IMA Conference on Mathematics of Robotics
9 – 11 September 2015, St Anne's College, University of Oxford

Stochastic Variational Integrators for System Propagation and Linearization

By Gerardo De La Torre and Evangelos A. Theodorou

Autonomous Control and Decision Systems Laboratory
The Daniel Guggenheim School of Aerospace Engineering,
Georgia Institute of Technology, Atlanta, Georgia, U.S.

Abstract

In this paper we present a stochastic variational integrator and its linearization. In order to motivate the use of the proposed variational integrator the Stochastic Differential Dynamical Programming (S-DDP) algorithm is considered as a benchmark for comparison. Specifically, we are interested in investigating if it is advantageous to utilize the variational integrator to propagate system trajectories and linearize system dynamics. Through numerical experiments we show that the Stochastic Differential Dynamical Programming algorithm becomes less dependent on the discretization time step and more predictable when it utilizes the proposed integrator. Furthermore, we show that a significant reduction in computational time can be achieved without sacrificing algorithm performance. Therefore, the proposed variational integrator can be used to enable real-time implementation of nonlinear optimal control algorithms.

1. Stochastic Variational Integrators and Structured Linearization

This section presents a stochastic variational integrator used to obtain accurate discretized trajectories of stochastic dynamical Hamiltonian systems represented as

$$\mathrm{d}\frac{\partial \mathcal{L}(q,\dot{q})}{\partial \dot{q}} = \frac{\partial \mathcal{L}(q,\dot{q})}{\partial q}\mathrm{d}t + F_0(q,\dot{q},u)\mathrm{d}t + \sum_{i=1}^{p} F_i(q,u) \circledS \, \mathrm{d}\omega_i, q(0) = q_0, \dot{q}(0) = \dot{q}_0, \quad (1.1)$$

where $\mathcal{L}(q,\dot{q})$ is the mechanical system's Lagrangian given as the difference between the system's kinetic energy, $T(q,\dot{q})$, and potential energy, $V(q)$, such that $\mathcal{L}(q,\dot{q}) = T(q,\dot{q}) - V(q)$, q is the state configuration vector, $\dot{q}$ is the time derivative of the state configuration vector, $\mathrm{d}\omega_i,\ i = 1,\ldots,m$ are Brownian noises, $F_i(q,u),\ i = 1,\ldots,m$ are the diffusion vector fields, $F_0(q,\dot{q},u)$ is the forcing function representing deterministic non-conservative external forces, and $\circledS$ indicates the stochastic integral is evaluated in the Stratonovich sense Kloeden & Platen (1992). Note equation (1.1) provides the fundamental characteristics of a dynamical system and describes how the system configuration vector propagates through time. However, it does not provide any method or algorithm that can be used to solve for the trajectory of the system. Simply using numerical integration schemes developed for general second order stochastic differential equations will result in numerical errors since the system's fundamental characteristics are ignored.

A variational integrator computes a discretized trajectory $\mathbf{q} = \{q_0,\ldots,q_N\}$ that approximates the systems trajectory $q(t)$ such that $q_k \approx q(t_k)$, where $t_0 = 0$, $t_N = t_\mathrm{f}$ and $t_{k+1} - t_k = \Delta t$. The derivation of variational integrators relies on the discretization of the classical Hamilton's variational principle and, in forced systems, the Lagrange-d'Alembert principle. As a result, variational integrators are able to ensure (or strongly enforce) the conservation of fundamental quantities such as momentum and energy Marsden & West (2001). Furthermore, since the derivation begins with the classical Hamilton's variational principle the dynamical system's fundamental characteristics are implicitly considered. A complete and rigorous treatment of variational integrators was done by Hairer *et. al.* (2004) and Marsden & West (2001). The presented representation of a stochastic Hamiltonian system and a derivation of a variation integrator was presented by Bou-Rabee *et. al.* (2008). Stochastic variational integrators have been investigated in the context of electrical circuits by Ober-Blöbaum *et. al.* (2013), multi-scale mechanical systems by Tao *et. al.* (2010), and Bou-Rabee & Owhadi (2008) studies their long-run accuracy. Scalable variational integrators were implemented to generic mechanical systems in generalized coordinates by Johnson & Murphey (2009). First- and second-order linearizations of a deterministic variational integrator was formulated by Johnson *et. al.* (2015). The presented stochastic variational integrator and its first-order linearization are based on the deterministic variational integrator and its linearization reported in Johnson & Murphey (2009) and Johnson *et. al.* (2015).

1.1. *Stochastic Variational Integrators*

To being our review, the discrete Lagrangian, $L_\mathrm{d}(q_k, q_{k+1})$, is introduced to obtain an approximation of the action integral over a short interval, $L_\mathrm{d}(q_k, q_{k+1}) \approx \int_{t_k}^{t_{k+1}} \mathcal{L}(q(s), \dot{q}(s))\, \mathrm{d}s$. We use the generalized midpoint approximation to formulate L_d as $L_\mathrm{d}(q_k, q_{k+1}) = \mathcal{L}\big((1-\alpha)q_k + \alpha q_{k+1}, \frac{q_{k+1}-q_k}{\Delta t}\big)\Delta t$, where $\alpha \in [0,1]$ and $\alpha = 1/2$ results in second order accuracy as discussed in West (2004). Left and right discrete forces, $F_\mathrm{d}^-(q_k, q_{k+1}, u_k)$ and $F_\mathrm{d}^+(q_k, q_{k+1}, u_k)$, approximate the non-conservative forces as

$$\int_{t_k}^{t_{k+1}} F_0(q(s), \dot{q}(s), u(s)) \cdot \delta q \,\mathrm{d}s \approx F_{0,\mathrm{d}}^-/2 \cdot \delta q_k + F_{0,\mathrm{d}}^+/2 \cdot \delta q_{k+1}, \tag{1.2}$$

and the Stratonovich integral used to represent the stochastic effects is converted to its equivalent Itō's representation (see Kloeden & Platen (1992)) and then approximated as

$$\int_{t_k}^{t_{k+1}} F_i(q, \dot{q}, u) \cdot \delta q \circledS \,\mathrm{d}\omega_i = \frac{1}{2}\int_{t_k}^{t_{k+1}} F_i'(q, \dot{q}, u) F_i(q, \dot{q}, u) \cdot \delta q \mathrm{d}t + \int_{t_k}^{t_{k+1}} F_i(q, u) \cdot \delta q \mathrm{d}\omega_i$$
$$\approx {F'}_{i,\mathrm{d}}^{-} F_{i,\mathrm{d}}^{-}/4 \cdot \delta q_k + {F'}_{i,\mathrm{d}}^{+} F_{i,\mathrm{d}}^{+}/4 \cdot \delta q_{k+1} + F_{i,d}^{s} \Delta\omega_{i_k} \cdot \delta q_k, \tag{1.3}$$

where $F_{0,\mathrm{d}}^{\pm}$, $F_{i,\mathrm{d}}^{\pm}$, and ${F'}_{i,\mathrm{d}}^{\pm}$ are evaluated according to $\mathcal{G}_\mathrm{d}^{\pm}(q_k, q_{k+1}, u_k) = \mathcal{G}\big((1-\alpha)q_k + \alpha q_{k+1}, \frac{q_{k+1}-q_k}{\Delta t}, u_k\big)\Delta t$, $F_{i,d}^s(q_k, \cdot, u_k) = F_i\big(q_k, u_k\big)$ (second argument purposely left blank for consistency with other terms), $u_k = u(t_k)$, and $\Delta\omega_{i_k} = \omega_i(t_{k+1}) - \omega_i(t_k) \sim \mathcal{N}(0, \Delta t)$. The discrete form of the Lagrange-d'Alembert principle can be used to relate the derived approximations as

$$\delta \sum_{k=0}^{N-1} L_{k+1} + \frac{1}{2}\sum_{k=0}^{N-1} \big(F_{k+1}^- \cdot \delta q_k + F_{k+1}^+ \cdot \delta q_{k+1}\big) + \sum_{k=0}^{N-1} F_{k+1}^s \Delta\omega_k \cdot \delta q_k = 0, \tag{1.4}$$

where $F_k^{\pm} = F_{0,\mathrm{d}}^{\pm}(q_{k-1}, q_k, u_{k-1}) + \sum_{i=1}^{p} {F'}_{i,\mathrm{d}}^{\pm}(q_{k-1}, q_k, u_{k-1}) F_{i,\mathrm{d}}^{\pm}(q_{k-1}, q_k, u_{k-1})/2$, $F_k^s = [F_{1,\mathrm{d}}^s(q_{k-1}, \cdot, u_k), \ldots, F_{p,\mathrm{d}}^s(q_{k-1}, \cdot, u_k)]$, $L_k = L_\mathrm{d}(q_{k-1}, q_k)$, and $\Delta\omega_k = [\Delta\omega_{1,k}, \ldots, \Delta\omega_{p,k}]^\mathrm{T}$. Solving equation (1.4) leads to a forced Discrete Euler-Lagrange (DEL) equation†

$$D_2 L_k + D_1 L_{k+1} + F_k^+ + F_{k+1}^- + F_{k+1}^s \Delta\omega_{i_k} = 0. \tag{1.5}$$

The integrator equation can now be defined as

$$f(q_{k+1}) = p_k + D_1 L_{k+1} + F_{k+1}^- + F_{k+1}^s \Delta\omega_{i_k}, \tag{1.6}$$

† We write $D_i G(Y_1, Y_2, \ldots)$ for the derivative of a function with respect to its i^{th} argument.

where, for computation convenience, $p_k = D_2L_k + F_k^+$. Note that p_k does not depend on q_{k+1} and, in the unforced case, p_k is the generalized momentum quantity conserved by the integrator as discussed in Johnson & Murphey (2009). Furthermore, the derivative of (1.6) with respect to q_{k+1} is $Df(q_{k+1}) = D_2D_1L_{k+1} + D_2F_{k+1}^-$. The integrator equation (1.6) is an implicit one-step mapping $(q_k, p_k, u_k, \Delta\omega_k) \to (q_{k+1}, p_{k+1})$ and is solved using Algorithm 1 as shown in Johnson & Murphey (2009). The selection of ϵ_{tol} not only determines the accuracy of the variational integrator, but also the computational time in implementation. Note that if q_0 and q_1 are known then $q_k,\ k \geq 1$ can be obtained by utilizing the presented integrator. The presented variational integrator is utilized by the Stochastic Differential Dynamical Programming algorithm to propagate the *expected* trajectory of the system (i.e. $\Delta\omega_{ik} = 0$). Though stochastic effects are not considered when propagating the system configuration the linearization utilized by the algorithm requires a stochastic representation.

Algorithm 1 Simple Root Finder

while $|f(q_{k+1})| > \epsilon_{\text{tol}}$ **do**
  $q_{k+1} \leftarrow q_{k+1} - Df^{-1}(q_{k+1}) \cdot f(q_{k+1})$
end while

1.2. *Stochastic Structured Linearization*

In this section the derived forced DEL equation (1.5) is used to obtain a first-order linearization of the discrete dynamics. The linearization of the system is utilized by the Stochastic Differential Dynamical Programming algorithm as shown in Algorithm 2. To begin, the linearization is given in the following form †

$$\begin{bmatrix} \delta q_{k+1} \\ \delta p_{k+1} \end{bmatrix} = \begin{bmatrix} \frac{\partial q_{k+1}}{\partial q_k} & \frac{\partial q_{k+1}}{\partial p_k} \\ \frac{\partial p_{k+1}}{\partial q_k} & \frac{\partial p_{k+1}}{\partial p_k} \end{bmatrix} \begin{bmatrix} \delta q_k \\ \delta p_k \end{bmatrix} + \begin{bmatrix} \frac{\partial q_{k+1}}{\partial u_k} \\ \frac{\partial p_{k+1}}{\partial u_k} \end{bmatrix} \delta u_k + \begin{bmatrix} \frac{\partial q_{k+1}}{\partial \Delta\omega_k} \\ \frac{\partial p_{k+1}}{\partial \Delta\omega_k} \end{bmatrix} \Delta\omega_k. \quad (1.7)$$

To begin the forced DEL equation is represented in its equivalent position-momentum form (see Johnson *et. al.* (2015) for further details)

$$p_k + D_1L_{k+1} + F_{k+1}^- + F_{k+1}^s \Delta\omega_k = 0, \quad (1.8)$$

$$p_{k+1} = D_2L_{k+1} + F_{k+1}^+. \quad (1.9)$$

Implicitly differentiating equation (1.8) with respect to q_k yields

$$\frac{\partial}{\partial q_k}[p_k + D_1L_{k+1} + F_{k+1}^- + F_{k+1}^s \Delta\omega_{i_k} = 0] \to$$

$$0 + D_1D_1L_{k+1} + D_2D_1L_{k+1}\frac{\partial q_{k+1}}{\partial q_k} + D_1F_{k+1}^- + D_2F_{k+1}^-\frac{\partial q_{k+1}}{\partial q_k} + D_1F_{k+1}^s\Delta\omega_{i_k} = 0$$

$$\frac{\partial q_{k+1}}{\partial q_k} = -M_{k+1}^{-1}[D_1D_1L_{k+1} + D_1F_{k+1}^- + D_1F_{k+1}^s\Delta\omega_{i_k}] \quad (1.10)$$

where $M_{k+1} = D_2D_1L_{k+1} + D_2F_{k+1}^-$, is assumed to be non-singular at q_k, p_k, and u_k. Repeating this procedure yields

$$\frac{\partial q_{k+1}}{\partial p_k} = -M_{k+1}^{-1},\quad \frac{\partial q_{k+1}}{\partial u_k} = -M_{k+1}^{-1}[D_3F_{k+1}^- + D_3F_{k+1}^s\Delta\omega_{i_k}],\quad \frac{\partial q_{k+1}}{\partial \Delta\omega_k} = -M_{k+1}^{-1}F_{k+1}^s.$$

† As discussed in Johnson & Murphey (2009), the one-step mapping (1.6) implicitly defines a function $g(x_k, u_k, \Delta\omega_k)$ such that $x_{k+1} = g(x_k, u_k, \Delta\omega_k)$ where $x_{k+1} = [p_{k+1}^{\mathrm{T}}, q_{k+1}^{\mathrm{T}}]^{\mathrm{T}}$. The linearization is equivalently defined as $\delta x_{k+1} = \nabla_{x_k} g(x_k, u_k, \Delta\omega_k)\delta x_k + \nabla_{u_k} g(x_k, u_k, \Delta\omega_k)\delta u_k + \nabla_{\Delta\omega} g(x_k, u_k, \Delta\omega_k)\Delta\omega_k$.

The remaining derivatives can be found by explicitly differentiating (1.9) such that

$$\frac{\partial p_{k+1}}{\partial q_k} = N_{k+1}\frac{\partial q_{k+1}}{\partial q_k} + D_1D_2L_{k+1} + D_1F^{+}_{k+1}, \qquad \frac{\partial p_{k+1}}{\partial p_k} = N_{k+1}\frac{\partial q_{k+1}}{\partial p_k},$$

$$\frac{\partial p_{k+1}}{\partial u_k} = N_{k+1}\frac{\partial q_{k+1}}{\partial u_k} + D_3F^{+}_{k+1}, \qquad \frac{\partial p_{k+1}}{\partial \Delta\omega_k} = N_{k+1}\frac{\partial q_{k+1}}{\partial \Delta\omega_k},$$

where $N_{k+1} = D_2D_2L_{k+1} + D_2F^{+}_{k+1}$. Note that q_{k+1} is needed in order to evaluate the derivatives and can by found by solving (1.6). Furthermore, note that, in general, the partial derivatives with respect to q_k and u_k are stochastic. However, these derivatives are affine in $\Delta\omega_k$ ($A + B\Delta\omega_k$ where A and B are deterministic). Therefore, the expectation of the linearization of the discrete dynamics (1.7) can be found. It should be noted that F^{s}_{k+1} was formulated such that it did not depend on q_{k+1}. This was possible since the Stratonovich integral was converted to its equivalent Itō's representation. Note that by invoking this equivalence, we replace an explicit dependence of q_{k+1} with a projection of stochastic effects on the system, captured by the term $\frac{1}{2}\int_{t_k}^{t_{k+1}} F_i' F_i \cdot \delta q \, \mathrm{d}t$. As a result, M_{k+1} is not stochastic and, therefore, ratios of stochastic quantities are avoided (e.g. $\frac{\partial q_{k+1}}{\partial q_k} = -M^{-1}_{k+1}[D_1D_1L_{k+1} + D_1F^{-}_{k+1} + D_1F^{s}_{k+1}\Delta\omega_k]$). Therefore, the expectation and central moments of the linearization are well-defined and computed easily. Note that the change in the representation avoids ratios of stochastic quantities, but this is not to say that using a Stratonovich representation in this case is ill-posed. Conversion between a Stratonovich and a Itō representation (or another representation) is often done when one definition is more convenient.

2. Stochastic Differential Dynamical Programming

The Stochastic Differential Dynamical Programming (S-DDP) algorithm published in Theodorou *et. al.* (2010) in its more general form numerically solves nonlinear stochastic optimal control problems using first and second order expansions of stochastic dynamics and cost along nominal trajectories. It is based on the classic Differential Dynamical Programming algorithm, but considers stochastic dynamical systems of the form $\mathrm{d}x = f(x,u)\mathrm{d}t + F(x,u) \circledS \mathrm{d}\omega$ where x is the system state, u is the control input, and $d\omega$ is Brownian noise. The S-DDP algorithm considers a cost parameterized as $v(x,u,t) = E[\int_{t_0}^{t_f} l(x(\tau),u(\tau),\tau)\mathrm{d}\tau + h(x(t_g f))]$ where $h(\cdot)$ is the terminal cost and $l(\cdot)$ is the running cost. Note that the cost is an expectation since the underlining system dynamics are stochastic. Due to space limitations a formal derivation of the S-DDP algorithm cannot be given. However, the essence of the algorithm lies in finding the local variation in control δu that minimizes the given cost. First, given an initial discrete control input $u(t)$ the discretized expected trajectory of the system is found. Then, the cost-to-go function and system dynamics are linearized along the trajectory. An approximation of the cost-to-go function is found through back-propagation. The approximated function is used to find δu, δx (optimal state deviation), and an optimal feedback gain. The discrete control input is updated as $u_{\text{new}}(t) = u_{\text{old}}(t) + \gamma\delta u(t)$ and the process can then be repeated. In this paper γ is found through an Armijo line search (see Kelley (1999) for further details) and the algorithm is terminated when consecutive final costs differ by less then a defined amount. It should be noted that the algorithm relies heavily on the accuracy of the propagation of the expected trajectory of the system and the linearization of the system dynamics. The process is outline in Algorithm 2 .

3. Numerical Experiments

In this section we demonstrate the benefits of using the proposed stochastic variational integrator and its linearization when implementing the S-DDP algorithm. We consider an experiment involving the control of a dynamical system representing a human finger (3-link planar manipulator) (see Li & Valero-Cuevas (2009) for system model and parameters) . It is shown that the solution of the S-DDP algorithm when utilizing the variational integrator is far less dependent on

Algorithm 2 Outline of S-DDP with Armijio Line Search

Require:
Initial discrete control input $u(t)$, Algorithm parameters α, β, ϵ
Cost function $v(x,u,t)$, Stochastic dynamics $dx = f(x,u)dt + F(x,u)d\omega$
while Cost updates results in more than ϵ in difference **do**
  Find discretized expected trajectory
  Linearize the value function and system dynamics along the trajectory
  Approximate the value function through back-propagation and compute δu and δx
  while $\text{Cost}_\text{p} > \text{Cost} + \alpha\beta(\delta x^\text{T}\nabla_x v(x,u,t) + \delta u^\text{T}\nabla_u v(x,u,t))$ **do**
    Find the proposed input $u_\text{p} \leftarrow u + \beta^j \delta u$ and corresponding trajectory, x_p
    Find proposed cost $\text{Cost}_\text{p} \leftarrow v(x_\text{p}, u_\text{p}, t)$ and update $j \leftarrow j+1$ if needed
  end while
  Update the control, trajectory and Cost $u \leftarrow u_\text{p}, x \leftarrow x_\text{p}, \text{Cost} \leftarrow \text{Cost}_\text{p}$
end while

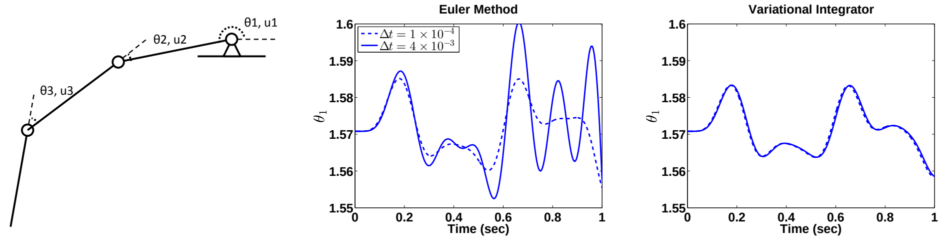
the discretization step size than when the Euler method is used †. The lack of dependence allows the algorithm to obtain a solution utilizing a relatively large step size that approximates a solution acquired utilizing a small step size. As a result, the computational time of the algorithm can be significantly reduced without degrading its performance. As shown in Figure 1a the dynamical system is described by three coordinates given by the relative angles between adjacent links, $\theta(t) = [\theta_1(t), \theta_2(t), \theta_3(t)]$, and three control inputs, $u(t) = [u_1(t), u_2(t), u_3(t)]$. Figures 1b and 1c show a comparison between the propagation of $\theta_1(t)$ using the Euler method and the presented variational integrator with two different step sizes subject to initial conditions $\theta(t_0) = [\pi/2, 0, 0]$ and $\dot{\theta}(t_0) = [0,0,1]$, $u(t) = 0$, and $\epsilon_\text{tol} = 1 \times 10^{-12}$ (when the variational integrator was used). Note that the accuracy of the Euler method is degraded when the discretization step size is increased while the variational integrator is negligibly affected.

For all cases considered the reference tracking based cost function is $J(t) = \int_{t_0}^{t_f} [10(\theta(\tau) - \theta_t(\tau))^2 + 0.1u(\tau)^2]d\tau + (\theta(t_f) - \theta_t(t_f))^2$, where $\theta_t(t) = [1 + 0.25\sin(4\pi t), 0.1\sin(2\pi t), 0]$ and $t_f - t_0 = 1.5$. Algorithm 2 parameters were set to $\epsilon = 1 \times 10^{-4}$, $\alpha = 1 \times 10^{-8}$, and $\beta = 0.25$.

In order to examine the performance of the variational integrator in a variety of situations three cases were considered: white noise, control-dependent noise, and state-dependent noise. In the first case, $F_i(q, \dot{q}, u) = Ce_i,\ i = \{1,2,3\}, C = 5 \times 10^{-5}$ ‡. As discussed by Theodorou *et. al.* (2010) if the additive noise is neither state nor control dependent the S-DDP algorithm is equivalent to the classic DDP solution. Figures 2a and 2b display the calculated optimal solutions using the Euler method and the variational integrator with two different step sizes. It is concluded that the Euler method is far more dependent on the discretization step size. Figure 2c shows the optimized cost as a function of the running computational time of the S-DDP algorithm. Note that when the variational integrator is used the required computational time can be drastically reduced without significantly changing the optimized cost. Table 1 and Figure 5a show that the optimized cost increases more rapidly as the discretization time step is increased when the Euler method is used. Additionally, the number of algorithm iterations is more predictable when the variational integrator is used. Depending on the allowable deviation from the true optimal solution (assuming it is obtained when $\Delta t = 0.0001$) the required computational time can be drastically reduced by utilizing a larger step size when the variational integrator is used. Finally, it should be noted that the growth of the optimized cost when the variational integrator is used is cause, in part, by the

† Given a stochastic differential equation $dx = f(x,u)dt + \sum^p F_i(x,u)d\omega_i$ the resulting integration equation using the Euler method is $x_{k+1} = x_k + f(x_k,u_k)\Delta t + \sum^p F_i(x_k,u_k)\Delta\omega_{ik}$ and its linearization is $\delta x_{k+1} = (\text{I} + \nabla_{x_k} f + \sum^p \nabla_{x_k} F_i \Delta\omega_{ik})\delta x_k + (\nabla_{u_k} f + \sum^p \nabla_{u_k} F_i \Delta\omega_{ik})\delta u_k + \sum^p F_i(x_k,u_k)\Delta\omega_{ik}$.

‡ The standard basis is used to define vectors $e_i, i \in \{1,2,3\}$ (e.g. $e_1 = [1,0,0]^\text{T}$).



(a) (b) (c)

Figure 1: (a): Diagram of the studied dynamical system. (b) and (c): Propagation of the $\theta_1(t)$ with initial condition $\theta(t_0) = [\pi/2, 0, 0]$ and $\dot{\theta}(t_0) = [0, 0, 1]$ subjected to input $u(t) = 0$. Dotted lines indicate a step size of $\Delta t = 1 \times 10^{-4}$ while solid lines indicate $\Delta t = 2 \times 10^{-2}$.

reduction of the controller's bandwidth. However, how much of the growth can be attributed to the change in the controller's bandwidth and not other effects (integrator accuracy, numerical precision, etc.) is an area of future research.

The next case considers control-dependent noise where $F_i(q, \dot{q}, u) = Ce_i u_i^2(t),\ i = \{1, 2, 3\}$, $C = 5 \times 10^{-4}$. As shown in the first case, Figures 3a and 3b show that the optimized solution obtained utilizing the Euler method is far more dependent on the discretization step size. Furthermore, note that since $\Delta\omega_{i_k} \sim \mathcal{N}(0, \sqrt{\Delta t})$ the perceived amount of noise at discretization points increases with Δt. Therefore, as shown in 3c the optimal control input will be dependent on the time step since it is advantageous to reduced the amount of induced noise. As a result, the optimized cost should increase as the time step increases. However, how much of the growth can be attributed to increases in noise and not other effects already discussed is unknown. Nevertheless, as shown in Figure 5b the growth in the cost is far greater when the Euler method is used.

State-dependent noise is considered in the final case, $F_i(q, \dot{q}, u) = C(\theta_i(t) - \theta_i(t_0))^2,\ i = \{1, 2, 3\}, C = 5 \times 10^{-4}$. Figures 3a and 3b display the calculated optimal solutions using the Euler method and the variational integrator with two different step sizes. Note both methods are strongly dependent on the discretization step size. As in case 2, the perceived amount of noise at discretization points increases with Δt. In this case, it is advantageous to keep the system near the initial equilibrium state. Therefore, a trade-off between the tracking performance and the amount of induce noise occurs. As illustrated in Figure 4, the optimized state trajectory is dependent on the step size. However, as shown in Table 1 the growth in the optimized cost is far greater when the Euler method is used.

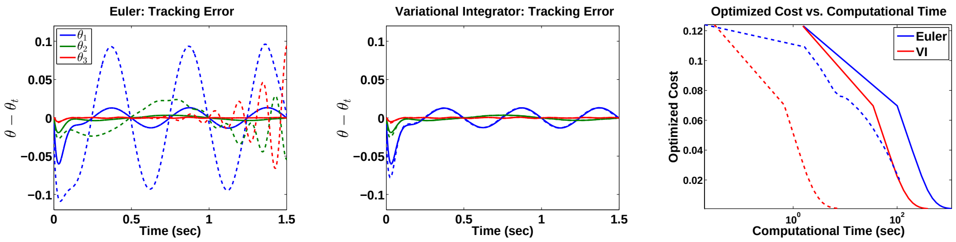
4. Conclusion

In this paper we proposed a stochastic variational integrator and its linearization. The Stochastic Differential Dynamical Programming algorithm was used to compare the performance of the variational integrator to that of the standard Euler method. We demonstrated that the S-DDP algorithm was far less dependent on the discretization time step when the variational integrator was used to propagate system trajectories and linearize system dynamics. Therefore, a significant reduction in computational time can be achieved without sacrificing algorithm performance. We stress that these benefits are not limited to S-DDP and similar improvements can be expected to any process that utilizes propagated system trajectories or linearized system dynamics.

5. Acknowledgment

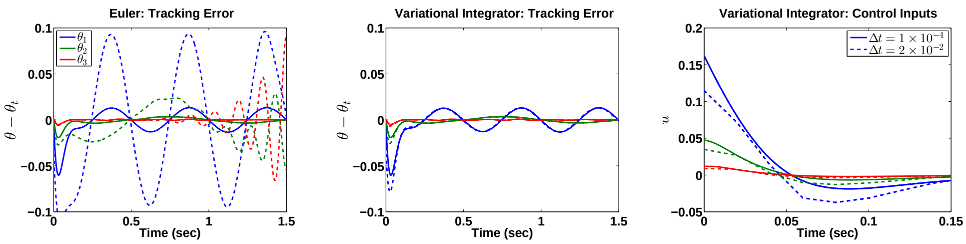
This research is partially supported by the NSF NRI-1426945.

REFERENCES



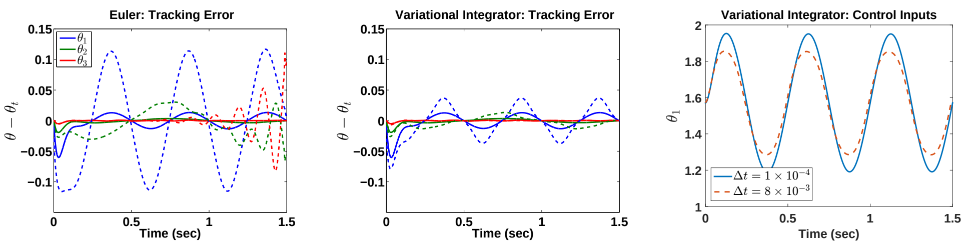
(a) (b) (c)

Figure 2: (a) and (b): Tracking error of the system states in Case 1. (c): Optimized cost versus the running computational time of the S-DDP algorithm (computational time plotted on a logarithmic scale). Dotted lines indicate a step size of $\Delta t = 1 \times 10^{-4}$ while solid lines indicate $\Delta t = 6 \times 10^{-3}$.



(a) (b) (c)

Figure 3: (a) and (b): Tracking error of the system states in Case 2. Dotted lines indicate a step size of $\Delta t = 1 \times 10^{-4}$ while solid lines indicate $\Delta t = 6 \times 10^{-3}$. (c): Optimal control inputs when the variational integrator method is used (initial 0.1 seconds shown). Dotted lines indicate a step size of $\Delta t = 1 \times 10^{-4}$ while solid lines indicate $\Delta t = 2 \times 10^{-2}$.



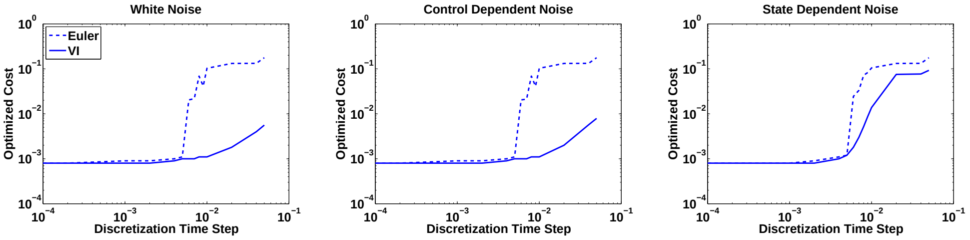
(a) (b) (c)

Figure 4: (a) and (b): Tracking error of the system states in Case 3. (c): Optimal θ_1 trajectories when the variational integrator is used. Dotted lines indicate a step size of $\Delta t = 1 \times 10^{-4}$ while solid lines indicate $\Delta t = 6 \times 10^{-3}$.

MARSDEN, J. E. & WEST, M. 2001 Discrete Mechanics and Variational Integrators. *Acta Numerica* **10**, 357–514.

JOHNSON, E. R. & MURPHEY, T. D. 2009 Scalable Variational Integrators for Constrained Mechanical Systems in Generalized Coordinates. *IEEE Transactions on Robotics* **25:6**, 1249–1261.

JOHNSON, E. R., SCHULTZ, J., & MURPHEY, T. D. 2015 Structured Linearization of Discrete Mechanical Systems for Analysis and Optimal Control. *IEEE Transactions on Automation Science and Engineering* **12:1**, 140–152.



(a) (b) (c)

Figure 5: Optimized cost as a function of the discretization time step. Dotted lines indicate the Euler method was used and solid lines indicate that the variational integrator was used.

Δt	0.0001	0.001	0.002	0.005	0.006	0.007	0.008	0.009	0.01	0.02
White Noise, Euler										
CPU Time (sec)	1119.7	107.7	62.0	56.6	112.5	175.4	26.5	240.6	26.4	2.3
Iterations 13	13	14	31	71	130	23	236	31	6	
White Noise, VI										
CPU Time (sec)	393.5	39.4	19.9	8.5	7.2	6.0	5.3	4.6	4.32	2.4
Iterations	13	13	13	13	13	13	13	13	14	14
Input Dep., Euler										
CPU Time (sec)	989.2	98.1	54.1	51.1	100.2	154.4	23.3	224.0	24.8	2.5
Iterations	13	13	14	31	71	131	23	241	31	6
Input Dep., VI										
CPU Time (sec)	412.1	38.1	19.1	7.8	6.4	5.3	4.8	4.6	4.2	2.2
Iterations	13	13	13	13	13	13	13	13	14	14
State Dep., Euler										
CPU Time (sec)	990.4	104.9	52.8	48.4	86.0	128.0	95.4	64.6	24.3	1.9
Iterations	13	14	14	31	63	107	92	71	28	6
State Dep., VI										
CPU Time (sec)	371.3	37.9	20.2	9.3	7.5	6.4	5.9	4.9	4.1	1.1
Iterations	13	13	14	15	15	15	15	15	14	4

Table 1: Summary of Data Collected from Numerical Experiments

WEST, M. 2004 Variational Integrators. Doctoral dissertation, California Institute of Technology.

HAIRER, E., LUBICH, C., & WANNER, G. 2004 Geometric numerical integration: structure-preserving algorithms for ordinary differential equations. *Springer Ser. in Computational Mathematics* Vol. 31.

LI, W. & VALERO-CUEVAS, F. 2009 Linear quadratic optimal control of contact transition with fingertip. *American Control Conference*.

KELLEY, C. T. 1999 Iterative methods for optimization. *Society for Industrial and Applied Mathematics*.

THEODOROU, E., TASSA, Y., & TODOROV, E. 2010 Stochastic differential dynamic programming. *American Control Conference*.

KLOEDEN, P. E., & PLATEN, E. 1992 Numerical solution of stochastic differential equations. *Springer Science & Business Media*.

TAO, M., OWHADI, H., & MARSDEN, J.E. 2010 Nonintrusive and structure preserving multiscale integration of stiff ODEs, SDEs, and Hamiltonian systems with hidden slow dynamics via flow averaging. *Multiscale Modeling & Simulation* **8:4**, 1269–1324.

BOU-RABEE, N., & OWHADI, H. 2010 Long-run accuracy of variational integrators in the stochastic context. *SIAM Journal on Numerical Analysis* **48:1**, 278–297.

BOU-RABEE, N. & OWHADI, H. 2008 Stochastic variational integrators. *IMA J. of Numerical Analysis*.

OBER-BLÖBAUM, S., TAO, M., CHENG, M., OWHADI, H., & MARSDEN, J.E. 2013 Variational integrators for electric circuits. *Journal of Computational Physics* **242**, 498–530.