Maximizing Fisher Information Using Discrete Mechanics and Projection-Based Trajectory Optimization

Andrew D. Wilson and Todd D. Murphey

Abstract— **This paper reformulates an optimization algorithm previously presented in continuous-time to one using structured integration and structured linearization methods from discrete mechanics. The objective is to synthesize trajectories for dynamic robotic systems that improve the estimation of model parameters by using a metric on Fisher information in a nonlinear projection-based trajectory optimization algorithm. A simulation of a robot with a suspended double pendulum is used as an example system to illustrate the algorithm. Results from the simulation show that the change to a discrete mechanics formulation reduces the computation time by a factor of 19 when compared to the continuous algorithm while maintaining the same two orders of magnitude improvement in the Fisher information from the continuous-time formulation. Through the Cramer-Rao bound, the improvement in the Fisher information results in a maximum expected error reduction of the parameter estimates by up to a factor of 10^2.**

I. Introduction

When creating models of robotic systems, the need to estimate model parameters arises in widely ranging applications such as industrial manipulation [1], [2], motion planning and localization [3]–[5], and control [6], [7] in order to improve model accuracy with possibly unknown parameters such as inertias, damping ratios, and geometric quantities. Improving the accuracy of the parameters allows for better control and overall performance of the robot. Typically, a robot executes an experimental trajectory and measurements are taken and compared to the model predictions to improve the model parameter set. However, the choice of experimental trajectory itself may have a significant impact on the overall precision of the estimation algorithm. Ideally, a robot could automatically generate this experimental trajectory to give the best possible estimate of the parameter set given the current best guess of the model and parameter values.

This paper provides a discrete-time formulation of the continuous-time Fisher information maximization algorithm published in [8]. The Fisher Information Matrix (FIM) provides a best-case estimate of the estimator's performance given a set of measurements from a robot through the Cramer-Rao bound [9]. The algorithm uses a gradient-descent projection-based trajectory optimization algorithm [10] to find a trajectory that locally maximizes the information of an experimental trajectory to produce the best estimate of the model parameters.

The motivation for designing an estimation algorithm using discrete mechanics is two-fold. First, the continuous-time algorithm requires numerical solutions to several differential equations which provides flexibility in its implementation; however, continuous-time algorithms may result in long computation times depending on the numerical method that is used. Using a result of much recent work in the area of discrete mechanics, variational integrators, also known as *structured integrators* [11], can be used to provide a time-discretized version of the action integral which results in a symplectic form with stable long-term energy behavior, even with large time-steps [12]. These integrators have been applied to the simulation of complex mechanical systems using generalized coordinates with accuracy comparisons to traditional numerical integrators in [13].

Second, the continuous-time algorithm requires a continuous cost function which necessitates the approximation of Fisher information, defined as $\tilde{I}(\theta)$ in [8] as a continuous measure along the trajectory. This is an approximation which converges to the actual Fisher information as the measurement frequency increases. Formulating the optimization in discrete-time allows for a discrete cost, which includes the exact Fisher information of the trajectory.

An important step of the trajectory optimization algorithm is obtaining accurate linearizations of the nonlinear models for use in an LQ regulation problem along the robot's trajectory. Computing the linearizations can be difficult in discrete-time due to numerical instability [14]; however, we base our derivation on exact representations of the linearizations which have been obtained for arbitrary mechanical systems in [15].

A number of related works in the field of optimal experimental design have generated a variety of methods for nonlinear and linear systems [16]–[20]. An algorithm developed by Emery [21] presents a Fisher information maximization technique to create an optimal trajectory. Although this work and others [22] perform the optimization in discrete-time, the dynamics are not structurally discretized using a symplectic technique, which in turn does not guarantee stable long-term energy behavior.

The main contribution of this paper is the formulation of the FIM maximization algorithm in discrete-time using discrete mechanics for the modeling, linearization, and integration of the system. First-order sensitivities are computed for use in the LQ calculation at each iteration of the trajectory optimization algorithm. The first-order sensitivity has previously been used by Caldwell [7] in a least-squares parameter estimation experiment. While the formulation of the discrete-time algorithm preserves the same cost function and is expected to perform similarly in terms of the improvement

A. D. Wilson and T. D. 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

of information to the continuous algorithm, the results in Section V show a significant reduction in computational time due to structure discritization of time from using the discrete mechanics techniques.

The paper is organized as follows: Section II provides an overview of the discrete mechanics necessary to form the Discrete Euler-Lagrange (DEL) equations for the system. Section III presents the objective function and sensitivity equations used in the optimization algorithm. Section IV provides the formulation of the discrete-time trajectory optimization algorithm and Section V presents results of a robot cart-pendulum simulation.

II. Discrete Mechanical Systems

In this paper, we simulate mechanical systems in a discrete-time framework. Given a system with a configuration $q \in Q$, where Q is the configuration space, a sequence $\{(t_0, q_0), (t_1, q_1), ..., (t_n, q_n)\}$ can be found that approximates a trajectory in continuous time where $q_k \approx q(t_k)$. Instead of numerically integrating differential equations derived from a continuous Lagrangian, L_c, a discrete Lagrangian, L_d, is chosen such that the action integral is approximated over a discrete time-step

$$L_d(q_k, q_{k+1}, \theta) \approx \int_{t_k}^{t_{k+1}} L_c(q(s), \dot{q}(s), \theta)\mathrm{d}s.$$

The action integral can then be approximated by a discrete action sum given by

$$\int_{t_0}^{t_f} L_c(q(\tau), \dot{q}(\tau), \theta)d\tau \approx \sum_{k=0}^{k_f} L_d(q_k, q_{k+1}, \theta).$$

Given the discrete action sum, the DEL equations including forcing have been derived for the discrete-time system [23]. The unconstrained DEL equations are given by[1]

$$\begin{aligned} D_2L_d(q_{k-1}, q_k, \theta) + F_d^+(q_{k-1}, q_k, u_{k-1}, \theta) + \\ D_1L_d(q_k, q_{k+1}, \theta) + F_d^-(q_k, q_{k+1}, u_k, \theta) = 0. \end{aligned}$$

where $F_d^{\pm}$ are the right and left discrete force approximations of a continuous forcing model, F_c. For clarity, we will drop the arguments to the discrete Lagrangian and instead refer to the indexes using the following subscript notation.

$$\begin{aligned} L_{k+1} &= L_d(q_k, q_{k+1}, \theta) \\ F_{k+1}^{\pm} &= F_d^{\pm}(q_k, q_{k+1}, u_k, \theta) \end{aligned}$$

In order to create a suitable analog to the continuous time framework, the DEL equations can be rewritten in a one-step map using a discrete momentum term, p_k. The first equation is implicit in terms of q_{k+1} and is solved using a root-finding algorithm. The following is the one-step update of $\{q_k, p_{k+1}\}$

$$\begin{aligned} p_k + D_1L_{k+1} + F_{k+1}^- &= 0 \\ p_{k+1} &= D_2L_{k+1} + F_{k+1}^+ \end{aligned} \quad (1)$$

Therefore the discrete states, x_k are of the form

$$x_k = \begin{bmatrix} q_k \\ p_k \end{bmatrix}.$$

A. Variational Integrators

In order to solve the DEL equations, the discrete Lagrangian and discrete forcing functions must be specified. A variational integrator is created by specifying the approximation model used to convert the continuous-time Lagrangian to a discrete-time analog. For the scope of this paper, we will use the most common form with midpoint approximation. The discrete Lagrangian is approximated as:

$$L_d(q, q_{k+1}, \theta) = L_c\left(\frac{q_{k+1} + q_k}{2}, \frac{q_{k+1} - q_k}{\Delta t}, \theta\right) \quad (2)$$

The discrete forcing function is chosen using the same approximation resulting in the following discrete forces

$$\begin{aligned} F_d^-(q_k, q_{k+1}, u_k, \theta) &= f_c\left(\frac{q_{k+1} + q_k}{2}, \frac{q_{k+1} - q_k}{\Delta t}, u_k\right)\Delta t \\ F_d^+(q_k, q_{k+1}, u_k, \theta) &= 0 \end{aligned}$$

Since the right discrete force is zero using this approximation method, the derivations and equations that follow will not include the right discrete force term for simplicity; however, if a different approximation method is used, this term may need to be added to the derivations.

B. Kinematic Configuration Variables

In the example provided, we employ the use of kinematic configuration variables when simulating and optimizing the system. For certain subsets of a robot model, it may be reasonable to assume a kinematic model where the actuators are strong enough to accurately realize any reasonable trajectory in the configuration space. For the cart-pendulum example, the position of the cart x will be treated as kinematic, *i.e.*, the input at any time t_k will be the position of the cart rather than the force upon the cart. While this separation is not required or essential, it tends to simplify implementation in practice by allowing a higher frequency, low-level controller to provide position control for the kinematic states while running a lower frequency controller on the dynamic states.

To implement the kinematic configuration variables, new states are added for the kinematic components. Thus, the state becomes

$$x_k = \begin{bmatrix} q_k \\ \rho_k \\ p_k \\ \nu_k \end{bmatrix},$$

where $\rho_k \in \mathbb{R}^{nk}$ is the set of kinematic configuration variables and $\nu_k \in \mathbb{R}^{nk}$ is the set of kinematic velocities. The discrete Lagrangian and discrete forces are expanded to include these kinematic configurations; however, the solution to the equation remains the same since the variables are predefined as inputs.

$$\begin{aligned} L_{k+1} &= L_d(q_k, q_{k+1}, \rho_k, \rho_{k+1}, \theta) \\ F_{k+1}^{\pm} &= F_d^{\pm}(q_k, q_{k+1}, \rho_k, \rho_{k+1}, u_k, \theta) \end{aligned}$$

[1] The slot derivative notation, $D_\kappa\alpha(x, y, z)$ represents the partial derivative of α w.r.t the κ^{th} argument.

The input vector is then defined as the force inputs u_k at the current time t_k and the kinematic configuration at the next time-step ρ_{k+1} which will be notated as

$$\bar{u}_k = \left[\begin{array}{c} u_k \\ \rho_{k+1} \end{array} \right].$$

III. Problem Formulation

For this algorithm, we consider the estimation of a set of model parameters in a system which is subject to measurement noise but negligible process noise. Unlike the previous work, both the parameter estimation and information maximization are done in a discrete-time setting. The system model is represented by the discrete update map (1) with the output y_k given by

$$y_k = g(x_k, \bar{u}_k, \theta) + \mathrm{w}_y.$$

$x_k \in \mathbb{R}^{2n}$ defines the discrete system state, $y_k \in \mathbb{R}^h$ defines the measured output, $\bar{u}_k \in \mathbb{R}^{m+nk}$ defines the inputs to the system, $\theta \in \mathbb{R}^s$ defines the set of model parameters to be estimated, and w_y is additive output noise where $p(\mathrm{w}_y) = N(0, \Sigma)$.

A. Parameter Estimation

Given the assumption of normally distributed measurement noise on y_k, the least squares estimator is equivalent to a maximum likelihood estimator which can be written as

$$\hat{\theta} = \arg\min_{\theta} \beta(\theta) \tag{3}$$

where

$$\beta(\theta) = \frac{1}{2} \sum_{k}^{k_f} (\tilde{y_k} - y_k)^T \cdot \Sigma^{-1} \cdot (\tilde{y_k} - y_k).$$

$\tilde{y_k}$ is the observed state at the k^{th} index of t_f/dt measurements, $\Sigma \in R^{h \times h}$ is the covariance matrix associated with the sensor measurement error, and $\hat{\theta}$ is the least-squares estimate of the parameter set.

Given the estimator, we may use a gradient descent method to find the optimal set of parameters, $\hat{\theta}$. Therefore, the first derivative of $\beta(\theta)$ must be calculated w.r.t θ.

$$D_\theta \beta(\theta) = \sum_{k}^{k_f} (\tilde{y}_k - y_k)^T \cdot \Sigma^{-1} \cdot \Gamma_k$$

where

$$\Gamma_k = D_1 g(x_k, \bar{u}_k, \theta) \cdot \frac{dx_k}{d\theta} + D_3 g(x_k, \bar{u}_k, \theta)$$

This requires the evaluation of $\frac{dx_k}{d\theta}$ which can be found using first-order linearizations of x_k from the DEL equations. Since ρ and ν are kinematic configurations, $\frac{d\rho_k}{d\theta}$ and $\frac{d\nu_k}{d\theta}$ are zero for all k. To match notation in the continuous time problem, we note that the sensitivity ψ_k is given by

$$\psi_k = \left[\begin{array}{c} \frac{dq_k}{d\theta} \\ \frac{dp_k}{d\theta} \end{array} \right].$$

The first-order sensitivity is thus given by

$$\psi_{k+1} = A_k \cdot \psi_k + \frac{\partial x_k}{\partial \theta} \tag{4}$$

where

$$\begin{aligned} \frac{\partial q_{k+1}}{\partial \theta} &= -M^{-1} \cdot (D_5 D_1 L_{k+1} + D_6 F^-_{k+1}) \\ \frac{\partial p_{k+1}}{\partial \theta} &= D_2^2 L_{k+1} \cdot \tfrac{\partial q_{k+1}}{\partial \theta} + D_5 D_2 L_{k+1} \\ A_k &= \left[\begin{array}{cc} \frac{\partial q_{k+1}}{\partial q_k} & \frac{\partial q_{k+1}}{\partial p_k} \\ \frac{\partial p_{k+1}}{\partial q_k} & \frac{\partial p_{k+1}}{\partial p_k} \end{array} \right]. \end{aligned}$$

We refer readers to [15] for the equations representing the components of A_k and M which are derived and presented in full.

B. Fisher Information

The FIM for this system is formulated in the same manner as the previous work. Assuming that the measurement noise of the system is normally distributed with zero process noise, the FIM is given by,

$$I(\theta) = \sum_{k=k_0}^{k_f} \Gamma_k^T \cdot \Sigma^{-1} \cdot \Gamma_k.$$

A lower bound on the precision of the parameter estimate returned by the least-squares estimator can be quantified by the Cramer-Rao bound given by

$$cov_\theta(\hat{\theta}) \geq I(\theta)^{-1}$$

where $\hat{\theta}$ is the least-squares estimator defined in (3).

IV. Discrete-Time Trajectory Optimization

The same trajectory optimization algorithm is used in the results to follow as the continuous-time case; however, the algorithm itself has been reformulated in [24] into a discrete-time framework. This discrete-time version is extended in the following section to include Fisher information as an additional objective.

A. Objective Function

The objective function will consist of three components: the Fisher information cost, a trajectory tracking cost, and a control cost. Since the optimization algorithm is formated in discrete-time, the Fisher information can be directly used in the algorithm versus the continuous-time formulation presented in the previous work. Since the FIM is a matrix quantity, we use E-optimality, which uses the minimum eigenvalue of the FIM as the cost metric.

The optimization objective function is defined as

$$\begin{aligned} J = \frac{Q_p}{\lambda_{min}} + \sum_{k=k_0}^{k_f} \big[&(x_k - \hat{x}_k)^T \cdot Q_\tau \cdot (x_k - \hat{x}_k) \\ &+ (\bar{u}_k - \hat{\bar{u}}_k)^T \cdot R_\tau \cdot (\bar{u}_k - \hat{\bar{u}}_k) \big] \end{aligned} \tag{5}$$

where λ_{min} is the minimum eigenvalue of $I(\theta)$, Q_p is the information weight, $\hat{x}_k$ is a reference trajectory, $\hat{\bar{u}}_k$ is a reference control signal, Q_τ is a trajectory tracking weighting matrix, and R_τ is a control effort weighting matrix. The weights must be chosen such that $Q_p \geq 0$, Q_τ is positive semi-definite, and R_τ is positive definite.

The various weights allow for design choices in the optimal trajectory that is obtained. The requirements of positive definiteness and positive semi-definiteness of the weighting matrices are necessary to maintain a locally convex optimization problem including the fact that $\lambda_{min} \geq 0$ [10]. Increasing the control weight will result in less aggressive trajectories, generally decreasing the obtained information. Using a reference trajectory allows for an optimal solution that remains in the neighborhood of a known trajectory.

B. Extended Dynamic Constraints

To extend the optimal control algorithm to include the FIM metric in (5), the first-order sensitivity ψ_k must be computed along the trajectory. In order to directly take variations on ψ_k, the term is treated as an additional state. Appending the parametric sensitivity to the state vector as an additional dynamic constraint allows for variations in ψ_k in the optimization algorithm. For convenience, the extended state will be defined by $\bar{x}_k = (x_k, \psi_k)$, and $\eta_k = (\bar{x}_k, \bar{u}_k)$ defines a curve that satisfies the nonlinear system dynamics.

C. Projection Operator

In the same manner as the continuous-time trajectory optimization problem, the minimization of (5) is subject to constraints from the dynamics and sensitivity given by (1) and (4). Instead of attempting to directly solve this nonlinear constrained optimization problem, the constraints are linearized and an LQ problem is solved iteratively to produce a descent direction followed by a projection that maps the infeasible trajectory, formed by the sum of the current iterate and the descent direction, onto the nonlinear constraints as detailed in [10]. The projection operator uses a stabilizing feedback law to take a feasible or infeasible trajectory, defined by $\xi_k = (\bar{\alpha}_k, \bar{\mu}_k)$, and maps it to a feasible trajectory, $\eta_k = (\bar{x}_k, \bar{u}_k)$.

The discrete form of the projection operator used in this paper is given by

$$P(\xi_k) : \begin{cases} \bar{u}_k = \bar{\mu}_k - K_k(\bar{x}_k - \bar{\alpha}_k) \\ x_{k+1} = \text{solution to (1)} \\ \psi_{k+1} = \text{equation (4)} \end{cases}$$

The feedback gain K_k can be optimized as well by solving an additional linear quadratic regulation problem. Details of the optimal gain problem can be found in [10], but any stabilizing feedback may be used.

D. Optimization Algorithm

Algorithm 1 defines the iterative method using a gradient descent approach to solve the optimization problem. Each iteration requires a descent direction, $\zeta_k = (\bar{z}_k, \bar{v}_k)$ to be computed from the following equation [10]:

$$\zeta_k = \arg\min_{\zeta_k} DJ(P(\xi_k)) \circ \zeta_k + \frac{1}{2}\langle \zeta_k, \zeta_k \rangle \tag{6}$$

such that

$$\bar{z}_{k+1} = \bar{A}_k \bar{z}_k + \bar{B}_k \bar{v}_k$$

where $\zeta_k \in T_{\eta_k}\mathcal{T}$, i.e., the descent direction for each iteration lies in the tangent space of the trajectory manifold at η_k. The components of the descent direction $\zeta_k = (\bar{z}_k, \bar{v}_k)$ are defined by $\bar{z}_k$, the perturbation to the extended state, and $\bar{v}_k$, the perturbation to the control. The quantity $\langle \zeta_k, \zeta_k \rangle$ is a local quadratic model computed as an inner product of ζ_k. Matrices $\bar{A}_k$ and $\bar{B}_k$ are the linearizations of the extended state dynamics. The descent direction (6) can be computed by solving an LQ regulation problem which is detailed in the following section.

Algorithm 1 Trajectory Optimization

```
Initialize $\eta_k^0 \in \mathcal{T}$, tolerance $\epsilon$, $i = 0$
while $DJ(\eta_k^i) \circ \zeta^i > \epsilon$ do
    Calculate descent, $\zeta^i$:
        $\zeta^i = \arg\min_{\zeta^i} DJ(P(\xi^i)) \circ \zeta^i + \frac{1}{2}\langle \zeta^i, \zeta^i \rangle$
            minimizing (7) to compute $\zeta^i$
    Compute $\gamma^i$ with Armijo backtracking search
    Calculate the infeasible step:
        $\xi_k^{i+1}(t) = \eta_k^i + \gamma^i \zeta_k^i$
    Project trajectory onto dynamics constraints:
        $\eta_k^{i+1} = P(\xi_k^{i+1})$
    $i = i + 1$
end while
```

Given the descent direction ζ^i, a backtracking line-search of the projection, $P(\eta^i + \gamma^i \zeta^i)$, provides a feasible trajectory assuming the step size γ^i satisfies the Armijo sufficient decrease condition [25]. Iterations upon the feasible trajectories continue until a given termination criteria is achieved.

E. Calculating the Descent Direction, ζ_k

To find a descent direction for the optimal control algorithm, (6) must be solved. As shown in (6), the descent direction depends on the linearization of the cost function, $DJ(P(\xi_k))$, and the local quadratic model, $\langle \zeta_k, \zeta_k \rangle$. Using a quadratic model and expanding the linearizations of the cost function, (6) is rewritten as

$$\begin{aligned} \arg\min_{\zeta_k} = \sum_{k=k_0}^{k_f} & 2a_k^T \bar{z}_k + 2b_k^T \bar{v}_k + \bar{z}_k^T Q_n \bar{z}_k \\ & + \bar{v}_k^T R_n \bar{v}_k \; dt \end{aligned} \tag{7}$$

such that

$$\bar{z}_{k+1} = \bar{A}\bar{z}_k + B\bar{v}_k$$

where a_k and b_k are the linearizations of the cost function with respect to $\bar{x}$ and $\bar{u}$, and Q_n and R_n are weighting matrices for the local quadratic model approximation. Design of these weighting matrices can lead to faster convergence of the optimal control algorithm depending on the specific problem. Since the derivations of the cost function linearizations follows the same steps as the continuous-time formulation, we refer reader to [8] for the derivations. We present the equations here in discrete-time for reference.

$$
\begin{aligned}
a_k =& \left[\begin{array}{c} (x_k - \hat{x}_k)^T Q_\tau \\ \{0\}^{1\times n\times p} \end{array} \right] - \\
& \frac{Q_p}{\lambda_{min}^2} \omega_s^T \left[\begin{array}{c} 2\,\Gamma_k^T \Sigma_k^{-1} \left(\frac{\partial^2 g}{\partial x_k^2} \psi_k + \frac{\partial^2 g}{\partial\theta\partial x_k} \right) \\ 2\,\Gamma_k^T \Sigma_k^{-1} \frac{\partial g}{\partial x_k} E \end{array} \right] \nu_s \\
b_k =& (\bar{u}_k - \hat{\bar{u}}_k)^T \cdot R_\tau
\end{aligned}
$$

$$
\begin{aligned}
\bar{A}_k =& \left[\begin{array}{cc} \frac{\partial x_{k+1}}{\partial x_k} & \frac{\partial x_{k+1}}{\partial \psi_k} \\ \frac{\partial \psi_{k+1}}{\partial x_k} & \frac{\partial \psi_{k+1}}{\partial \psi_k} \end{array} \right] \\
=& \left[\begin{array}{cc} A_k & \{0\}^{n\times n\times p} \\ \frac{\partial^2 x_{k+1}}{\partial x_k^2} \cdot \psi_k + \frac{\partial^2 x_{k+1}}{\partial\theta\partial x_k} & A_k \cdot E \end{array} \right] \qquad (8) \\
\bar{B}_k =& \left[\begin{array}{c} \frac{\partial x_{k+1}}{\partial \bar{u}_k} \\ \frac{\partial \psi_{k+1}}{\partial \bar{u}_k} \end{array} \right] \\
=& \left[\begin{array}{c} B_k \\ \frac{\partial^2 x_{k+1}}{\partial x_k \partial \bar{u}_k} \cdot \psi_k + \frac{\partial^2 x_{k+1}}{\partial\theta\partial \bar{u}_k} \end{array} \right] \qquad (9)
\end{aligned}
$$

where ω_s and ν_s are the left and right eigenvectors of $I(\theta)$ respectively and definitions of $\frac{\partial^2 x_{k+1}}{\partial\theta\partial x_k}$ and $\frac{\partial^2 x_{k+1}}{\partial\theta\partial \bar{u}_k}$ are available in the Appendix. E is a tensor of the form $E_{ijkl} = \delta_{ik}\delta_{jl}$ where δ as the Kronecker delta function.

After computing the linearization of the cost function and dynamics at each time step along the trajectory, (6) can be solved using the discrete-time algebraic Riccati equation. The formulation of this LQ problem is detailed in the appendix of [24].

V. Results

To examine the results of the discrete-time formulation, the algorithm is tested in simulation on a 2-link cart-pendulum robot. The robot has two dynamic configuration variables, $q = (\phi_1(t_k), \phi_2(t_k))$, and one kinematic configuration variable, $\rho = x(t_k)$, where $x(t_k)$ is the horizontal displacement of the robot, and $\phi_i(t_k)$ is the rotational angle of each link as shown in Fig. 1.

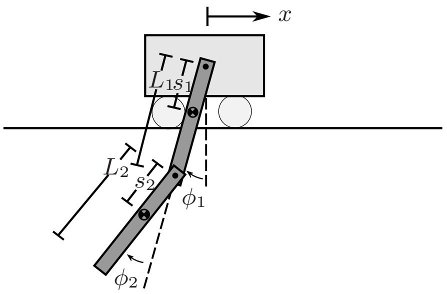
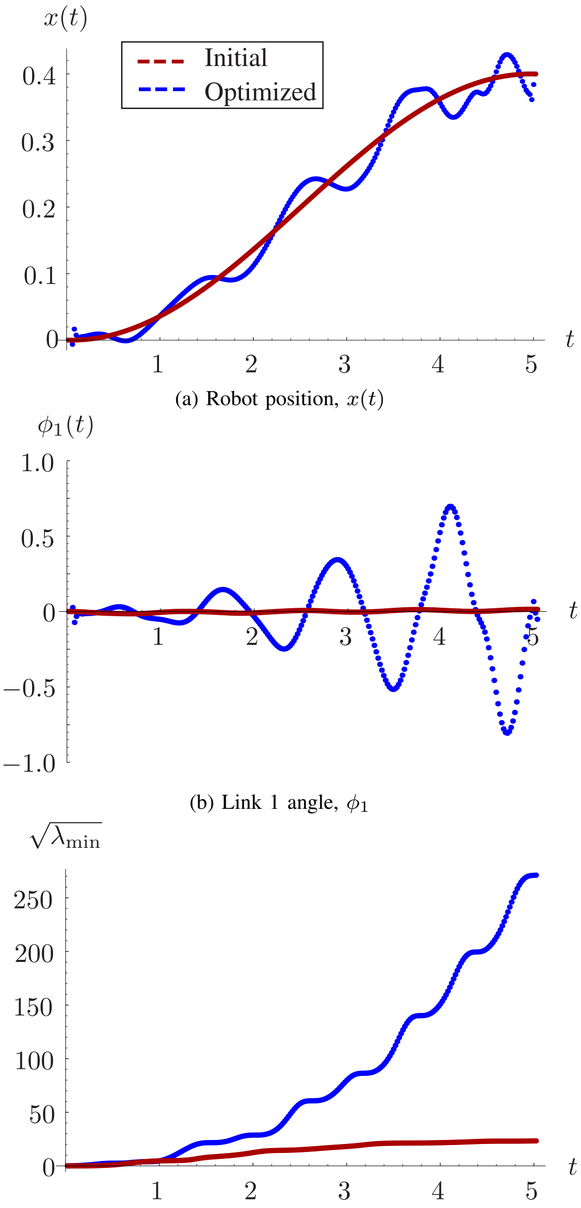


Fig. 1: Diagram of the cart-pendulum robot.

As indicated in the section on kinematic configuration variables, the input to the robot is the position $x(t_k)$. The robot can move in one dimension with positive motion to the right. Rotational friction is modeled at each pendulum joint, but the joints remain unactuated. For comparision to the continuous-time results, the goal of the optimization algorithm and the simulated model remains the same as in [8] - estimating the mass of the top pendulum link m_1 and the damping coefficient c of the joints.



(a) Robot position, $x(t)$

(b) Link 1 angle, ϕ_1

(c) Square root of the minimum eigenvalue of the Fisher information matrix $I(\theta)$ over time.

Fig. 2: Comparisons of the trajectory before and after Fisher information optimization.

A. Discrete-time Model Approximation

An approximation of the continuous Lagrangian in discrete time is found using (2). The same values for model parameters are used as defined in [8]. We will assume that sensor measurements occur at 50 Hz, which will also be used as the simulation time-step, dt. In general, measurements may occur at a lower frequency than the simulation time-step. In that case, Σ_k^{-1} should be set to zero for all time-steps k in which measurements do not occur.

B. Optimization Results

The optimization algorithm was run until a convergence criterion of $|DJ(\xi_k) \circ \zeta| < 10^{-2}$ was satisfied, starting from an initial cost of 23.0. A comparison of initial and optimized trajectories can be seen in Fig. 2a-b. The optimized trajectory improves the Fisher information of the measured trajectory by a factor of 10^2. These results can be seen in Table I. Additionally, the Fisher information gained over time is shown in Fig. 2c. The optimized trajectory greatly exceeds the rate of information gained compared to the initial trajectory. Using the Cramer-Rao bound [26], error estimates in the parameter set will decrease up to a factor of 10^2 due to this increase in Fisher information.

One motivation for creating a discrete-time version of the information maximization algorithm is to reduce computation time compared to the continuous version. Since the structured integration allows for stable results at large time-steps, such as 0.02 s used for this results, the computation time for each iteration is reduced by a factor of 19 as shown in Table II.

While these results show the discrete-time algorithm converging in fewer iterations, we do not expect this trend to be generally true. In this case, the numerical differences happen to produce a slightly more efficient descent direction. However, even on a per iteration basis, the improvement in computation time is significant. Iterations of the continuous and discrete algorithms were timed in Mathematica on the same 3.0 GHz Intel i7 machine. Both algorithms return similar optimized costs, resulting in expected information improvements of 10^2.

Another concern may be that the resulting trajectory of the discrete algorithm does not precisely track the continuous trajectory. While the example system and model parameters are the same for both sets of results, we have incorporated the notion of kinematic control inputs into the discrete algorithm. This results in a different set of states compared to the continuous algorithm and different weights, Q_τ and R_τ. Additionally, the formulation in discrete time results in discrete momenta states p_k rather than velocity states. This also modifies the local quadratic approximation of the cost function, J_τ. Finally, the discrete algorithm uses the exact Fisher information, which slightly modifies the cost compared to the continuous approximation. Therefore, we expect that the algorithms will both achieve the goal of maximizing Fisher information but under slightly different objectives due to these differences which results in the varying trajectories.

TABLE I: Optimization Results

	λ_1	λ_2	J
Initial:	5.95×10^3	543.2	23.0
Optimized:	6.91×10^6	7.34×10^4	9.0
- - - - - -	- -		
	Fisher Information Matrix		
Initial:	$\begin{bmatrix} 5.94 \times 10^3 & 98.6 \\ 98.6 & 544.9 \end{bmatrix}$		
Optimized:	$\begin{bmatrix} 6.90 \times 10^6 & 2.79 \times 10^3 \\ 2.79 \times 10^3 & 8.48 \times 10^4 \end{bmatrix}$		

TABLE II: Execution Time

	Discrete-Time	Continuous-Time
Avg. Time per iteration:	39.9 s	762.5 s
Number of Iterations:	2	36
Min. Eigenvalue Improvement:	factor of 135	factor of 110

VI. Conclusion and Future Work

This paper presented a discrete-time method to automatically maximizing the Fisher information with respect to a set of model parameters by optimizing the robot's trajectory. We compare the method and results to the continuous-time analog published in [8]. In both cases, the algorithms significantly result in higher information which improves the estimation of uncertain parameters within the system model. The discrete formulation presented in the paper results in a factor of 19 improvement in the computation speed per iteration over the continuous algorithm while maintaining similar performance on the objective function. The algorithm also incorporates the actual Fisher information given a discrete set of measurements along the robot's trajectory rather than the approximation used in the continuous-time algorithm.

One question that remains for future work is how to exactly map weights such as Q_τ and R_τ from a continuous-time formulation to the discrete-time domain. An exact mapping may allow for more similar trajectories between the two algorithms if desired. This is particularly important if the trajectory tracking is more important than the information maximization.

While computational speed has been significantly improved, expanding the dimensionality of the system still poses a formidable problem in terms of computation time. While the algorithm applies to larger systems, the need for extended state representations significantly affects general scalability. There remains a need to investigate alternative trajectory optimization methods or sensitivity representation that may eventually allow for real-time application of the algorithm.

Nonetheless, the improvement in computational efficiency using structured integration and structured linearization provides more opportunities to use a information maximization algorithm on a robot or embedded system where decisions and trajectory planning need to be made in a matter of minutes rather than hours. This also may allow for use in a learning type algorithm in the future where more iterations of the algorithm can be performed on-line as information is gained from previous trajectory trials facilitating even greater performance of the estimator and nonlinear controllers.

Acknowledgment

This material is based upon work supported by the National Institute of Health under NIH R01 EB011615 and by the National Science Foundation under Grant CMMI 1334609. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Appendix I
Descent direction derivatives

In order to compute the linearization of the extended state dynamics, $\frac{\partial^2 x_{k+1}}{\partial\theta\partial x_k}$ and $\frac{\partial^2 x_{k+1}}{\partial\theta\partial\bar{u}_k}$ are required to compute (8) and (9).

$\frac{\partial^2 x_{k+1}}{\partial\theta\partial x_k}$ is constructed from the following components

$$
\begin{aligned}
\frac{\partial^2 q_{k+1}}{\partial\theta\partial q_k} =& - M^{-1}[(D_2D_5D_1L_{k+1} + D_2D_6F^-_{k+1})\frac{\partial q_{k+1}}{q_k} + \\
& D_1D_5D_1L_{k+1} + D_1D_6F^-_{k+1}] \\
\frac{\partial^2 q_{k+1}}{\partial\theta\partial p_k} =& - M^{-1}[(D_2D_5D_1L_{k+1} + D_2D_6F^-_{k+1})\frac{\partial q_{k+1}}{p_k}] \\
\frac{\partial^2 p_{k+1}}{\partial\theta\partial q_k} =& (D_2D_2D_2L_{k+1}\frac{\partial q_{k+1}}{\partial q_k} + D_1D_2D_2L_{k+1})\frac{\partial q_{k+1}}{\partial\theta} \\
& + D_2D_2L_{k+1}\frac{\partial^2 q_{k+1}}{\partial\theta\partial q_k} + D_2D_5D_2L_{k+1}\frac{\partial q_{k+1}}{\partial q_k} \\
& + D_1D_5D_2L_{k+1} \\
\frac{\partial^2 p_{k+1}}{\partial\theta\partial p_k} =& (D_2D_2D_2L_{k+1}\frac{\partial q_{k+1}}{\partial p_k})\frac{\partial q_{k+1}}{\partial\theta} \\
& + D_2D_2L_{k+1}\frac{\partial q^2_{k+1}}{\partial\theta\partial p_k} + D_2D_5D_2L_{k+1}\frac{\partial q_{k+1}}{\partial p_k}
\end{aligned}
$$

and $\frac{\partial^2 x_{k+1}}{\partial\theta\partial\bar{u}_k}$ is constructed from the following components

$$
\begin{aligned}
\frac{\partial^2 q_{k+1}}{\partial\theta\partial u_k} =& - M^{-1}[(D_2D_5D_1L_{k+1} + D_2D_6F^-_{k+1})\frac{\partial q_{k+1}}{\partial u_k} \\
& + D_5D_6F^-_{k+1}] \\
\frac{\partial^2 q_{k+1}}{\partial\theta\partial \rho_{k+1}} =& - M^{-1}[(D_2D_5D_1L_{k+1} + D_2D_6F^-_{k+1})\frac{\partial q_{k+1}}{\partial \rho_{k+1}} \\
& + D_4D_5D_1L_{k+1} + D_4D_6F^-_{k+1}] \\
\frac{\partial^2 p_{k+1}}{\partial\theta\partial u_k} =& (D_2D_2D_2L_{k_1}\frac{\partial q_{k+1}}{\partial u_k})\frac{\partial q_{k+1}}{\partial\theta} \\
& + D_2D_2L_{k+1}\frac{\partial q^2_{k+1}}{\partial\theta\partial u_k} + D_2D_5D_2L_{k+1}\frac{\partial q_{k+1}}{\partial u_k} \\
\frac{\partial^2 p_{k+1}}{\partial\theta\partial \rho_{k+1}} =& (D_2D_2D_2L_{k_1}\frac{\partial q_{k+1}}{\partial \rho_{k+1}})\frac{\partial q_{k+1}}{\partial\theta} \\
& + D_2D_2L_{k+1}\frac{\partial q^2_{k+1}}{\partial\theta\partial \rho_{k+1}} + D_4D_5D_2L_{k+1} \\
& + D_2D_5D_2L_{k+1}\frac{\partial q_{k+1}}{\partial \rho_{k+1}}.
\end{aligned}
$$

References

[1] A. Kinsheel and Z. Taha, "Identification of the parameters of robot manipulators dynamics about an operating point using perturbed dynamics," in *2010 International Conference on Control Automation Robotics & Vision*, pp. 144–148, Dec. 2010.

[2] M. Gautier and S. Briot, "Dynamic parameter identification of a 6 DOF industrial robot using power model," in *2013 IEEE International Conference on Robotics and Automation*, pp. 2914–2920, May 2013.

[3] J. Muller, C. Gonsior, and W. Burgard, "Improved Monte Carlo localization of autonomous robots through simultaneous estimation of motion model parameters," in *2010 IEEE International Conference on Robotics and Automation*, pp. 2604–2609, May 2010.

[4] D. J. Webb, K. L. Crandall, and J. van den Berg, "Online parameter estimation via real-time replanning of continuous Gaussian POMDPs," in *2014 IEEE International Conference on Robotics and Automation*, pp. 5998–6005, May 2014.

[5] J. A. Delmerico, P. David, and J. J. Corso, "Building facade detection, segmentation, and parameter estimation for mobile robot localization and guidance," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1632–1639, Sept. 2011.

[6] B. S. Park, S. J. Yoo, J. B. Park, and Y. H. Choi, "A Simple Adaptive Control Approach for Trajectory Tracking of Electrically Driven Nonholonomic Mobile Robots," *IEEE Transactions on Control Systems Technology*, vol. 18, pp. 1199–1206, Sept. 2010.

[7] T. M. Caldwell, D. Coleman, and N. Correll, "Optimal Parameter Identification for Discrete Mechanical Systems with Application to Flexible Object Manipulation," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014.

[8] A. D. Wilson, J. A. Schultz, and T. D. Murphey, "Trajectory Synthesis for Fisher Information Maximization," *IEEE Transactions on Robotics*, vol. 30, no. 6, pp. 1358–1370, 2014.

[9] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*. John Wiley & Sons, 2004.

[10] J. Hauser, "A projection operator approach to the optimization of trajectory functionals," in *World Congress*, vol. 15, p. 310, July 2002.

[11] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration*, vol. 31 of *Springer Series in Computational Mathematics*. Berlin/Heidelberg: Springer-Verlag, 2nd ed., 2006.

[12] A. Lew, J. E. Marsden, M. Ortiz, and M. West, "Variational time integrators," *International Journal for Numerical Methods in Engineering*, vol. 60, pp. 153–212, May 2004.

[13] E. Johnson and T. Murphey, "Scalable Variational Integrators for Constrained Mechanical Systems in Generalized Coordinates," *IEEE Transactions on Robotics*, vol. 25, pp. 1249–1261, Dec. 2009.

[14] A. J. Laub, R. V. Patel, and P. M. Van Dooren, "Numerical and Computational Issues in Linear Control and System Theory," in *The Control Systems Handbook*, Boca Raton, FL, USA: CRC Press, 2nd ed., 2010.

[15] E. Johnson, J. Schultz, and T. Murphey, "Structured Linearization of Discrete Mechanical Systems for Analysis and Optimal Control," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 1, pp. 140–152, 2015.

[16] M. Gautier and W. Khalil, "Exciting Trajectories for the Identification of Base Inertial Parameters of Robots," *The International Journal of Robotics Research*, vol. 11, pp. 362–375, Aug. 1992.

[17] J. Swevers, C. Ganseman, D. Tukel, J. de Schutter, and H. Van Brussel, "Optimal robot excitation and identification," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 5, pp. 730–740, 1997.

[18] W. Wu, S. Zhu, X. Wang, and H. Liu, "Closed-loop Dynamic Parameter Identification of Robot Manipulators Using Modified Fourier Series," *International Journal of Advanced Robotic Systems*, May 2012.

[19] W. Rackl, R. Lampariello, and G. Hirzinger, "Robot excitation trajectories for dynamic parameter estimation using optimized B-splines," in *2012 IEEE International Conference on Robotics and Automation*, pp. 2042–2047, May 2012.

[20] Yunjin Gu, Hyuk Wang, Jang-Ho Cho, and Doo Yong Lee, "Identification of dynamic parameters of an industrial robot using a recursively-optimized trajectory," in *2010 International Conference on Control Automation and Systems*, pp. 1450–1455, 2010.

[21] A. F. Emery and A. V. Nenarokomov, "Optimal experiment design," *Measurement Science and Technology*, vol. 9, pp. 864–876, June 1998.

[22] T. L. Vincent, C. Novara, K. Hsu, and K. Poolla, "Input design for structured nonlinear system identification," *Automatica*, vol. 46, pp. 990–998, June 2010.

[23] J. E. Marsden and M. West, "Discrete mechanics and variational integrators," *Acta Numerica 2001*, vol. 10, pp. 357–514, May 2001.

[24] E. Johnson, *Trajectory Optimization and Regulation for Constrained Discrete Mechanical Systems*. PhD thesis, Northwestern University, 2012.

[25] L. Armijo, "Minimization of functions having Lipschitz continuous first partial derivatives.," *Pacific Journal of Mathematics*, vol. 16, no. 1, pp. 1–3, 1966.

[26] C. R. Rao and J. Wishart, "Minimum variance and the estimation of several parameters," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 43, p. 280, Oct. 1947.