Toward Exploiting the Natural Dynamics of Series Elastic Robots by Actuator-Centered Sequential Linear Program

Rachel Schlossman[1], Gray C. Thomas, and Luis Sentis

Abstract— **Dynamically consistent trajectory optimization for robots with series elastic joints is often ignored, with an obvious penalty to tracking performance. Yet it presents an opportunity for efficient solution due to exploitable problem structure. In this paper we explore a local planning structure which allows plans to be dynamically consistent with a series elastic actuated system, and which is furthermore able to exploit the dynamics of these actuators to improve over the performance limitations of the decoupled model. Our framework couples a linear second order actuator admittance model discretized by zero-order hold and a traditional force controlled nonlinear robot impedance model at a shifted port, and uses sequential convex optimization to find dynamically consistent trajectories. The port is shifted by our introduction of a tunable pseudo-mass parameter that improves the numerical tractability of this approach. We demonstrate the algorithm in simulation by maximizing end effector velocity for the series elastic actuated Draco robot and comparing to a rigidly actuated equivalent.**

I. INTRODUCTION

Since its inception [1], a primary drawback of series elastic actuation has been the additional challenge for the control system. Human-centered robots commonly make use of series elastic actuators (SEAs), which offer the benefits of compliance—for safe interaction with humans—increased robustness, and force sensitivity [2]. The compliant element is able to store and release energy, presenting an opportunity for increased efficiency and agility as compared to rigid actuators [3]. Both feedback controllers and trajectory planners are faced with a more complex challenge when interfacing with these systems, yet modern control systems for human-centered robots (largely based on [4]) rely on a force-control planning abstraction which specifies an unmeetable goal for the low level feedback controller and provides those controllers with planned trajectories that do not respect their dynamic limitations.

Interest in modified series elastic actuators with clutches [5] and variable stiffness compliant elements [6], [7], [8] has driven many groups to derive bang-bang style optimal behaviors to illustrate improved mechanical performance. Few groups, however, have investigated more general behaviors which allow for nonlinearities in the system. In [5] a convex optimization problem is formulated to maximize joint velocity, but it is accomplished by computing the switching times between rigid and compliant actuator behavior via the use of a clutch, and avoids the issue.

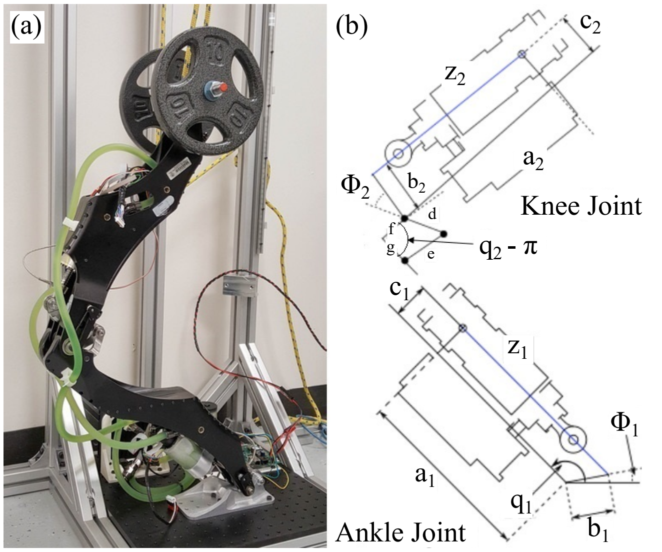

Fig. 1. (a) Draco leg prototype. Our simulations are modeled after a jumping task for this robot. (b) The schematics show the nonlinear transmission between actuator length and joint angle for the ankle and knee.

Iterative regulator-based optimal control has been successful in handling nonlinearities in these systems, but are restricted in capturing state and input constraints, e.g., transmission speed or spring deflection limits. The iLQR indirect method has been modified to allow input constraints [9], [10]—but only input constraints. Variable damping and the iLQR have been combined to achieve rapid motions [9], but this approach falls short of considering state constraints. In [10] it is used in combination with variable stiffness actuators to increase end effector velocity to throw a ball. Though the energy storing capability of the compliant element is leveraged, the motor position constraint can only be captured indirectly through the input constraint. Inequality state constraints in [11] are reformulated as canonical input constraints and the iLQR strategy is used for a two-link arm with compliant components. Yet the number of constraints possible with this strategy is at most the number of inputs.

A review of trajectory optimization techniques for series elastic walking provides insight into additional strategies that allow compliance while upholding constraints. Hybrid zero dynamics strategies have employed virtual, holonomic, and actuator constraints while implementing nonlinear programming (NLP) [12], [13] to minimize the energy consumption during each step. Another strategy, employed in the COMAN

*This work was supported by NASA Space Technology Research Fellowships NNX15AQ33H (G.C.T) and 80NSSC17K0188 (R.S.), and Office of Naval Research, ONR Grant N000141512507

[1] rachel.schlossman@utexas.edu

Authors are with The Departments of Mechanical Engineering (R.S., G.C.T.) or Aerospace Engineering (L.S.), University of Texas at Austin, Austin, TX 78712-0292, USA

robot, mimicked human center of mass trajectories and tuned stepping frequency to best use the compliant elements [14]. Our work offers the benefit of exploring problem structure at a lower level to efficiently handle nonlinearities and leverage compliance. We do so via sequential convex optimization, which allows us to leverage the speed of solving convex subproblems.

The state of the art in constrained planning for series elastic robots appears to be the NLP approach suggested in [15], in which optimal walking trajectories are produced to be consistent with compliant dynamics subject to all relevant constraints. Motion is parameterized via fifth order splines, and the spline's derivatives (linear in the spline knots) are set to match constrained dynamic equations at certain points along the spline. While it is hard to directly compare—their five-link biped has six actuators and performs a low-frequency-content behavior with only 54 total trajectory parameters whereas our robot has only two actuators and gentler nonlinearities, but roughly 98 parameter behaviors—the approach in [15] does not segregate the problem into linear and non-linear parts. While the team did not publish a computer run time, their rigidly actuated problem from two years earlier, [16], produced optimal trajectories in roughly 30 minutes, which leads us to suspect that our half minute run-time is a net efficiency improvement as well.

In this paper, we propose a direct optimization algorithm which considers the nonlinear effects of the transmission linkages, robot dynamics, input and state constraints, and the energy storing capabilities of the series elastic element. The algorithm uses linear programming to optimize a 1-norm and end velocity objective to demonstrate its ability to produce high performance behaviors while respecting system constraints. We formulate the problem as input selection for a time-varying discrete time linear system approximation that is updated iteratively. We formulate the system dynamics to connect the actuator space to the joint space, and use a fictitious pseudo-mass to improve discretization accuracy for the actuator component at large time steps. We exploit problem structure by separating the linear and nonlinear components of our model, which supports algorithm convergence. Convergence is not guaranteed, but is typically eight iterations or so.

II. Modeling

A. Actuator Dynamics

Our model considers internal actuator dynamics, which are common for control design, but rare for trajectory design. We follow the advice of [17] and [18], and construct a model from a constraint between three second-order systems to develop an unlumped model of the SEA.

The actuator model, as pictured in Figure 2, comprises the spring system; the motor system with input current, u; and the load system, which corresponds to the total motion. The states considered are spring displacement, δ; spring velocity, $\dot{\delta}$; motor displacement, y; and motor velocity, $\dot{y}$. The motor subsystem is reflected to prismatic motion through the transmission—hence, all parameters of the subsystems are expressed in linear units. The three systems are connected through a three-way differential. The state dynamics are related to the actuator displacement, z, and actuator velocity, $\dot{z}$, through the differential law relationship:

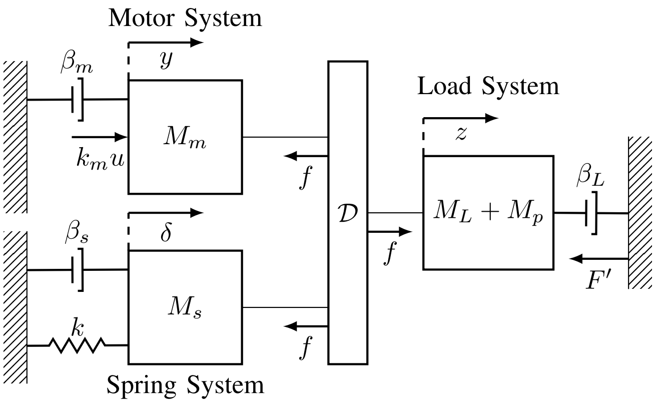

Fig. 2. Internal Dynamics of the SEA for the three mass differential constraint model. A pseudo-mass term, M_p is introduced to allow discretization with longer time steps.

$$z = \delta + y. \tag{1}$$

This is illustrated as the constraint block, $\mathcal{D}$, in Figure 2. The dynamics of the three subsystems are:

$$M_s\ddot{\delta} + \beta_s\dot{\delta} + k\delta = -f \tag{2}$$

$$(M_L + M_p)\ddot{z} + \beta_L\dot{z} = f - (F - M_p\ddot{z}) \tag{3}$$

$$M_m\ddot{y} + \beta_m\dot{y} = k_m u - f. \tag{4}$$

M_s, M_L, and M_m are the masses of the spring, load, and motor systems, respectively; β_s, β_L, and β_m are these systems' respective damping coefficients; k is the spring constant; and k_m is the reflected motor constant. The second input, F, is the force output from the actuator, which is used to link with the robot dynamics and the nonlinearities in the system. M_p is a fictitious pseudo-mass, which will be used to tune the convergence behavior of our algorithm, as discussed in Section III. To uphold conservation of mass:

$$F' \triangleq F - M_p\ddot{z}. \tag{5}$$

The variable f is equal to the back forces from the differential and, equivalently, the Lagrange multiplier which enforces the differential constraint.

In a single-link system, the nonlinearities that are captured by F' are associated with the actuator transmission component and the gravitational force. The dynamic equation for a single-link system is:

$$E_o\dot{x} = A_o x + B_{o,u}u + B_{o,F}F', \tag{6}$$

where state vector $x \triangleq \begin{bmatrix}\delta & \dot{\delta} & y & \dot{y}\end{bmatrix}^T$ and

$$E_o \triangleq \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & M_s + M_L + M_p & 0 & M_L + M_p \\ 0 & 0 & 1 & 0 \\ 0 & M_L + M_p & 0 & M_m + M_L + M_p \end{bmatrix}$$

$$A_o \triangleq \begin{bmatrix} 0 & 1 & 0 & 0 \\ -k & -(\beta_s+\beta_L) & 0 & -\beta_L \\ 0 & 0 & 0 & 1 \\ 0 & -\beta_L & 0 & -(\beta_L+\beta_m) \end{bmatrix}$$

$$B_{o,u} \triangleq \begin{bmatrix} 0 & 0 & 0 & k_m \end{bmatrix}^T, \quad B_{o,F} \triangleq \begin{bmatrix} 0 & -1 & 0 & -1 \end{bmatrix}^T.$$

Rearranging (6),

$$\dot{x} = A_1 x + B_{1,u} u + B_{1,F} F', \tag{7}$$

where

$$A_1 \triangleq E_0^{-1} A_0, \quad B_{1,u} \triangleq E_0^{-1} B_{o,u}, \quad \text{and}$$

$$B_{1,F} \triangleq E_0^{-1} B_{0,F}.$$

As we proceed, we will discuss the application of this formulation for the general case of p joints. Our state vector will be:

$$x = [x_1^T, x_2^T, \ldots, x_p^T]^T, \tag{8}$$

where each x_i captures the four states described in (6) for their respective actuator system.

B. Robot Dynamics

The force F connects the actuator to the robot dynamics. In general, for a multi-link system, the dynamics are:

$$M(q)\ddot{q} + C(q,\dot{q}) + G(q) = \tau = L(q)^T F, \tag{9}$$

where M, C, and G represent inertia, Coriolis and centrifugal, and gravitational forces, respectively, and q is the generalized joint angle vector.

The angle-dependent moment arm between the actuator and the linkage, $L(q)$, serves as the Jacobian between the joint space and the actuator space: $L(q)\dot{q} = \dot{z}$. Solving for F':

$$F' = F - M_p \ddot{z} = (L^{-T} M(q) L^{-1} - M_p)\ddot{z} + b(q,\dot{q}), \tag{10}$$

where

$$b(q,\dot{q}) \triangleq L^{-T}(C(q,\dot{q}) + G(q)) - (L^{-T} M(q) L^{-1} - M_p)\dot{L}\dot{q}. \tag{11}$$

This is an expression for the impedance of the robot at the $\{\dot{z}, F'\}$ port.

C. Discretization

To prepare for discrete time u optimization, the state space model is discretized into N time steps of length ΔT. By the continuous state space model in (7), acceleration at the actuator output can be computed as:

$$\ddot{z} = S(A_1 x + B_1 \begin{bmatrix} u \\ F' \end{bmatrix}). \tag{12}$$

This is actuator admittance at the $\{\dot{z}, F'\}$ port. S is formulated (using the Kronecker product $\otimes$) to capture the acceleration terms for the p-link system:

$$S = I_p \otimes \begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix}. \tag{13}$$

This formulation gives a value for F' in terms of the states:

$$\begin{aligned} F' =& [I - (L^{-T} M(q) L^{-1} - M_p) S B_{1,F}]^{-1} [L^{-T} G(q) + \\ &+ L^{-T} C(q,\dot{q}) + (L^{-T} M(q) L^{-1} - M_p)(S A_1 x + \\ &+ S B_{1,u} u - \dot{L}\dot{q})]. \end{aligned} \tag{14}$$

We discretize the linear actuator admittance model under the zero-order hold assumption for both u and F'. The discrete state space model is then:

$$x_{n+1} = A x_n + B \begin{bmatrix} u_n \\ F'_n \end{bmatrix}, \tag{15}$$

where

$$A \triangleq e^{A_1 \Delta T}, \quad B \triangleq \int_0^{\Delta T} e^{A_1(\Delta T - \tau)} B_1 d\tau. \tag{16}$$

Combining discrete time admittance and impedance at the $\{\dot{z}, F'\}$ interface, the discretized (time varying) update equation is:

$$x_{n+1} = A_{lin,n} x_n + B_{lin,n} u_n + bias_n, \tag{17}$$

where A_{lin} and B_{lin} capture all terms associated with the actuator states and input current, respectively, and $bias$ captures the nonlinear robot impedance. This model forms the foundation from which our algorithm is developed.

III. Iterative Linear Programming

For trajectory optimization of a p-link system, our approach follows a strategy that culminates in a convex subproblem. Our local optimization approach requires a baseline trajectory, z_{base}, to initialize the nonlinear parts of the dynamics. A slow trajectory or a static position both serve as good choices. There is no need for a similar baseline trajectory for the actuator states due to our exploitation of their linear problem structure. The z_{base} trajectory allows us to compute the time varying matrices used in (14) to compute F'. We can then compute the linearization components, A_{lin}, B_{lin}, and $bias$ are for each time step (effectively saving our solver from eliminating the F' variable itself).

New displacement and velocity trajectories for the spring and motor subsystems are then computed via convex optimization. The resulting trajectory for z becomes the new z_{base}, and is used to compute the linearization for the next iteration. Trust region constraints will keep the next z trajectory close to this updated z_{base} trajectory. The algorithm continues to run until the optimal value stops changing.

All relevant actuator state and input constraints can be included in the formulation. The constraints are associated with the upper and lower bounds of the allowable spring deflections, $\overline{\delta}$, joint limits, actuator ballscrew velocity, $\overline{\dot{y}}$, and input currents, $\overline{u}$. $\overline{\Delta z}$ and $\overline{\Delta \dot{z}}$ define the trust region, which can be used to aid convergence of the iteration scheme. We note that the dimension of this trust region is small relative to the full dimension of x. The final state can be subject to partial end point constraints. Our convex subproblem is thus:

$$\begin{aligned} &\underset{x,u}{\text{minimize}} && h(x,u) && \\ &\text{subject to} && \text{dynamics: (17)} && \forall\, n \in \mathcal{N}/N \end{aligned} \tag{18}$$

a trust region:

$$\begin{aligned} &|z_{i,n} - z_{i,n,base}| \leq \overline{\Delta z} && \forall\, i \in \mathcal{P},\ n \in \mathcal{N} \\ &|\dot{z}_{i,n} - \dot{z}_{i,n,base}| \leq \overline{\Delta \dot{z}} && \forall\, i \in \mathcal{P},\ n \in \mathcal{N} \end{aligned}$$

state constraints:

$$\begin{aligned} &|\delta_{i,n}| \leq \overline{\delta} && \forall\, i \in \mathcal{P},\ n \in \mathcal{N} \\ &z_{min,i} \leq z_{i,n} \leq z_{max,i} && \forall\, i \in \mathcal{P},\ n \in \mathcal{N} \\ &|\dot{y}_{i,n}| \leq \overline{\dot{y}} && \forall\, i \in \mathcal{P},\ n \in \mathcal{N} \\ &|u_{i,n}| \leq \overline{u} && \forall\, i \in \mathcal{P},\ n \in \mathcal{N}/N \end{aligned}$$

and problem-specific boundary conditions, in our case:

$$\begin{aligned} &x_0 = x_{\text{init.}},\ z_N = z_{\text{fin.}} \\ &J_{\text{hip\_x\_velocity}} \dot{z}_N = 0 \end{aligned}$$

where $\forall\ n \in \mathcal{N}$ means $n = 1, \ldots, N$, $\forall\ i \in \mathcal{P}$ means $i = 1, \ldots, p$, and / means omitting an element from the set.

A. Algorithm Tuning

To achieve convergence, we choose $\overline{\Delta z}$ and $\overline{\Delta \dot{z}}$ to limit planning to the region where our linearized dynamics are not too inaccurate. M_p is selected so that the fastest eigenvalue of A_{lin} and A_1 approximate the fastest eigenvalue of the continuous system, ensuring an accurate approximation of the system dynamics. When $M_p = 0$, the spring dynamics settle faster than one time step, and our algorithm fails to uphold conservation of energy.

IV. Apptronik Draco®-Inspired System

The formulation in the previous section can be applied to the two-link Draco robot. The Draco humanoid robot leg prototype is driven by viscoelastic actuators at its ankle and knee joints. The state space model in (17) is used with $p = 2$. The two actuators have equivalent spring, motor, and load dynamics.

Bolted as it is to the floor, the Draco leg, excluding the actuation linkages, is essentially a two-link manipulator, the dynamics of which are available in standard references. The process to develop the dynamic equations of the robot to include the actuator states follows that described in Section II. The variables F_1' and F_2' are obtained from Lagrangian dynamics with $M(q) \in R^{2\text{x}2}$, and $C(q, \ddot{q})$, $G(q) \in R^2$. Due to space limitations, the coefficients of $M(q)$, $C(q, \dot{q})$, and $G(q)$ can be found in [19], and our code is available in the author's public repository [20].

To use (14), the moment arms, $L_1(q_1)$ and $L_2(q_2)$, of the ankle and knee joints, respectively, must be considered such that:

$$L \triangleq \begin{bmatrix} L_1(q_1) & 0 \\ 0 & L_2(q_2) \end{bmatrix}. \tag{19}$$

The viscoelastic actuators used in the Draco system are very stiff, approximately $8e^6$ N/m. For this study, we explore the advantages of implementing softer springs in this system. We chose to demonstrate our algorithm for the goal of maximum velocity at the end tip of the upper leg, corresponding to the hip, to obtain an optimal trajectory for a jumping motion. The parameters used for the simulation were guided by system identification of our lab's SEA and the parameters of the Draco leg. Select parameters are included in Table I.

TABLE I
DYNAMICS

M_S (kg)	1.7
k_S (N/m)	300k
β_S (Ns/m)	0
M_m (kg)	293
β_m (Ns/m)	1680
M_L (kg)	0
M_p (kg)	1110
β_L (Ns/m)	0
I_1 (kg-m^2)	0.077
I_2 (kg-m^2)	0.150
$m_1 (kg)$	3.77
m_2 (kg)	30.61
l_1 (m)	0.5
l_2 (m)	0.5

TABLE II
TANSMISSIONS

a_1 (m)	0.21
b_1 (m)	0.04
c_1 (m)	0.02
ϕ_1 (rad)	.464
a_2 (m)	0.2
b_2 (m)	0.05
c_2 (m)	0.04
d (m)	0.04
e (m)	0.03
f (m)	0.03
g (m)	0.01
ϕ_2 (rad)	.524

TABLE III
CONSTRAINTS

$\overline{\delta}$ (m)	0.01
$z_{min,1}$ (m)	.1710
$z_{min,2}$ (m)	.1563
$z_{max,1}$ (m)	.2351
$z_{max,2}$ (m)	.2304
$\overline{\dot{y}}$ (m/s)	0.3
$\underline{u_{limit}}$ (A)	15
$\overline{\Delta z}_{1\&2}$ (m)	0.1
q_{1N} (rad)	$5\pi/8$
q_{2N} (rad)	$7\pi/4$
N	49
ΔT (s)	.0095

The parameters I_1 and I_2, m_1 and m_2, and l_1 and l_2 equal the moments of inertia, masses, and lengths of the lower and upper legs, respectively.

V. Simulation

We explore the situation in which a 25 lb weight is attached to the end of the upper leg. The simulation configuration is nearly identical to that shown in Figure 1.

A. Velocity Maximization

In this study, the cost function to be minimized expresses the goal to maximize the upward y-velocity of the hip at the final time, $V^* \triangleq J_{\text{hip\_y\_velocity}} \dot{z}_N$, where this Jacobian is known a-priori due to our constrained final position, $z_{\text{fin.}}$. We also seek to avoid unnecessary input current, and amend the cost function (to be minimized) to include the one-norm of control signal $u_{i,n}$:

$$h(x, u) = -J_{\text{hip\_y\_velocity}} \dot{z}_N + \alpha \sum_{i \in \mathcal{P}} \sum_{n \in \mathcal{N}/N} |u_{i,n}|, \tag{20}$$

where α equals $1e^{-9}$.

The initial condition is at equilibrium with the two springs, which drives the formulation of z_{base}. The initial and final conditions capture that the leg position starts and ends at the same angular configurations, q_{1N} and q_{2N}. The final constraint is that the x-component of velocity at the end of the upper leg is equal to zero at the final time.

The sequential convex optimization problem is solved using the Matlab CVX library [21]. The algorithm converges in approximately eight iterations, i, within a tolerance of 0.001 m/s difference between V_i^* and V_{i-1}^*. The corresponding behavior is shown in Figure 3. A time period of 0.4655 s is considered. An optimal value of 1 m/s upward velocity is achieved. One will notice spring oscillations, demonstrating the utilization of the two springs to store and release energy. Draco bends down and springs upward, following a human-like, shallow jumping trajectory.

We also sought to validate that the approximation of $\ddot{z}$ in (12) ensures a well-conditioned algorithm. Our selected

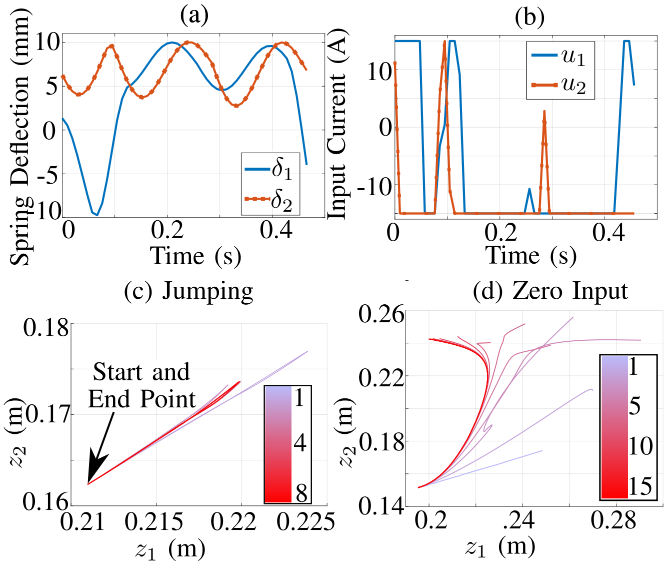

Fig. 3. (a) Spring deflection trajectories for the compliant leg's optimal behavior. (b) The corresponding optimal u's to produce the optimal trajectory, which operate at the input limits. (c) The z trajectories produced over eight iterations to produce the jumping behavior, demonstrating convergence. (d) The z trajectories produced over 15 iterations, showing convergence for the system's zero-input behavior.

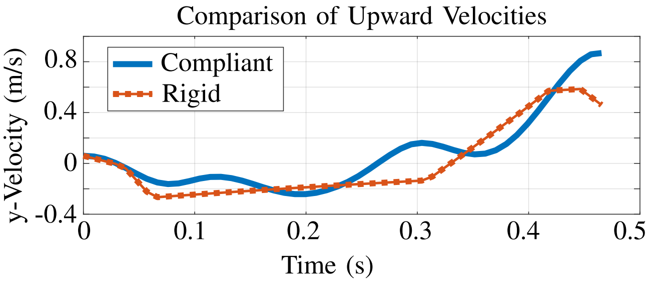

Fig. 4. This comparison between the upward velocity components of the optimal trajectories for the rigid and compliant systems shows that the final velocity of the compliant system is 1.9 times that of the rigid system.

$\ddot{z}$ approximation supports the algorithm's stability better than Euler's method of calculating the A_{lin}, B_{lin}, and *bias* matrices. The Euler alternative begins to converge only when the time step is less than 0.006.

Because the time varying robot impedance at $\{\dot{z}, F'\}$ is captured in F', our model can also be simulated directly for rigid reference. Specifically, (14) is used without substituting in for $\ddot{z}$, which allows for our algorithm to be run without any actuator dynamics. The cost function in (20) is used for the rigid and compliant cases, and the resulting optimal velocity is compared. The joint limit $z_{max,1}$ was tuned to 0.216 m and α was tuned to $1e^{-3}$ in order to perform a direct comparison. The optimal velocity in the compliant configuration, 0.8683 m/s, is 90% higher than that of the the rigid configuration, 0.4593 m/s. This result demonstrates the gains that can be achieved from leveraging the dynamics of the spring. One will notice in Figure 4 a dip in the upward velocity at the end of the time period for the rigid system. When we removed the constraint on the final hip x-velocity, this dip vanished and the gap reduced to 12% higher, which suggests the dip is required for the rigid system to avoid violating the final hip velocity constraint.

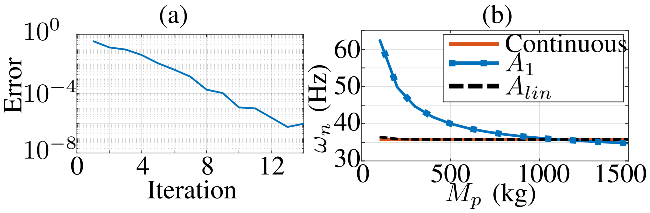

Fig. 5. (a) With the 15th iteration trajectory as a baseline, the results suggest that the error decays exponentially. (b) We select an M_p value where the natural frequency of the continuous system, A_1, and A_{lin} align.

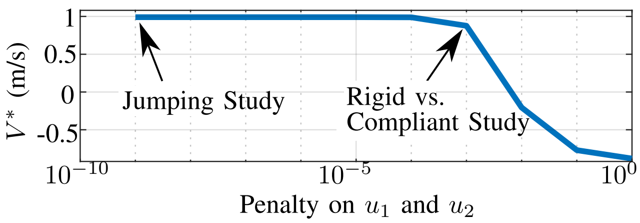

Fig. 6. Our algorithm converges as the α value is tuned, demonstrating the limits on optimal upward velocity, V^*, as the penalty increases.

B. Zero Input Behavior

To validate simulation accuracy, we considered the situation in which no current was sent to the system. We ensured that energy was conserved throughout the simulation. A test was conducted in which the system was released from rest from a nearly-vertical position. In this case, the system was more heavily influenced by the changing transmission as the links fell downward due to gravity. A time period of 0.5 seconds is considered with $\Delta T = 1e^{-4}$ to allow comparison to the numerically-difficult $M_p = 0$ case. As shown in Figure 3.d, the algorithm converged quickly, in about 12 iterations, even in this highly nonlinear case, thereby demonstrating its success in handling nonlinearities in the system. Energy fluctuates 16 times more with the $M_p = 0$ case compared to when M_p = 1,110 kg. With the nominal $\Delta T = 0.0095$ s, energy fluctuates by 1.89% as compared to the system energy gained from falling. These outcomes further support the importance of the M_p parameter in upholding that our model respects physics.

C. Algorithm Tuning

As discussed before, the largest eigenvalue of the linearized system should approximate that of the continuous system. Based on the output in Figure 5, a reflected load mass of 1,110 kg is selected for the two actuators. Additionally, the time step ΔT used to discretize the system in Section II-A must also be tuned such that the associated sampling frequency is significantly higher than the largest eigenvector of the continuous system. Figure 5 demonstrates exponential convergence of our iteration scheme. We also experimented with providing our algorithm with a better-informed nominal trajectory. Specifically, we used the optimal trajectory of the rigid system as the nominal trajectory for the compliant leg. With the same tolerance (0.001 m/s), the algorithm converges in four iterations rather than eight. This result suggests standard planning as a warm start heuristic for actuator-centric planning.

Figure 6 shows the results of modifying the input penalization parameter. The results show the expected decrease in optimal velocity after reaching a threshold value of α, as well as success in algorithm convergence for various tuned cost functions.

VI. Discussion

The proposed method for trajectory optimization offers several advantages. While not unprecedented, directly capturing all relevant state and input constraints is an essential feature for a dynamically consistent trajectory. Our new robot–actuator interface, modified by pseudo-mass M_p, allows us to exploit the structural difference between a linear actuator admittance and a nonlinear robot impedance—which is novel and efficient.

The performance of our approach depends strongly on the subproblem solver. Attention to details, including some mentioned in [15]—the particular parallelized sparse matrix inversion algorithm, problem formulation at the low-level solver interface, choice of programming environment, preprocessing of problem data, and solver—could potentially extend the approach to larger time scales, since CVX has trouble with the numerical conditioning of our subproblem beyond roughly $N = 75$, or roughly 720 variables.

Our actuator model could add on to any standard robot impedance model, including the very general constrained floating base model—and trajectories which are optimal for the original robot dynamics appear to be useful starting points for actuator-centered optimization.

ACKNOWLEDGMENT

The authors would like to thank Junhyeok Ahn, Orion Campbell, Donghyun Kim, Nicholas Paine, and Tom Gannon for their support.

References

[1] G. A. Pratt and M. M. Williamson, "Series elastic actuators," in *Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, vol. 1. IEEE, 1995, pp. 399–406.

[2] N. Paine, S. Oh, and L. Sentis, "Design and control considerations for high-performance series elastic actuators," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 3, pp. 1080–1091, 2014.

[3] K. Sreenath, H.-W. Park, I. Poulakakis, and J. W. Grizzle, "A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on MABEL," *The International Journal of Robotics Research*, vol. 30, no. 9, pp. 1170–1193, 2011.

[4] J. Pratt, T. Koolen, T. De Boer, J. Rebula, S. Cotton, J. Carff, M. Johnson, and P. Neuhaus, "Capturability-based analysis and control of legged locomotion, part 2: Application to m2v2, a lower-body humanoid," *The International Journal of Robotics Research*, vol. 31, no. 10, pp. 1117–1133, 2012.

[5] L. Chen, M. Garabini, M. Laffranchi, N. Kashiri, N. G. Tsagarakis, A. Bicchi, and D. G. Caldwell, "Optimal control for maximizing velocity of the compact compliant actuator," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 516–522.

[6] S. Haddadin, M. Weis, S. Wolf, and A. Albu-Schäffer, "Optimal control for maximizing link velocity of robotic variable stiffness joints," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 6863–6871, 2011.

[7] M. Garabini, A. Passaglia, F. Belo, P. Salaris, and A. Bicchi, "Optimality principles in variable stiffness control: The vsa hammer," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 3770–3775.

[8] S. Haddadin, F. Huber, and A. Albu-Schäffer, "Optimal control for exploiting the natural dynamics of variable stiffness robots," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 3347–3354.

[9] A. Radulescu, M. Howard, D. J. Braun, and S. Vijayakumar, "Exploiting variable physical damping in rapid movement tasks," in *Advanced Intelligent Mechatronics (AIM), 2012 IEEE/ASME International Conference on*. IEEE, 2012, pp. 141–148.

[10] D. Braun, M. Howard, and S. Vijayakumar, "Optimal variable stiffness control: formulation and application to explosive movement tasks," *Autonomous Robots*, vol. 33, no. 3, pp. 237–253, 2012.

[11] D. J. Braun, F. Petit, F. Huber, S. Haddadin, P. Van Der Smagt, A. Albu-Schäffer, and S. Vijayakumar, "Robots driven by compliant actuators: Optimal control under actuation constraints," *IEEE Transactions on Robotics*, vol. 29, no. 5, pp. 1085–1101, 2013.

[12] K. Sreenath, H.-W. Park, I. Poulakakis, and J. W. Grizzle, "Embedding active force control within the compliant hybrid zero dynamics to achieve stable, fast running on MABEL," *The International Journal of Robotics Research*, vol. 32, no. 3, pp. 324–345, 2013.

[13] A. Hereid, E. A. Cousineau, C. M. Hubicki, and A. D. Ames, "3D dynamic walking with underactuated humanoid robots: A direct collocation framework for optimizing hybrid zero dynamics," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1447–1454.

[14] F. L. Moro, N. G. Tsagarakis, and D. G. Caldwell, "A human-like walking for the COmpliant huMANoid COMAN based on CoM trajectory reconstruction from kinematic motion primitives," in *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*. IEEE, 2011, pp. 364–370.

[15] A. Werner, R. Lampariello, and C. Ott, "Trajectory optimization for walking robots with series elastic actuators," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*. IEEE, 2014, pp. 2964–2970.

[16] ——, "Optimization-based generation and experimental validation of optimal walking trajectories for biped robots," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 4373–4379.

[17] V. L. Orekhov, C. S. Knabe, M. A. Hopkins, and D. W. Hong, "An unlumped model for linear series elastic actuators with ball screw drives," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 2224–2230.

[18] S. Schütz, A. Nejadfard, C. Kötting, and K. Berns, "An intuitive and comprehensive two-load model for series elastic actuators," in *Advanced Motion Control (AMC), 2016 IEEE 14th International Workshop on*. IEEE, 2016, pp. 573–580.

[19] H. Asada and J. Leonard, *2.12 Introduction to Robotics*. Fall Massachusetts Institute of Technology: MIT OpenCourseWare, License: Creative Commons BY-NC-SA, 2005. [Online]. Available: https://ocw.mit.edu

[20] R. Schlossman, https://github.com/rschloss123, 2017.

[21] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," Mar. 2014.