Sampling-Based Nonlinear Stochastic Optimal Control for Neuromechanical Systems

Emily A. Reed,[1] Marcus A. Pereira[2], Francisco J. Valero-Cuevas[3,*], and Evangelos A. Theodorou[2]

Abstract— Determining how the nervous system controls tendon-driven bodies remains an open question. Stochastic optimal control (SOC) has been proposed as a plausible analogy in the neuroscience community. SOC relies on solving the Hamilton-Jacobi-Bellman equation, which seeks to minimize a desired cost function for a given task with noisy controls. We evaluate and compare three SOC methodologies to produce tapping by a simulated planar 3-joint human index finger: iterative Linear Quadratic Gaussian (iLQG), Model-Predictive Path Integral Control (MPPI), and Deep Forward-Backward Stochastic Differential Equations (FBSDE). We show that averaged over 128 repeats these methodologies can place the fingertip at the desired final joint angles but–because of kinematic redundancy and the presence of noise–they each have joint trajectories and final postures with different means and variances. iLQG in particular, had the largest kinematic variance and departure from the final desired joint angles. We demonstrate that MPPI and FBSDE have superior performance for such nonlinear, tendon-driven systems with noisy controls.

Clinical relevance— The mathematical framework provided by MPPI and FBSDE may be best suited for tendon-driven anthropomorphic robots, exoskeletons, and prostheses for amputees.

I. INTRODUCTION

Understanding how the nervous system controls tendon-driven bodies - an important research area known as *'neuromuscular control'* - is vital to advancing rehabilitation and treatment technologies and theories for the restoration of motor movement in patients with motor disabilities, as well as creating robotic exoskeletons and prostheses for amputees. Tendon-driven systems, in particular, represent a class of mechanical systems that are simultaneously under- and over-determined [1]–[3]. Under-determined because multiple combinations of tendon tensions can produce the same net joint torques, and over-determined because the failure of any one tendon to lengthen appropriately can disrupt or even lock-up joint rotations.

There have been several theories that aim to explain neuromuscular control [2], [4]–[8].

*Equal senior authorship.

[1]Reed is with the Ming Hsieh Electrical and Computer Engineering Department, University of Southern California, Los Angeles, CA 90007, USA emilyree@usc.edu

[2]Pereira and Theodorou are with the Daniel Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA marcus.pereira@gatech.edu, evangelos.theodorou@gatech.edu

[3]Valero-Cuevas is with the Department of Biomedical Engineering, University of Southern California, Los Angeles, CA 90007, USA valero@usc.edu

In this paper, we more deeply investigate stochastic optimal control (SOC) as a theory of neuromuscular control for fingers to evaluate its potential in controlling tendon-driven robotic prosthetic devices.

Stochastic Optimal Control relies on dynamic programming, leading to the Hamilton-Jacobi-Bellman (HJB) equation, a backward nonlinear partial differential equation (PDE) that aims to minimize the expectation of the desired cost function based on a prescribed task. In general, solutions of this backward PDE suffer from the well known "curse of dimensionality," which describes the exponentially increasing computational complexity of solving the HJB as the dimensionalty of the system increases [9]. Therefore, several approximate methods have been presented that attempt to solve the HJB equation [10]–[14]. These approximate methods try to address the "curse of dimensionality;" however, in this work, we only consider scalable stochastic optimal control methods that solve the HJB locally around a nominal system trajectory without approximation.

We compare and contrast three SOC methods for controlling a tendon-driven index finger modeled in simulation to perform a tapping task. These three methods are iterative Linear Quadratic Gaussian (iLQG), Model Predictive Path Integral (MPPI), and Forward-Backward Stochastic Differential Equations (FBSDEs). Previous work has demonstrated that the iLQG method can be used to control the tendon-driven index finger model [15].

iLQG relies on linearization of the dynamics of the system and quadratic approximation of the cost function [16]. Thus, conditions on differentiability of the model is necessary to employ this method [17].

MPPI is advantageous in its ability to sample from nonlinear dynamics without requiring linearization [18], [19]. In addition, MPPI is a parallelizable algorthm that can run in real time when implemented on a Graphical Processing Unit (GPU). However, an assumption between control authority and the system noise variance is required, which prohibits tuning the control authority independent of the noise entering the dynamics, thereby contributing to the potential difficulty in computing the optimal control policy [20], [21].

FBSDEs do not require the aforementioned assumptions in iLQG and MPPI and is therefore a more general framework. However, the backward SDE introduces computational complications, which require sophisticated machine learning tools for computing solutions.

This paper is organized as follows: Section II introduces the index finger model derived in [15], [22]. Section III discusses the mathematical formulation of the stochastic

optimal control problem being solved and presents an intuitive explanation of each of the three different control methods being compared. Section IV presents the results of the simulation of the index finger completing a tapping task. Section V discusses the limitations of current commercially available robotic prostheses and analyzes the applicability of these methods to this technology. Section VI presents our conclusions and directions for future work.

II. INDEX FINGER DYNAMICAL MODEL

The skeleton of the human index finger is made up of three joints connected by three rigid links. Two of the joints, the *interphalangeal proximal* (PIP) and the *interpahalangeal distal* (DIP), are described as hinge joints that can generate flexion - extension. The *metacarpophalangeal joint* (MCP), located closest to the human arm, is a saddle joint, so it can generate flexion - extension in addition to abduction - adduction.

The index finger has seven muscles. There are two flexor muscles called the *Flexor Digitorum Profundus* (FDS) and the *Flexor Digitorum Superficialis* (FDP). There are two extensor muscles called the *Extensor Communis* (EC) and the *Extensor Indicis* (EI). The last three muscles are called the *Radial Interosseous* (RI), the *Ulnar Interosseous* (UI) and the *Lumbrical* (LU).

The index finger has 11 tendons, seven of which actively actuate the seven muscles of the index finger producing torques at the three joints. The other four tendons are passive since they are attached between the other tendons and the bones. The passive tendons are called the *Terminal Extensor* (TE), the *Radial Band* (RB), the *Ulnar Band* (UB), and the *Extensor Slip* (ES).

The details of the complete index finger model including modeling the tendon excursion, moment arm matrix, and velocity of the tendons is thoroughly explained in [15]. Hence, we will simply present the index finger dynamics, which are represented as stochastic differential equations (SDEs).

$$\mathrm{d}\dot{\theta} = (-\mathbf{I}(\theta)^{-1}\mathbf{C}(\theta,\dot{\theta}) + \mathbf{B}\dot{\theta} + \mathbf{I}(\theta)^{-1}\mathbf{T})\mathrm{d}t \qquad (1)$$

$$\mathbf{T} = \mathbf{M}(\theta)\mathbf{F} \qquad (2)$$

$$\mathrm{d}\mathbf{F} = -\frac{1}{\tau}(\mathbf{F} - \mathbf{u})\mathrm{d}t + \mathrm{d}\mathbf{w} \qquad (3)$$

$$\mathbf{u} > 0 \qquad (4)$$

where $\mathbf{I}(\theta) \in \mathbb{R}^{6\times 6}$ is the inertia matrix, $\mathbf{C}(\theta,\dot{\theta}) \in \mathbb{R}^{6\times 1}$ is a vector of centripetal and Coriolos forces, $\mathbf{B} \in \mathbb{R}^{3\times 3}$ is the joint friction matrix, $\mathbf{M} \in \mathbb{R}^{3\times 7}$ is the moment arm matrix, $\mathbf{T} \in \mathbb{R}^{3\times 1}$ is the torque vector, $\mathbf{F} \in \mathbb{R}^{7\times 1}$ is the tensions (forces) on the tendons, $\mathbf{u} \in \mathbb{R}^{7\times 1}$ is the control vector, $\mathbf{w} \in \mathbb{R}^{7\times 1}$ is Brownian motion noise vector with variance $\sigma^2 I_{7\times 7}$, and $\mathrm{d}\dot{\theta} \in \mathbb{R}^{6\times 1}$ describes the joint space kinematics (i.e. the joint angles and joint angular velocities) for all three joints. For our simulations, we have excluded

the abduction-adduction movement of the MCP joint, so we only examine planar movements. Therefore, the state-space formulation of the index finger model has a dimensionality of 13, corresponding to six states for the joint space kinematics ($\theta = (\theta_1, \theta_2, \theta_3, \dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3)$) and seven states for the tensions (**F**) applied on the seven active tendons. Since tendons can only pull on the muscles, imposing the constraint $\mathbf{u} > 0$ gives $\mathbf{F} > 0$. Equation (3) is used to model delays in the generation of tensions on the tendons. τ is the time constant of muscle activation, which is 0.04 in all simulations.

A figure of the human index finger with labeled joints and tendons is shown in Figure 1. While this model does not consider the muscle activations, the model can easily be extended to include them similar to the work in [17].
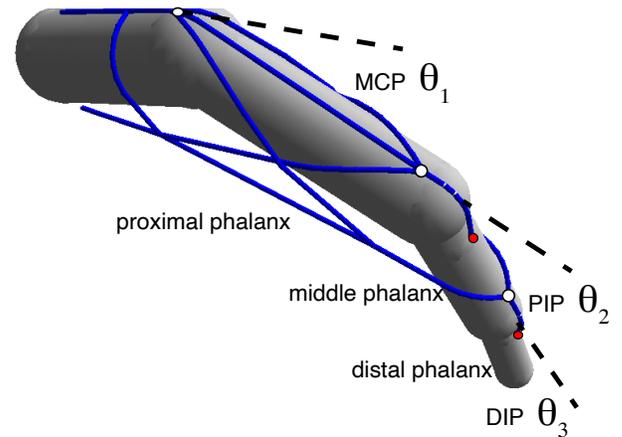
Fig. 1. Anatomy of the index finger model used here and in [15]. The finger flexes on the plane of the flexion angles θ_1, θ_2, and θ_3. Figure adapted from [8].

III. STOCHASTIC OPTIMAL CONTROL

Stochastic Optimal Control (SOC) seeks to find the control input (**u**) at each time step that is optimal with respect to a prescribed cost function (J) subject to the stochastic dynamics of the system. In general, the cost function describes the goal of the problem by penalizing the distance that the system state is away from a desired target state or trajectory as well as penalizing the amount of control effort necessary to achieve the final state. In the case of our specific problem, the goal is for the index finger to reach the final prescribed tapping position specified by joint angles while minimizing the amount of force necessary to achieve this goal. In general SOC problems, the cost function is formulated as follows

$$J(\mathbf{x}_0, t_0) = \mathbb{E}\left[\phi(\mathbf{x}_T, T) + \int_{t_0}^{T}\mathscr{L}(\mathbf{x}_t, \mathbf{u}_t, t)\mathrm{d}t\right] \qquad (5)$$

where $\phi(\mathbf{x}_T, T)$ is the terminal cost and depends only on the final state $\mathbf{x}_T$, $\mathscr{L}(\mathbf{x}_t, \mathbf{u}_t, t)$ is the running state and control cost, and T is the time horizon such that $0 < t < T < \infty$. Let $(\Omega, \mathscr{F}, \{\mathscr{F}_t\}_{t\geq 0}, \mathbb{Q})$ be a complete, filtered probability space on which a *m*-dimensional standard Brownian motion **w** is defined, such that $\{\mathscr{F}_t\}_{t\geq 0}$ is the normal filtration

of **w**. The expectation in (5) is taken with respect to the probability measure $\mathbb{Q}$ over the space of trajectories induced by controlled stochastic dynamics. In general, the dynamics are written as a nonlinear stochastic differential equation

$$d\mathbf{x} = F(\mathbf{x}, \mathbf{u})dt + B(\mathbf{x}, \mathbf{u})d\mathbf{w} \tag{6}$$

where $\mathbf{x} \in \mathbb{R}^{n \times 1}$ is the state vector, $\mathbf{u} \in \mathbb{R}^{m \times 1}$ is the control vector, and $\mathbf{w} \in \mathbb{R}^{m \times 1}$ is a vector of Brownian motions with Σ the covariance matrix.

The solution seeks to find the optimal balance between achieving the desired final state while still minimizing the amount of control input being injected into the system. The optimal control ($\mathbf{u}^*$) gives the minimum cost (J^*). The optimal cost is defined as the minimum expected cost evaluated with respect to the controlled trajectories satisfying the SDE system dynamics accumulated over the time horizon $(t_0, \ldots, T)$ starting from the initial state $\mathbf{x}_{t_0}$ going to the final state $\mathbf{x}_T$. The optimal control and cost are described by the following two equations

$$J(\mathbf{x}_{t_0}, t_0) = \min_{\mathbf{u}(\cdot)} \mathbb{E}\left[\phi(\mathbf{x}_T, T) + \int_{t_0}^{T} \mathscr{L}(\mathbf{x}_t, \mathbf{u}_t, t)dt\right] \tag{7}$$

The optimal control is given by solving the following

$$\mathbf{u}(\cdot) = \text{argmin}_{\mathbf{u}(\cdot)} \mathbb{E}\left[\phi(\mathbf{x}_T, T) + \int_{t_0}^{T} \mathscr{L}(\mathbf{x}_t, \mathbf{u}_t, t)dt\right] \tag{8}$$

subject to the dynamics of the system. We consider dynamics that are affine with respect to the control **u**.

$$F(\mathbf{x}, \mathbf{u}) = f(\mathbf{x}, t) + G(\mathbf{x}, t)\mathbf{u} \tag{9}$$

In this paper, we consider only quadratic costs so

$$\phi(x_T, T) = (\mathbf{x} - \eta)^{\text{T}} \mathbf{Q}_T (\mathbf{x} - \eta) \tag{10}$$

where η is the target state and $\mathbf{Q}_T$ specifies the cost of the deviation of the states from their desired target at the final time. The running cost is

$$\mathscr{L}(\mathbf{x}_t, \mathbf{u}_t, t) = (\mathbf{x} - \eta)^{\text{T}} \mathbf{Q}(\mathbf{x} - \eta) + \frac{1}{2}\mathbf{u}_t^{\text{T}} \mathbf{R}\mathbf{u}_t \tag{11}$$

where **Q** is a matrix specifying the cost of the deviation of the states from their desired target at each time step and **R** is a symmetric positive definite matrix that specifies the cost of the control effort. For any given initial conditions (x_{t_0}, t_0), we wish to solve (7). The solution is obtained by solving the associated HJB for the Value function [23]. As in [21], with the set of admissible controls U, we can define the value function as

$$\begin{cases} V(\mathbf{x}, t) = \inf_{\mathbf{u}(\cdot) \in U[0, T]} J(\mathbf{x}, t) \\ V(\mathbf{x}_T, T) = \phi(\mathbf{x}_T, T) \end{cases} \tag{12}$$

Using Bellman's stochastic principle, as shown in [24], if the value function satisfies certain conditions, then its solution can be found with Itô's Differentiation Rule to satisfy the HJB equation

$$\begin{cases} V_t + \inf_{\mathbf{u}(\cdot) \in U[0, T]} \left\{ \frac{1}{2}\text{tr}(V_{xx}BB^{\text{T}}) + V_x^{\text{T}}(f + G\mathbf{u}) \right. \\ \left. + (\mathbf{x} - \eta)^{\text{T}}\mathbf{Q}(\mathbf{x} - \eta) + \frac{1}{2}\mathbf{u}^{\text{T}}\mathbf{R}\mathbf{u} \right\} = 0 \\ V(\mathbf{x}_T, T) = \phi(\mathbf{x}_T, T) \end{cases} \tag{13}$$

where V_x, V_{xx} denote the gradient and Hessian of V respectively. For the specific case of control-affine dynamics and quadratic control cost, the infimum operation can be carried out by taking the gradient with respect to **u** and setting it to zero.

$$G^{\text{T}}(\mathbf{x}, t)V_x(\mathbf{x}, t) + \mathbf{R}\mathbf{u} = 0 \tag{14}$$

Therefore, the optimal control is obtained as

$$\mathbf{u}^* = -\mathbf{R}^{-1}G^{\text{T}}(\mathbf{x}, t)V_x(\mathbf{x}, t) \tag{15}$$

While the details of the specific control algorithms can be found in [17], [19], [21], we briefly give an intuitive explanation of each method.

A. iLQG

iLQG starts by discretizing and linearizing the dynamics using Taylor's approximation up to the first order. Next, the cost function is approximated to the second order (i.e. quadratic). In this paper, we only consider quadratic costs, so we can skip this step. The main idea is to take expansions on both sides of the value function up to the second order and equate the terms. By substituting in the expanded terms and taking the gradient with respect to the control, we obtain a control update law, which can be added to the control policy at the current iteration. By applying the new control, the nonlinear dynamics are propagated and a new trajectory is generated. The algorithm is repeated again until convergence.

B. MPPI

MPPI is a subclass of Path Integral Control, which employs an information theoretic approach to solve the HJB equation. Path Integral control is based on the SOC and HJB formulation, while MPPI is based on the information theoretic approach. In general these are not related, however for the special case of control-affine dynamics and quadratic control cost, the two overlap [25, Sec. III]. Since we have restricted our problem to control-affine dynamics and quadratic control cost, we can use either MPPI or Path Integral control.

MPPI equates a concept in information theory known as the free energy to the Value function. Using this relationship, which is obtained via the Feynman-Kac Lemma, the algorithm avoids taking gradients of the value function by instead sampling the controls from the stochasticity present in the dynamics. MPPI uses a model-predictive control (MPC) formulation to determine the controls that give the best trajectories specified by the prescribed cost function for a certain time horizon. At each time step, MPC calculates several trajectories. The samples of trajectories are used to calculate a cost for each trajectory. An exponential transformation is applied to all of the costs, which scales these costs

to values between 0 and 1. Based on this transformation, the control update law is in a sense a weighted average of the sampled controls that gave the best trajectories in terms of the minimum cost. In MPPI, the control authority is a function of the stochasticity in the system.

C. FBSDEs

This method transforms the HJB equation into a system of Forward-Backward Stochastic Differential Equations using a nonlinear version of the Feynman-Kac lemma [26]. Intuitively, the nonlinear Feynman-Kac lemma is a mathematical result that connects the solution of parabolic PDEs to a system of FBSDEs.

FBSDEs is a more general approach compared to the Path Integral Control framework, in that it does not rely on any assumptions between control authority and noise. The main challenge is in finding the solution to the backward SDE. A backward SDE is a stochastic dynamical system that propagates backwards from terminal time to initial time. It turns out that this process requires the back-propagation of a conditional expectation as the noise entering the system evolves forward in time [27, Chapter 2]. One way to solve this is by combining importance sampling while using a Deep Neural Network (DNN) that approximates the gradient of the value function at every time step. A recurrent neural network architecture known as the Long-Short Term Memory (LSTM) can be trained with any one of the variants of Stochastic Gradient Descent. (For more details refer to our previous works [21], [28], [29]). Once trained, the network can be used to predict the optimal control at every time step starting from the given initial condition.

IV. SIMULATION RESULTS

The three algorithms were tested on controlling the index finger to tap from a starting joint angular position of $\theta_0 = \left(\frac{5\pi}{6}, \frac{\pi}{2}, \frac{\pi}{10}\right)$ to a final joint angular position of $\theta_T = \left(\frac{7\pi}{6}, \frac{\pi}{4}, \frac{\pi}{12}\right)$. For all three algorithms, we used a Brownian noise with $\sigma = 0.1$, a time horizon of 0.42 seconds, and found the mean and standard deviation across 128 trials. iLQG and MPPI were simulated with a time step of 0.01 whereas, FBSDE was simulated with a time step of 0.02. In addition, the learning rate for iLQG was 0.1 and for MPPI was 0.01.

The specific terminal, running state, and control cost matrices were found by hand tuning until the algorithm converged. For iLQG, the terminal cost was set to $\mathbf{Q}_T = \text{diag}(6.8e4, 1.23e4, 5.04e4, 0, 10, 50, 0, 0, 0, 0, 0, 0, 0)$, the running state cost was set to $\mathbf{Q} = \text{diag}(6e4, 1e2, 6.0075e4, 0.1, 0.1, 14, 0, 0, 0, 0, 0, 0, 0)$, and the control cost was set to $\mathbf{R} = rI_{7\times7}$, where $r = 2$. The simulation was run without parallelization in MATLAB on a Dell laptop with an Intel(R) Core(TM) i7-7500U CPU running at 2.70GHz and 12.0GB RAM. The total run time for iLQG was 6261.536317 seconds.

For MPPI, the discrete formulation was implemented using the software developed in [30]. The number of sample trajectories was set to 2000. The MPC horizon

Algorithm	Advantages	Limitations	Performance
iLQG	Requires low computational resources	Requires differentiability conditions	Adequate
MPPI	Applicability to non-differentiable models	Requires a GPU to run in real time	Best
FBSDE	Consistently reaches target	Computationally intensive	Good

was set to 20 time steps. The terminal cost was set to $\mathbf{Q}_T = \text{diag}(5e2, 4e2, 5.5e2, 1e2, 0, 1e2, 1, 1, 1, 1, 1, 1, 1)$. The running state cost was set to $\mathbf{Q} = \text{diag}(1e2, 1e2, 2e2, 0, 1, 10, 1, 1, 1, 1, 1, 1, 1)$. In MPPI, the control cost is set to be inversely proportional to the noise, so it was set to $r = 10$. The simulation was run without parallelization in MATLAB on a Dell laptop with an Intel(R) Core(TM) i7-7500U CPU running at 2.70GHz and 12.0GB RAM. The total run time for MPPI was 7923.996634 seconds.

For FBSDE, the terminal cost and running state cost were both set to $\mathbf{Q} = \mathbf{Q}_T = \text{diag}(5, 10, 10, 0.1, 0.5, 0.5, 0, 0, 0, 0, 0, 0, 0)$. The control cost was set to $r = 0.2$. We used two layer stacked LSTMs with eight hidden units each and trained the network for 10,000 iterations with a learning rate of 0.001 using the ADAM optimizer. The simulation was run without parallelization on the CPU version of TensorFlow [31] on a desktop with an AMD FX(tm)-8320 Eight-Core Processor 8 and 15.6 GiB RAM. The total run time for FBSDE was 8040.74667883 seconds. The results of the three joint angles achieving their desired final positions are presented in Figure 2.

We show that FBSDE produces very good results by consistently reaching the target with the lowest variance. iLQG gives inconsistent results with higher variance; however, it requires the least amount of computational resources. MPPI seems to converge the fastest and stays close to the desired target but sometimes misses the mark at the final time.

Furthermore in Figure 2, we notice that the third joint, the furthest linkage from the body, is seemingly more difficult to control. This is due to the inertial property and interaction torque that arise from the movement in other joints. This difficulty is observed in other mechanical systems such as a double inverted pendulum. We show that on average FBSDE is able to successfully control this third link, whereas MPPI has more difficulty and iLQG cannot seem to control the third link at all.

V. DISCUSSION

The state-of-the-art commercially-available prosthetic devices do not match the versatility of human movement in neither design nor control [32], [33]. Different methods are

Fig. 2. The trajectories of the three joint angles are shown comparing iLQG, MPPI, and FBSDE. We observe that iLQG produces the highest variance and is less likely to reach the desired target than the other two algorithms. FBSDE gives good results with the lowest variance and can consistently reach the desired target. MPPI gives the best results because it is able to converge the fastest but doesn't always reach the target at the end time.

available to control prosthetic devices, but '...their control is unnatural and requires a great amount of mental effort' [33]. Another example in [34] presents an approach that uses real-time biomechanical simulation to map between residual electromyograms (EMGs) and the motions of the intact hand. While real-time performance was achieved, the results showed the model to be unstable.

Prosthetic devices could benefit from a tendon-driven design [3], [35], and it is therefore important to find theoretically rigorous control strategies for them–and formal SOC is a natural candidate.

Our results show two SOC methods, namely MPPI and FBSDE, capable of accurately and consistently controlling a nonlinear tendon-driven finger to reach the designated target without a prescribed trajectory in the presence of noise. We hypothesize that iLQG gives poorer results due to the required linearization of the dynamics, whereas the sampling-based methods (MPPI and FBSDE) likely benefit from sampling the natural dynamics of the system. MPPI further has the better rise times to the target joint angles and smoother average trajectories as seen in Figure 2.

However, all of these algorithms require tuning the cost function based on building an intuition about the dynamical system, which is not trivial. One drawback to MPPI and FBSDE is that they require greater computational resources, see Table I. MPPI simulates 2,000 trajectories at every time step, and FBSDE requires a deep neural network to compute the optimal control. However, MPPI is a parallelizable algorithm that can be easily deployed on GPUs and can run in real time. Additionally, given the continual acceleration

of hardware, we expect that these two methods will soon be practical for prosthetic systems.

Despite the arduous tuning process and computational load, we believe that MPPI and FBSDE are the best solutions to control these tendon-driven systems. We envision that MPPI and FBSDE could be the algorithms of choice to control biomimetic hand prostheses like the one presented in [36] because they are the most robust to unavoidable motor and sensor noise and, –very likely–also robust to the external perturbations that are part and parcel of interacting with the environment.

VI. CONCLUSIONS

We compared three different stochastic optimal control strategies for producing smooth tapping movements with a tendon-driven human index finger in simulation. We found that MPPI and FBSDE were superior to iLQG because they produced lower variance in their trajectories and better average tapping accuracy. In the future, we plan to test these methods on tendon-driven hands and actual manipulation tasks, including intermittent finger-object contact as demonstrated in [37].

REFERENCES

[1] F. J. Valero-Cuevas, *Fundamentals of neuromechanics*. Springer, 2016.

[2] B. A. Cohn, M. Szedlák, B. Gärtner, and F. J. Valero-Cuevas, "Feasibility theory reconciles and informs alternative approaches to neuromuscular control," *Frontiers in computational neuroscience*, vol. 12, 2018.

[3] A. Marjaninejad, D. Urbina-Meléndez, B. A. Cohn, and F. J. Valero-Cuevas, "Autonomous functional movements in a tendon-driven limb via limited experience," *Nature machine intelligence*, vol. 1, no. 3, pp. 144–154, 2019.

[4] E. Todorov and M. I. Jordan, "Optimal feedback control as a theory of motor coordination," *Nature neuroscience*, vol. 5, no. 11, p. 1226, 2002.

[5] G. E. Loeb, "Optimal isn't good enough," *Biological cybernetics*, vol. 106, no. 11-12, pp. 757–765, 2012.

[6] M. Latash, M. Levin, J. Scholz, and G. Schöner, "Motor control theories and their applications," *Medicina*, vol. 46, no. 6, p. 382, 2010.

[7] E. Theodorou and F. J. Valero-Cuevas, "Optimality in neuromuscular systems," in *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*. IEEE, 2010, pp. 4510–4516.

[8] F. J. Valero-Cuevas, H. Hoffmann, M. U. Kurse, J. J. Kutch, and E. A. Theodorou, "Computational models for neuromuscular function," *IEEE reviews in biomedical engineering*, vol. 2, pp. 110–135, 2009.

[9] D. P. Bertsekas, *Dynamic programming and optimal control*. Athena scientific Belmont, MA, 1995, vol. 1, no. 2.

[10] I. M. Mitchell and C. J. Tomlin, "Overapproximating reachable sets by hamilton-jacobi projections," *journal of Scientific Computing*, vol. 19, no. 1-3, pp. 323–346, 2003.

[11] C. O. Aguilar and A. J. Krener, "Numerical solutions to the bellman equation of optimal control," *Journal of optimization theory and applications*, vol. 160, no. 2, pp. 527–552, 2014.

[12] M. B. Horowitz and J. W. Burdick, "Semidefinite relaxations for stochastic optimal control policies," in *2014 American Control Conference*. IEEE, 2014, pp. 3006–3012.

[13] M. B. Horowitz, A. Damle, and J. W. Burdick, "Linear hamilton jacobi bellman equations in high dimensions," in *53rd IEEE Conference on Decision and Control*. IEEE, 2014, pp. 5880–5887.

[14] A. A. Gorodetsky, S. Karaman, and Y. M. Marzouk, "Efficient high-dimensional stochastic optimal motion control using tensor-train decomposition." in *Robotics: Science and Systems*, 2015.

[15] E. Theodorou, E. Todorov, and F. J. Valero-Cuevas, "Neuromuscular stochastic optimal control of a tendon driven index finger model," in *Proceedings of the 2011 American Control Conference*. IEEE, 2011, pp. 348–355.

[16] E. Todorov, "Stochastic optimal control and estimation methods adapted to the noise characteristics of the sensorimotor system," *Neural computation*, vol. 17, no. 5, pp. 1084–1108, 2005.

[17] W. Li and E. Todorov, "Iterative linearization methods for approximately optimal control and estimation of non-linear stochastic system," *International Journal of Control*, vol. 80, no. 9, pp. 1439–1453, 2007.

[18] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Information-theoretic model predictive control: Theory and applications to autonomous driving," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1603–1622, 2018.

[19] ——, "Aggressive driving with model predictive path integral control," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1433–1440.

[20] I. Exarchos and E. A. Theodorou, "Stochastic optimal control via forward and backward stochastic differential equations and importance sampling," *Automatica*, vol. 87, pp. 159–165, 2018.

[21] M. A. Pereira, Z. Wang, I. Exarchos, and E. A. Theodorou, "Learning deep stochastic optimal control policies using forward-backward sdes," in *Robotics: science and systems*, 2019.

[22] N. Brook, J. Mizrahi, M. Shoham, and J. Dayan, "A biomechanical model of index finger dynamics," *Medical engineering & physics*, vol. 17, no. 1, pp. 54–63, 1995.

[23] P. Dorato, V. Cerone, and C. Abdallah, *Linear quadratic control: an introduction*. Krieger Publishing Co., Inc., 2000.

[24] J. Yong and X. Y. Zhou, *Stochastic controls: Hamiltonian systems and HJB equations*. Springer Science & Business Media, 1999, vol. 43.

[25] E. A. Theodorou and E. Todorov, "Relative entropy and free energy dualities: Connections to path integral and kl control," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. IEEE, 2012, pp. 1466–1473.

[26] E. Pardoux and A. Râșcanu, *Stochastic differential equations, Backward SDEs, Partial differential equations*. Springer, 2014, vol. 69.

[27] S. E. Shreve, *Stochastic calculus for finance II: Continuous-time models*. Springer Science & Business Media, 2004, vol. 11.

[28] Z. Wang, K. Lee, M. A. Pereira, I. Exarchos, and E. A. Theodorou, "Deep forward-backward sdes for min-max control," *arXiv preprint arXiv:1906.04771*, 2019.

[29] Z. Wang, M. A. Pereira, and E. A. Theodorou, "Deep 2fbsdes for systems with control multiplicative noise," *arXiv preprint arXiv:1906.04762*, 2019.

[30] acxz, "acxz/mppi: Mppi in octave/matlab," Nov. 2019. [Online]. Available: https://doi.org/10.5281/zenodo.3539762

[31] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: http://tensorflow.org/

[32] OttoBock. (2019) Solution overview upper limb prosthetics. [Online]. Available: https://www.ottobockus.com/prosthetics/upper-limb-prosthetics/solution-overview/

[33] A. L. Ciancio, F. Cordella, R. Barone, R. A. Romeo, A. D. Bellingegni, R. Sacchetti, A. Davalli, G. Di Pino, F. Ranieri, V. Di Lazzaro *et al.*, "Control of prosthetic hands via the peripheral nervous system," *Frontiers in neuroscience*, vol. 10, p. 116, 2016.

[34] D. Blana, E. K. Chadwick, A. J. van den Bogert, and W. M. Murray, "Real-time simulation of hand motion for prosthesis control," *Computer methods in biomechanics and biomedical engineering*, vol. 20, no. 5, pp. 540–549, 2017.

[35] A. Marjaninejad and F. J. Valero-Cuevas, "Should anthropomorphic systems be redundant;," in *Biomechanics of Anthropomorphic Systems, Springer Tracts in Advanced Robotics (STAR) series*. Springer, 2018.

[36] Zhe Xu and E. Todorov, "Design of a highly biomimetic anthropomorphic robotic hand towards artificial limb regeneration," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 3485–3492.

[37] Y. Tassa and E. Todorov, "Stochastic complementarity for local control of discontinuous dynamics," 2010.