, t_{k+1}) \tag{8b}$$

The term B_k depends on A_k and the previous term B_{k-1}. The linearization of Eq.(5) is needed for calculations of the parameter identification gradient.

III. Parameter Optimization

The goal of parameter optimization is to calculate the system parameters $a \in \mathcal{A}$ that minimize a cost functional. For the continuous problem, the cost functional is the integral of a running cost $\ell(x(t), t, a)$ plus a terminal cost $m(x(t_f), a)$:

Problem 1 (Continuous System Parameter Optimization): Calculate the parameters $a \in \mathcal{A}$ which solves:

$$\min_{a \in \mathcal{A}} \left[J(a) := \int_0^{t_f} \ell(x(t), a) dt + m(x(t_f), a) \right]$$

constrained to $\dot{x}(t) = f(x(t), a, t)$.

For the discrete problem, it is reasonable to choose a discrete cost function that approximates the continuous cost—i.e. $\ell_d(x_{k-1}, x_k, a) \approx \int_{t_{k-1}}^{t_k} \ell(x(\tau), a) d\tau$ and $m_d(x_{k_f}, a) \approx m(x(t_f), a)$. Alternatively, ℓ_d and m_d can be designed directly without first choosing an underlying continuous cost. The discrete parameter optimization problem is as follows:

Problem 2 (Discrete System Parameter Optimization): Calculate the parameters $a \in \mathcal{A}$ which solves:

$$\min_{a \in \mathcal{A}} \left[J_d(a) := \sum_{k=1}^{k_f} \ell_d(x_k, a) + m_d(x_{k_f}, a) \right]$$

constrained to $x_{k+1} = f(x_k, a, t_k)$, Eq.(5).

In optimal control theory, it is common practice to solve optimization problems using iterative methods. Iterative optimization methods repeatedly reduce the cost by stepping in a descending direction until a local optimum is found. Commonly, the step direction and step size is calculated using local derivative information **cite Kelley; Luenberger; Armijo**, which is the case for this paper. In the next section, we provide an adjoint-based calculation for the gradient of the cost function with respect to the parameters.

A. Discrete System Parameter Gradient

The gradient of the cost function given in problem 2 is provided in the following lemma.

Lemma 1: Suppose $L_d(q_k, q_{k+1}, a)$, $F_d^-(q_k, q_{k+1}, a, t_k, t_{k+1})$, $F_d^+(q_k, q_{k+1}, a, t_k, t_{k+1})$, and $h(q_k, a)$ are $\mathcal{C}^2$ with respect to q_k, q_{k+1} and a. Take A_k and B_k from Eq.(6) and assume M_k is always nonsingular. Then,

$$DJ_d(a) = \sum_{k=1}^{k_f} \lambda_k B_{k-1} + D_2 \ell_d(x_k, a) + D_2 m_d(x_{k_f}, a) \tag{9}$$

where λ_k is the the solution to the backward one-step mapping

$$\lambda_k = \lambda_{k+1} A_k + D_1 \ell_d(x_k, a) \tag{10}$$

starting from $\lambda_{k_f} = D_1 \ell(x_{k_f}, a) + D_1 m_d(x_{k_f}, a)$.

Proof: The derivative of the cost in the direction $\theta \in \mathbb{R}^{n_a}$ is

$$DJ_d(a)\theta := \sum_{k=1}^{k_f} D_1 \ell_d(x_k, a)\frac{\partial x_k}{\partial \theta} + D_2 \ell_d(x_k, a)\theta \\ +D_1 m_d(x_{k_f}, a)\frac{\partial x_{k_f}}{\partial \theta} + D_2 m_d(x_{k_f}, a)\theta. \tag{11}$$

Label $z_k := \frac{\partial x_k}{\partial \theta}$ for convenience. Also for convenience, label

$$H := \sum_{k=1}^{k_f} D_1 \ell_d(x_k, a) z_k + D_1 m_d(x_{k_f}, a) z_{k_f} \tag{12}$$

The linearized state, z_k, is the solution to the linearized state equation, Eq.(6). In other words,

$$z_{k+1} = A_k z_k + B_k \theta$$

starting from $z_0 = 0$. The linearized state's solution depends on the discrete state transition matrix defined as

$$\Phi(k_2, k_1) := \prod_{j=1}^{k_2 - k_1} A_{k_2 - j} = A_{k_2-1} A_{k_2-2} \cdots A_{k_1}$$

for integers $k_2 > k_1$ and where $\Phi(k_1, k_1) := I$, the identity matrix. Recalling $z_0 = 0$, the linearized state's solution is

$$z_k = \Phi(k, 0) z_0 + \sum_{s=0}^{k-1} \Phi(k, s+1) B_s \theta = \sum_{s=0}^{k-1} \Phi(k, s+1) B_s \theta$$

for $k = 1, \ldots, k_f$. Plugging z_k into H, Eq.(12), H becomes

$$\begin{aligned} H &= \textstyle\sum_{k=1}^{k_f} D_1\ell_d(x_k, a) \sum_{s=0}^{k-1} \Phi(k, s+1) B_s\theta \\ &\quad + D_1 m_d(x_{k_f}, a) \textstyle\sum_{s=0}^{k_f-1} \Phi(k_f, s+1) B_s\theta \\ &= \textstyle\sum_{k=1}^{k_f} \sum_{s=0}^{k-1} D_1\ell_d(x_k, a)\Phi(k, s+1) B_s\theta \\ &\quad + \textstyle\sum_{s=0}^{k_f-1} D_1 m_d(x_{k_f}, a)\Phi(k_f, s+1) B_s\theta \end{aligned}$$

Switch the order of the double sum.

$$\begin{aligned} H &= \textstyle\sum_{s=0}^{k_f-1} \sum_{k=s+1}^{k_f} D_1\ell_d(x_k, a)\Phi(k, s+1) B_s\theta \\ &\quad + D_1 m_d(x_{k_f}, a) \textstyle\sum_{s=0}^{k_f-1} \Phi(k_f, s+1) B_s\theta \\ &= \textstyle\sum_{s=0}^{k_f-1} \Big[\sum_{k=s+1}^{k_f} D_1\ell_d(x_k, a)\Phi(k, s+1) \\ &\quad + D_1 m_d(x_{k_f}, a)\Phi(k_f, s+1) \Big] B_s\theta \end{aligned}$$

Set λ_{s+1} as the resulting covector in the brackets so that

$$H = \sum_{s=0}^{k_f-1} \lambda_{s+1} B_s\theta = \sum_{k=1}^{k_f} \lambda_k B_{k-1}\theta.$$

The efficient calculation for λ_k is given in Eq.(10). Plugging H into Eq.(11), we find

$$DJ(a)\theta = \Big[\sum_{k=1}^{k_f} \lambda_k B_{k-1} + D_2\ell_d(x_k, a) + D_2 m_d(x_{k_f}, a) \Big] \theta$$

and the proof is finished. ∎

IV. Example

As an example, we model a flexible loop and manipulate it at a contact point in order to ascertain its stiffness properties. The loop model is composed of 6 links forming a hexagon as shown in Fig. 1. Each joint has two configuration variables, labeled θ_i and ψ_i. The configuration variables constitute rotations around the X-axis and Z-axis, respectively, of frame $L_{i,e}$. We attach the 6 links starting from the start frame $\mathcal{S}$. Therefore, the position and orientation of the frame at the end of the i^{th} link, frame $L_{i,e}$, depends on the configuration variables $\theta_1, \ldots, \theta_{i-1}$ and $\psi_1, \ldots, \psi_{i-1}$.

Each joint has two torsional springs, one for each configuration. These springs generate forces at the joints which drive the loop toward the nominal regular hexagonal shape. By assuming uniformity of the loop, the spring force on configuration variable θ_i (alt. ψ_i) is specified by spring constant κ_θ (κ_ψ) for each joint i—i.e. the spring torque is $T_{\theta_i} = \kappa_\theta \theta_i$ ($T_{\psi_i} = \kappa_\psi \psi_i$). The exercise in this example is to estimate the parameters κ_θ and κ_ψ and so we set $a = [\kappa_\theta, \kappa_\psi]^T$.

For arbitrary configuration values, there is no guarantee that the end loop frame, $\mathcal{E}$, lies on top of the start loop frame, $\mathcal{S}$. Therefore, we use 6 holonomic constraints to constrain the position and orientation of $\mathcal{E}$ to $\mathcal{S}$ in order to maintain the loop structure during integration of the dynamics. The first two constraints, h_1 and h_2, constrain the point $[1, 0, 0]^T$ in the $\mathcal{E}$ frame to the X-axis of the $\mathcal{S}$ frame. Similarly, h_3 and h_4 constrain the point $[0, 1, 0]^T$ in $\mathcal{E}$ to the Y-axis of $\mathcal{S}$ while h_5 and h_6 constrain $[0, 0, 1]^T$ in $\mathcal{E}$ to the Z-axis of $\mathcal{S}$. When the configuration variables are so that $h_i = 0$ for each $i = 1, \ldots, 6$, the $\mathcal{E}$ frame lies on top of the $\mathcal{S}$ frame and the loop is "closed".

Since we decided to connect the beginning and end of the loop at a corner, an additional configuration variable is needed so that the final link is not rigidly attached to the $\mathcal{E}$ frame. Figure 1a shows this final configuration variable, θ_7. Thus, the loop configuration is given by $\Theta := [\theta_1 \ldots, \theta_7]$ and $\Psi := [\psi_1, \ldots, \psi_6]$ and is $q = [\Theta, \Psi]^T$.

To give the holonomic constraint equations, $h(q, a) = [h_1(q), \ldots, h_6(q)]^T$, we refer to the notion of the rigid body transformation in $SE(3)$ (please see **cite Murray Li Sastry**. Specifically, we need the homogenous representation of the transformation from frame $\mathcal{S}$ to frame $\mathcal{E}$, which we label $g_{\mathcal{SE}}(q) \in \mathbb{R}^{4\times4}$, since $g_{\mathcal{SE}}(q)$ will give the position in the $\mathcal{S}$ frame of point w_E in the $\mathcal{E}$ frame:

$$\begin{bmatrix} w_S \\ 1 \end{bmatrix} = g_{\mathcal{SE}}(q) \begin{bmatrix} w_E \\ 1 \end{bmatrix}.$$

The additional 1 appended at the end of the vectors is an artifact of using $SE(3)$. Now, a reasonable choice for constraints h_1 and h_2, which constrain the point $[1, 0, 0]^T$ in the $\mathcal{E}$ frame to the X-axis of the $\mathcal{S}$ frame, is:

$$h_1(q) = \left(g_{\mathcal{SE}}(q) \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \right)^T \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \tag{13}$$

and

$$h_2(q) = \left(g_{\mathcal{SE}}(q) \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \right)^T \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}. \tag{14}$$

The other 4 constraints have a similar form.

A. Simulation

The continuous state is given by q and its time derivatives—i.e. $x(t) = [q(t), \dot{q}(t)]^T$. Similarily, the system's discrete time state is $x_k = [q_k, p_k]^T$ where p_k is defined in Eq.(4). The continuous dynamics are given by the constrained, forced Euler Lagrange equations, Eq.(1), which depends on the system Lagrangian $L(q, \dot{q}, a)$, external forces $F_c(q, \dot{q}, a, t)$ and holonomic constraints $h(q, a)$. Techniques to derive these formulas for rigid bodies are well understood **Cite Murray Li Sastry**.

For the loop example, we model the center of mass of each link to be at the center of each link and that the mass of each link is 0.2 Kg. We assume some manipulator applies a force at frame $L_{3,e}$ as shown in Fig. 1. The force in the $L_{3,e}$ is $F = [20\cos(t/50), 10\cos(t/50+\pi/3), 20\cos(t/50+2\pi/3)]^T$. Furthermore, we set the torisional spring constants as $\kappa_\theta = \kappa_\psi = 20$. Later we identify these constants using the techniques in Section XXXXXXX.

With the continuous dymamics, it is straightforward to obtain the discrete dynamics, which are given in Eq.(3). The discrete dynamics depend on the discrete system Lagrangian $L_d(q_k, q_{k+1}, a)$, discrete external left and right forces,

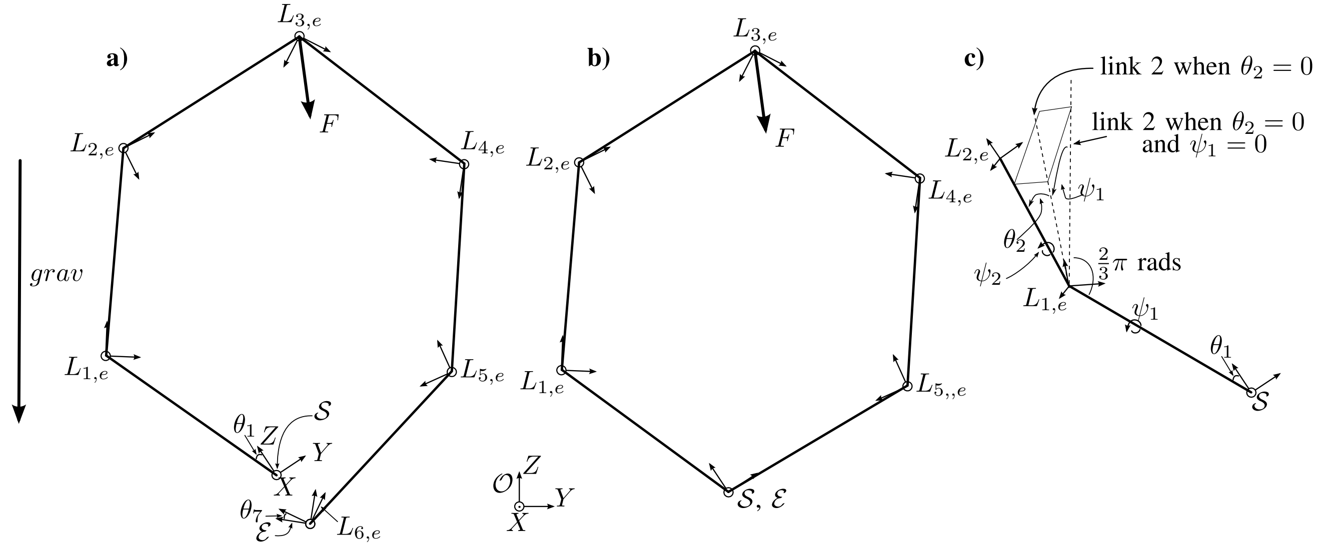

Fig. 1. Illustration of the loop approximated by a hexagon of links. The frame at the end of link 1, frame $L_{e,1}$, is defined from $\mathcal{S}$ by the following steps: 1) rotation around X-axis by θ_1 radians, 2) rotation around Z-axis by ψ_1 radians, 3) translation 0.5 meters along Z-axis, 4) rotation around X-axis by $2/3\pi$ radians. The frame $L_{e,i+1}$ is obtained from $L_{e,i}$ through the same rotations and translations except for rotation values θ_i and ψ_i. Finally, the end frame $\mathcal{E}$ is a rotation of θ_7 around the X-axis of $L_{6,e}$. When $\theta_1 = \cdots = \theta_7 = \psi_1 = \cdots = \psi_6 = 0$, the loop is a regular hexagon in the $Y-Z$ plane. **a)** shows the loop for arbitrary rotation values which violate the holonomic constraints that the loop is "closed"—i.e. $h \neq 0$ while **b)** shows rotation values that satisfy the constraint—i.e. $h = 0$. Furthermore, we apply a wrench at the $L_{3,e}$ frame as shown by F and gravity is in the negative Z direction of the $\mathcal{O}$ frame. **c)** shows the link from $L_{1,e}$ to $L_{2,e}$ for non-zero θ_2 and ψ_1.

$F_d^-(q_k, q_{k+1}, a, t_k, t_{k+1})$ and $F_d^+(q_k, q_{k+1}, a, t_k, t_{k+1})$, and holonomic constraints $h(q_k, a)$.

For simulation, we chose a constant time step $\Delta_t = t_k - t_{k+1}$ at each discrete time k. Further, we decided to approximate L_d from L using midpoint rule (see Eq.(15)):

$$L_d(q_k, q_{k+1}, a) = \Delta_t L(\frac{q_{k+1} + q_k}{2}, \frac{q_{k+1} - q_k}{\Delta_t}, a). \quad (15)$$

Similarily, using midpoint rule, we approximate F_c by F_d^- and F_d^+ where

$$\begin{cases} F_d^-(q_k, q_{k+1}, a, t_k, t_{k+1}) = \\ \qquad \Delta_t F_c(\frac{q_{k+1}+q_k}{2}, \frac{q_{k+1}-q_k}{\Delta_t}, a, \frac{t_{k+1}+t_k}{2}) \\ F_d^+(q_k, q_{k+1}, a, t_k, t_{k+1}) = 0. \end{cases} \quad (16)$$

Therefore, once the continuous system has been modeled it is simple to tranlate it to discrete time. Finally, in order to simulate the dynamics using the one-step mapping in Eq.(5), we additionally need certain partial derivatives of the Langrangian, external forces and constraints with respect to configuration variables, which can be found in **CITE Elliot**.

Aside: We used the software tool TREP **CITE** which simulates articulated rigid bodies using midpoint variational integrators. Furthermore, it provides additional partial derivative calculations that we need for the system linearization, Eqs.(7) and (8).

B. Linearization

The linearization of the the discrete equations of motion is given by matrices A_k and B_k in Eqs.(7) and (8). We need the linerization for the gradient calculation, Lemma 1, in order to perform a gradient-based descent algorithm like steepest descent. Partial derivatives of L_d and F^+ with respect to q_k and p_k can be obtained from Eqs.(15) and (16).

For the loop example, we need to calculate $D_3D_1L_d(q_k, q_{k+1}, a)$ and $D_3D_1L_d(q_k, q_{k+1}, a)$ for $a = [\kappa_\theta, \kappa_\psi]$. Note that the potential energy of the system can be written as:

$$V(q, a) = V_\theta(q, \kappa_\theta) + V_\psi(q, \kappa_\psi) + V_g(q)$$

where V_θ, V_ψ and V_g are the potential energies due to the Θ spring torques, the Ψ spring torques, and gravity, respectively. Recalling the first 7 configuration variables in q are in Θ, the potential energy due to the Θ spring torques is $V_\theta(q, \kappa_\theta) = \sum_{i=1}^{7} \frac{1}{2}\kappa_\theta\theta^2$. Approximating for the discrete time potential energy—see Eq.(15)—we find that

$$V_{\theta,d}(q_k, q_{k+1}, \kappa_\theta) = \sum_{i=1}^{7} \frac{\Delta_t}{2}\kappa_\theta(\frac{q_{k+1} + q_k}{2})^2.$$

Taking the needed partial derivatives:[1]

$$D_3D_1V_{\theta,d}(q_k, q_{k+1}, \kappa_\theta) = D_3D_2V_{\theta,d}(q_k, q_{k+1}, \kappa_\theta)$$
$$= [\tfrac{\Delta_t}{4}(q_{1,k+1} + q_{1,k}), \ldots, \tfrac{\Delta_t}{4}(q_{7,k+1} + q_{7,k}), 0, \ldots, 0]^T$$

Similarly,

$$D_3D_1V_{\psi,d}(q_k, q_{k+1}, \kappa_\psi) = D_3D_2V_{\psi,d}(q_k, q_{k+1}, \kappa_\psi)$$
$$= [0, \ldots, 0, \tfrac{\Delta_t}{4}(q_{8,k+1} + q_{8,k}), \ldots, \tfrac{\Delta_t}{4}(q_{13,k+1} + q_{13,k})]^T.$$

Since the kinetic energy does not depend on $a = [k_\theta, k_\psi]$,

$$D_3D_1L_d(q_k, q_{k+1}, a) =$$
$$-[D_3D_1V_{\theta,d}(q_k, q_{k+1}, \kappa_\theta), D_3D_1V_{\psi,d}(q_k, q_{k+1}, \kappa_\psi)].$$

Repeating the derivation for $D_3D_2L_d(q_k, q_{k+1}, a)$ we find that $D_3D_2L_d(q_k, q_{k+1}, a) = D_3D_1L_d(q_k, q_{k+1}, a)$

[1] Here, we index the i^{th} term of q_k as $q_{i,k}$.

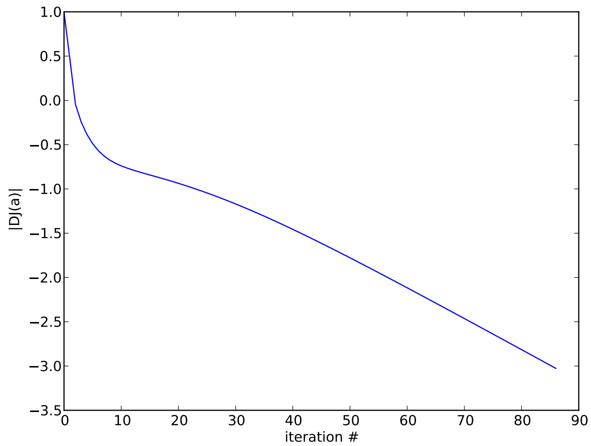

Fig. 2. Convergence of optimization algorithm. The convergence rate appears to be linear. **(NOTE I'LL MAKE THIS FIGURE NICER")**

Furthermore, we need to calculate $D_1 h(q_k, a)$, $D_1^2 h(q_k, a)$ and $D_1 D2h(q_k, a)$, the last of which is 0 since the constraints do not depend on the parameters a. These partial derivatives of h are given simply by chain rule and depend on $Dg_{\mathcal{SE}}(q_k)$ and $D^2 g_{\mathcal{SE}}(q_k)$ which can be found in **cite Elliot's Linearization paper**.

C. Optimal Parameter Identification

Earlier, we simulated the loop's dynamics for spring constant values of $\kappa_\theta = \kappa_\psi = 20$. Now, we wish to identify these spring constants using only position measurements that were taken at the frame where the force was applied, frame $L_{3,e}$. It is reasonable to assume a robotic manipulator could measure its end effector's position as it manipulates an object.

In order to optimally identify the parameters, the cost function, J_d must reflect the measurement data. Let $w_{k,meas}$ be the measured position of the end effector at time t_k, which corresponds to the origin of $L_{3,e}$ for the measured loop. The optimization problem is to find the parameters $a = [\kappa_\theta, \kappa_\psi]^T$ which correspond to a simulation for which the origin of $L_{3,e}$ best matches $w_{k,meas}$. Therefore, we set the running and terminal costs to: $\ell_d(q_k, a) = 1/2(w_k - w_{k,meas})^T (w_k - w_{k,meas})$ and $m_d(q_{k_f}, a) = 1/2(w_{k_f} - w_{k_F,meas})^T (w_k - w_{k_f,meas})$ where w_k is the origin of the frame $L_{3,e}$ for configuration q_k.

Starting with an initial guess of $a = [10, 25]^T$, we execute steepest descent with an Armijo line search which has parameters $\alpha = \beta = 0.4$ **cite Armijo**. After 86 iterations, the algorithm terminates with gradient norm $|DJ_d(a)| < 10^{-3}$. The parameters are identified as $[19.9997, 20.0120]^T$. The convergence is shown in Fig. 2.