2022 IEEE 5th International Conference on Soft Robotics (RoboSoft)
April 4-8, 2022, Edinburgh, UK

Model-Based Data-Driven System Identification and Controller Synthesis Framework for Precise Control of SISO and MISO HASEL-Powered Robotic Systems

Angella Volchko, Shane K. Mitchell, Timothy G. Morrissey, J. Sean Humbert

Abstract— **Soft robots require a complimentary control architecture to support their inherent compliance and versatility. This work presents a framework to control soft-robotic systems systematically and effectively. The data-driven model-based approach developed here makes use of Dynamic Mode Decomposition with control (DMDc) and standard controller synthesis techniques. These methods are implemented on a robotic arm driven by an antagonist pair of Hydraulically Amplified Self-Healing Electrostatic (HASEL) actuators. The results demonstrate excellent tracking performance and disturbance rejection, achieving a steady state error under 0.25% in response to step inputs and maintaining a reference orientation within 0.5 degrees during loading and unloading. The procedure presented in this work can be extended to develop effective and robust controllers for other soft-actuated systems without knowledge of their dynamics a priori.**

I. Introduction

Biological systems exploit soft structures to interact effectively with unpredictable complex environments and perform delicate operations [1], [2]. Inspired by nature's compliant behavior, soft technologies can be used to accomplish tasks that are not ideal for rigid mechanical solutions, such as human-machine interaction, manufacturing, manipulation, gripping, and locomotion [1]-[3]. Therefore, soft robots can benefit technologies in applications ranging from industrial automation to biomedical devices [1]-[3]. However, unlike their rigid mechanical competitors, soft robots were not designed to perform precise and repetitive motions and cannot adopt the same standard methods for control design [1], [2]. Due to their nearly infinite-dimensional nonlinear bodies [2], soft systems cannot be deterministically manipulated and thus, integrating them into the aforementioned industries remains a challenge. This paper develops and experimentally validates a systematic approach that can be used to synthesize feedback controllers to achieve prescribed performance requirements for soft robotic systems. Hydraulically Amplified Self-Healing Electrostatic (HASEL) actuators were selected as the soft actuator of choice for this work due to their high-performance and high-speed characteristics [4]. As a nascent technology, there is limited information on the dynamics of these actuators and their accompanying subsystems. Rothemund performs an analysis on the dynamics of HASEL actuators in [5]. However, this is only valid for a particular set of actuator materials, geometry, and operating conditions. To circumvent the paucity of information on HASEL dynamics, this paper exploits a controller design method that does not require any knowledge of the system a priori.

The five-step framework presented and demonstrated in this work combines the advantages of data-driven modelling techniques with model-based controller synthesis methods. Empirically determining the system dynamics allows us to repeat the procedure for various configurations of arbitrary soft robots without fully understanding the underlying dynamics of the system or deriving the governing equations of the system, while the model-based method allows us to exploit standard available linear controller synthesis strategies. Specifically, we use a method, known as Dynamic Mode Decomposition with control (DMDc), to determine an appropriate linear model for the system [6], [7]. After establishing an approximation of the system dynamics, we can employ standard linear controller synthesis techniques, including loop-shaping methods, Linear Quadratic Regulator (LQR)-based techniques, and robust μ ('mu')-synthesis approaches [8]-[11].

Implementing this framework, we demonstrate effective control of a robotic arm driven by an antagonist muscle pair of highly compliant, high-speed soft actuators known as Peano-HASEL actuators [12]. This paper is the first to illustrate how HASEL actuators can be built and controlled to mimic musculoskeletal systems; the actuators in this robotic system mimic the structure of the biceps-triceps muscles of the upper arm of humans. The two muscles exhibit co-contraction to vary the stiffness of the joint in the robotic arm. We transform an open-loop soft-actuated biceps-triceps apparatus into a closed-loop system that achieves user-specified orientations repeatedly in the presence of uncertainty.

We exploit the empirical nature of this framework on two separate cases: a single input single output (SISO) and a multiple input single output (MISO) version of the HASEL-driven robotic arm to achieve excellent tracking performance and disturbance rejection. The closed-loop MISO system

*This material is based upon work supported by U.S. Army Small Business Technology Transfer Program Office and Army Research Office Under Contract No. W911NF-21-P0040, National Science Foundation Under Grant No. EFMA-1830924, and Artimus Robotics

A. Volchko is with the Mechanical Engineering Department, University of Colorado, Boulder, CO 80309 USA (e-mail: Angella.Volchko@colorado.edu)

S. K. Mitchell is with Artimus Robotics Inc, Boulder, CO 80301 USA

T. G. Morrissey is with Artimus Robotics Inc, Boulder, CO 80301 USA

J. S. Humbert is with the Mechanical Engineering and Aerospace Engineering Departments, University of Colorado, Boulder, CO 80309 USA

2022 IEEE 5th International Conference on Soft Robotics (RoboSoft) | 978-1-6654-0828-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/ROBOSOFT54090.2022.9762220

was successful in response to step inputs, loading, and sudden perturbations. Importantly, this generic controller synthesis architecture can be extended to adapt to various HASEL morphologies with higher complexity, as well as other soft-robotic platforms.

The remainder of the paper is organized as follows. **Section II** presents a brief overview of HASEL actuators. **Section III** details the DMDc approach used in modelling the system. **Section IV** describes the five-step framework, as well as the experimental setup and methods used to model and control the HASEL-driven system. **Section V** highlights the results from the closed-loop systems. Finally, discussion of the research and ideas for future work are presented in **Section VI**.

II. Background of HASEL Actuators

Hydraulically Amplified Self-Healing ELectrostatic (HASEL) actuators combine the advantages of electrostatic stimulus with principles of hydraulic scaling to result in a versatile, compliant, and high-performance soft actuator [4]. HASEL actuators consist of a pair of opposing electrical conductors printed on a thin polymer shell that is filled with a dielectric fluid [4]. When a potential difference is applied across the opposing electrodes, an electrostatic force develops that causes the electrodes to "zip" or pull together and force the dielectric fluid to the region of the shell that is not covered in electrodes, thereby morphing the overall 3D shape of the actuator [4].

HASEL actuators can be designed and fabricated in a variety of geometries and sizes and assembled into various configurations [13]. Moreover, HASEL actuators can achieve all three basic modes of actuation (contraction, expansion, and rotation) [4], [12], [13]. The HASEL variation used in this work is a 15-unit Peano-HASEL (Part #: C-5020-15-01-C-BAAC-50-140, Artimus Robotics). These actuators were stacked in parallel to scale the contractile force under the same applied voltage. In addition to their customizability, HASEL actuators exhibit many other desirable traits that set them apart from other soft actuators. Most notably, HASEL actuators have a high bandwidth and precise operation. Quick response times and precise actuation of HASEL actuators have been demonstrated in the works of [12], [14]. HASEL actuators have been shown to actuate and perform upwards of 40 Hz [12], [14]. They also can be reproduced from affordable materials and accessible fabrication techniques. By locally displacing fluid within a soft structure, HASEL actuators do not require an external source of compressed fluid, and therefore, are not limited in portability and speed [12]. Additionally, they are fault tolerant having shown the capability to recover from dielectric breakdown, since the dielectric fluid can reflow to restore insulating conditions [4].

This work is the first to demonstrate precise closed-loop control of multiple channels of HASEL actuators to result in a single desired system response. Initial demonstrations of HASEL actuators, like those shown in [4], [12] use open loop control which is tuned to the repeatable task in a controlled environment. If HASEL actuators are to be used in real-world applications, they need excellent tracking performance and disturbance rejection characteristics, which cannot be offered by open-loop control methods. The work of Johnson et al. demonstrated effective closed-loop control of a single foldable HASEL actuator utilizing linear frequency response tests [14]. Additionally, work was conducted in [15] to model and control the soft actuators with a mass-spring-damper modelling strategy. However, to promote the customizability of HASEL actuators for their use in a wide range of applications, there needs to be a complimentary framework for systematically modelling and synthesizing controllers for the various morphologies, configurations, and assemblies of HASEL actuators.

III. Background of Dynamic Mode Decomposition with Control

Dynamic Mode Decomposition with control (DMDc) is a useful tool that can be used to model the complex and nonlinear dynamics of soft robots [7]. Dynamic Mode Decomposition (DMD) is a model reduction method that uses empirical snapshots of a system and its inputs to approximate the underlying dynamics of systems with a linear representation [6]. DMD with control (DMDc) is an extension of DMD and disambiguates the system's intrinsic dynamics from its response to actuation [7]. The following equation shows how a system's current state, $\boldsymbol{x}$, and control inputs, $\boldsymbol{u}$, are linearly mapped through a dynamic matrix, A and a control input matrix, B, respectively, to result in the change in state, $\dot{\boldsymbol{x}}$.

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{u} \tag{1}$$

Like DMD, DMDc is an approach that uses pairs of data vectors to extract the underlying low-dimensional dynamic characteristics of nonlinear systems. DMDc is also closely tied to the Koopman operator theory, as DMD approximates a linear operator to describe the dynamics of a nonlinear system [6], [7], [16]. As defined in [17], for any finite-dimensional Lipschitz continuous nonlinear system, there is an equivalent infinite dimensional linear representation (the Koopman operator) in the space of all scalar-valued functions of the system's state. Given that the Koopman operator is infinite dimensional, it is not practical in real-world applications. However, we can approximate the infinite dimensional operator as an n-dimensional matrix. As n approaches infinity, we approach the exact dynamics of the nonlinear system with a linear operator. One extension of DMD that bridges the relation between Koopman and DMDc is extended DMD (eDMD), in which the user can choose the dictionary of observables to better approximate their nonlinear system with an expanded state space. [16]

Employing DMDc to model our system exploits the advantages of data-driven methods with model-based controller synthesis techniques. DMDc bypasses the time-consuming and demanding effort to derive the system's equations of motion since it only requires empirical snapshots of the system state and external inputs. Since the nonlinearities and highly complex intrinsic dynamics of the HASEL actuators make it difficult to obtain a model through first principles and since we aim to synthesize controllers for various HASEL types and configurations, the data-driven approach is ideal. Thus, through this empirically driven method, we can readily reformulate the estimated dynamics of the system with each variation of HASEL morphology.

Many data-driven learning methods result in 'black-box' dynamics which do not lend themselves to standard controller synthesis methods. However, DMDc results in a linearized model of the system, and thus enables available model-based controller synthesis techniques.

DMDc can also reduce the order of high dimensional models by projecting them onto lower dimensional subspaces [6], [7]. However, the system controlled in this paper did not use a high dimensional state, and therefore the model reduction part of DMDc was not used.

The algorithm for DMDc can be described with the following steps as it was introduced in [7]. First, the experimentalist selects the observables and external inputs to describe the state of the system and actuation, respectively. The observables of the system are stacked into a snapshot vector, $\boldsymbol{x}_k$, and are recorded at each time step, $k\Delta t$ with $k \in [0, 1, 2, \ldots\ m-1]$. The state snapshot matrix is built by concatenating multiple snapshot vectors:

$$\mathrm{X} = [\boldsymbol{x}_1\ \boldsymbol{x}_2 \ldots \boldsymbol{x}_{m-1}]$$

Similar to the state snapshot matrix, the control snapshot matrix consists of vectors that are stacked actuation inputs to the system, $\boldsymbol{u}_k$:

$$\Upsilon = [\boldsymbol{u}_1\ \boldsymbol{u}_2 \ldots \boldsymbol{u}_{m-1}]$$

Lastly, the state snapshot matrix is copied and shifted by Δt. The time-shifted snapshot matrix is denoted

$$\mathrm{X}' = [\boldsymbol{x}_2\ \boldsymbol{x}_3 \ldots\ \boldsymbol{x}_m]$$

We then concatenate X and Υ:

$$\Omega = \begin{bmatrix}\mathrm{X}\\ \Upsilon\end{bmatrix}$$

We also concatenate the dynamic and control input matrices:

$$G = [\,A\ \ B\,]$$

Then, we formulate the discrete dynamics equation:

$$\mathrm{X}' = A\mathrm{X} + B\Upsilon \tag{2}$$

and we rewrite (2) in terms of stacked matrices:

$$\mathrm{X}' = G\Omega \tag{3}$$

We perform a pseudoinverse on Ω via a singular value decomposition (SVD):

$$\Omega \approx \mathrm{U}\Sigma\mathrm{V}^*$$

so that we can solve:

$$\mathrm{G} = \mathrm{X}'\Omega^\dagger \tag{4}$$

where † represents the pseudoinverse and p defines the truncation value of the SVD. Plugging in the pseudoinverse of Ω into (4), we can evaluate G:

$$\mathrm{G} = \mathrm{X}'\mathrm{V}\Sigma^{-1}U^* \tag{5}$$

and equate an approximation for the dynamic and input matrices:

$$[\,\mathrm{A}\ \ \mathrm{B}\,] \approx [\,\mathrm{X}'\mathrm{V}\Sigma^{-1}\mathrm{U}_1^*,\ \ \mathrm{X}'\mathrm{V}\Sigma^{-1}\mathrm{U}_2^*\,] \tag{6}$$

where

$$U^* = [\mathrm{U}_1^*,\ \mathrm{U}_2^*]$$

for which $\mathrm{U}_1^* \in \mathbb{R}^{n\times p}$ and $\mathrm{U}_2^* \in \mathbb{R}^{l\times p}$. Reduction methods can be used at this point following the algorithm outlined in [7]. However, due to the limited number of observables selected to describe the state of the robotic system in this work, reduction methods were not used.

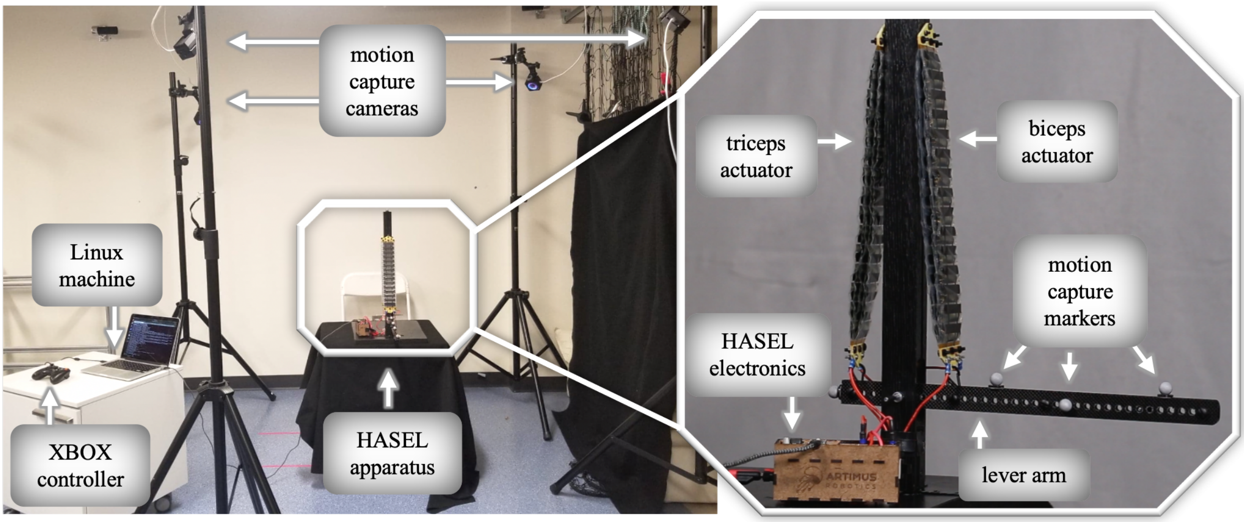

Figure 1: Experimental setup for both data collection and controller validation (left) and a detailed view of the HASEL arm apparatus (right).

IV. Methods

The experimental setup is depicted in Fig. 1. The robotic arm was comprised of a carbon fiber lever arm connected with a pin to a stationary wooden column secured to a wooden base. The lever arm was free to rotate about the pin joint with one main degree of freedom. The actuators attached to the shorter side of the lever arm and those attached to the opposite side of the lever arm will be referred to as the triceps and the biceps of the robotic system, respectively. These antagonist muscle-like actuators both consisted of five 15-unit contracting Peano-HASELs secured at the top with a nut and bolt and at the lever arm with a carabiner. Brass weights of varying masses were used as loads on the arm and were anchored to the end of the lever arm at 32.3 cm from the fulcrum. The biceps actuator was anchored 6.6 cm from the fulcrum and operated as a third-class lever with a mechanical advantage of 0.204, while the triceps actuator was anchored 5.7 cm from the fulcrum and operated as a first-class lever with a mechanical advantage of 0.176. Four OptiTrack Prime13W motion capture cameras were placed around the system to capture the change in angle of the lever arm with a frame rate of 240 Hz. Five motion capture retroreflective markers were adhered along the length of the rigid lever arm for tracking purposes. The electronics package used to drive the actuators can be purchased off-the-shelf from Artimus Robotics Inc. (Part #: HVPS-10-30-02-A-00). It consists of a microcontroller connected to proprietary circuitry which allows the Peano-HASEL actuators to charge (contract) and discharge (relax). Each actuator was controlled independently via pulse-width modulated (PWM) signals that ranged in duty cycle from 0-100% (an 8-bit value of 0-255), which corresponds to the charging rate of the actuator. The input power to the electronics was 24VDC. Lastly, the XBOX controller, was connected to the Linux machine via a wired USB to allow the user to command PWM signals to the system during data extraction and reference angles to the system during controller validation.

We used Robotic Operating System (ROS) on the Linux machine to record data and communicate between the motion capture system, microcontroller, and XBOX controller. ROS offers a platform that aids in communicating between various electronic devices, as those described above, in real time. ROS communicated with the microcontroller and the XBOX controller over USB serial and with the OptiTrack motion capture system over a wireless network connection.

The five-step framework used to develop the closed loop control laws for the soft actuator system is outlined as follows:

1. A physical system with unknown dynamics is acquired. The plant must be equipped with a device to manipulate its inputs and a sensor to interpret its response.
2. After the plant is set up with a means for actuation and sensing, arbitrary commands are sent to all actuators within the system and inputs and observables are recorded.
3. Next, the recorded snapshots get pushed into the DMDc architecture described in Section III to approximate the system's dynamics.
4. After the dynamic model is established, standard model-based controller synthesis techniques are determined, and the controller is designed.
5. Lastly, the synthesized controller is integrated into the physical plant.

The framework was tested on two variations of the soft actuator driven robotic arm. The initial test consisted of actuation of a single channel (charge PWM signal to the biceps), while the second validation of the framework used two control actuation channels, (independent charge PWM signal to the biceps and triceps).

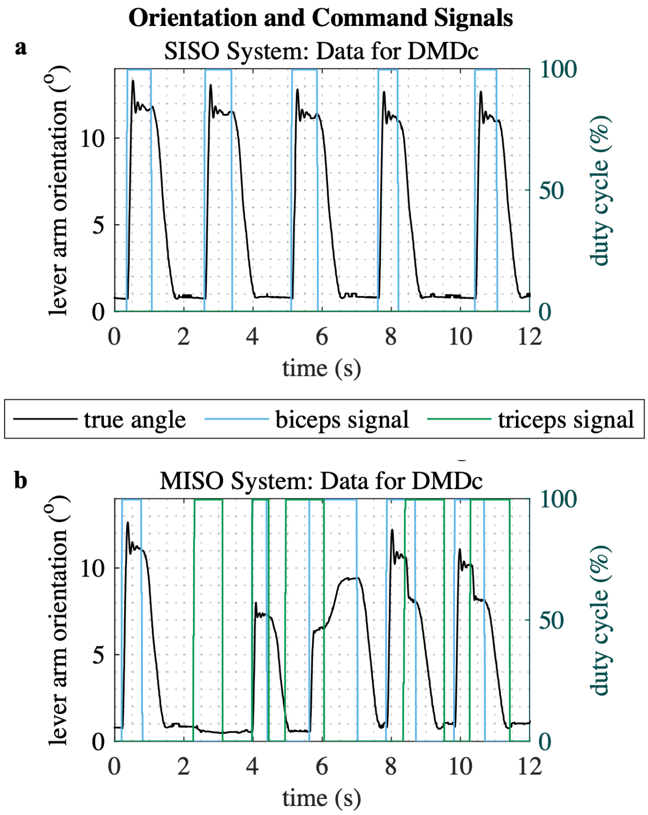

Figure 2: Data collected from system to be used in DMDc algorithm for approximating linear model of the dynamics of the system. The axis on the left shows values for the angle of the lever arm (black) in degrees. The right axis shows the percentage of the duty cycle of the PWM signal sent to the biceps channel (blue) and the triceps channel (green). The top plot (a) displays the data used to approximate the model of the single input single output system, and the bottom plot (b) shows the data used to approximate the model of the multiple input single output system.

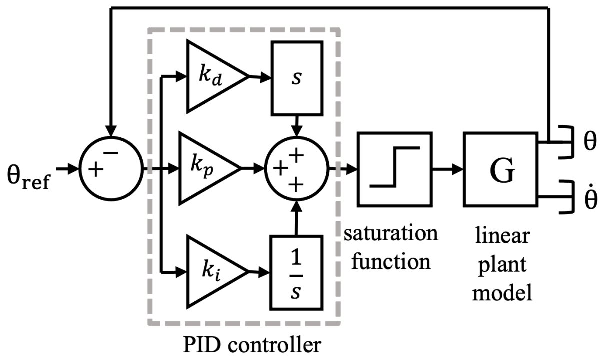

Figure 3: A representation of the block diagram of the PID control law made in simulation with the linearized model of the system, G. This block diagram represents the control law as it was implemented into the single input single output system. The saturation function was required in simulation to limit the PWM signals [0-255] sent to the actuators. The outputs of our system, θ and $\dot{\theta}$, represent the lever arm's current angular position and velocity, respectively. The PID controller takes an input of the difference between user-defined reference angle, θ_{ref} and the current lever arm angle and sums together the proportional, integral, and derivative terms. The proportional, integral, and derivative gains, (k_p, k_i, k_d) are multiplied by their corresponding functions, where the derivative and integral in the Laplace domain are represented by s and 1/s, respectively.

A. Control of SISO System

Initially, arbitrary PWM signals were sent to the biceps to actuate and affect the state of the system. The PWM values and angles were recorded with a timestamp in ROS. An interval of 12 seconds of data, shown in Fig. 2a, was selected and interpolated with a timestep of 0.01 seconds to be used in the DMDc algorithm. In addition to the recorded angle, the angular velocity was calculated at every time step and added as an observable to the state of the system such that the snapshot matrix had two rows:

$$X = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}$$

The state snapshot matrix at each timestep was stacked up and pushed into the DMDc architecture. The DMDc matrix calculations described in section III were performed in MathWorks' MATLAB offline and resulted in the following continuous dynamic and control input matrices:

$$A = \begin{bmatrix} 0.53 & 1.12 \\ -102.14 & -23.28 \end{bmatrix} \qquad B = \begin{bmatrix} -0.30 \\ 5.61 \end{bmatrix}$$

The following proportional, integral, and derivative gains (k_p, k_i, k_d respectively) were selected for the PID controller architecture depicted in Fig. 3:

$$k_p = 40 \qquad k_i = 200 \qquad k_d = 0$$

Loop shaping methods, as depicted in Fig. 4, were used to determine these gains following the procedure described in [8]. Addition of a derivative term was avoided due to noise in the system. The shaped frequency responses can be visualized in Fig. 4 along with estimated performance values. The performance measurements described here are not guaranteed on the physical system as the model used is only an approximation of the true system dynamics. The SISO controller was simulated in MathWorks' Simulink following the block diagram shown in Fig. 3 to understand the effects of saturating the PWM control signal with upper and lower bounds [0-255]. Finally, the control law was implemented in ROS using the motion capture orientation information as feedback.

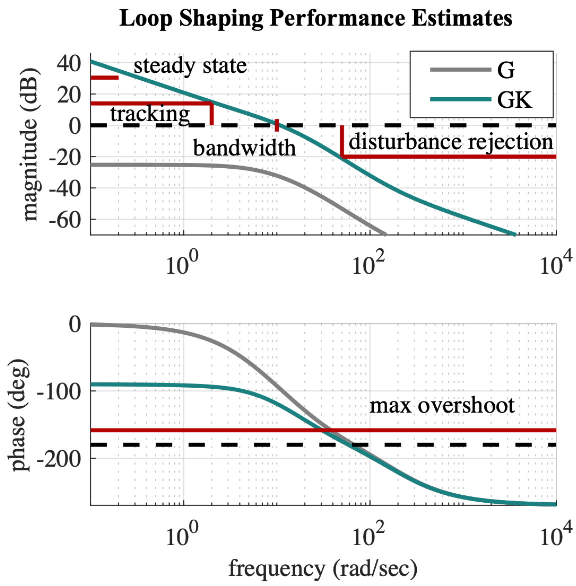

Figure 4: A frequency response plot that uses open loop shaping methods and the estimated plant model to indicate the estimated performance of our system without a controller, G (grey), and with the added PID controller GK (teal). These bode plots helped us find a PID controller that balanced the following performance specifications: minimal steady state errors and tracking errors at low frequencies, high closed-loop bandwidth, maximal disturbance rejection at high frequencies and minimal percent overshoot. Åström and Murray describe this loop shaping controller synthesis technique and the useful calculations [8]. The red lines indicate estimates of the performance of our closed loop system. The bode plot suggests that the closed-loop system will have a steady state value less than 3%, a tracking error less than 20% up to 2 rad/s, a bandwidth of 10 rad/s, a disturbance rejection of 10 times over 50 rad/s, and a maximum overshoot of 30%. Due to the linear approximation of the true system and saturation functions that were not accounted for in this loop shaping tool, these performance guarantees are only estimates of the final closed loop physical system.

B. Control of MISO System

Following the framework presented, first, PWM signals were sent to both the biceps and triceps channels to actuate and affect the state of the system. The PWM values and lever arm angles were recorded in ROS with their corresponding timestamp. Again, an interval of 12 seconds of data (Fig. 2b) was selected and interpolated with a timestep of 0.01 seconds to be used in the DMDc algorithm.

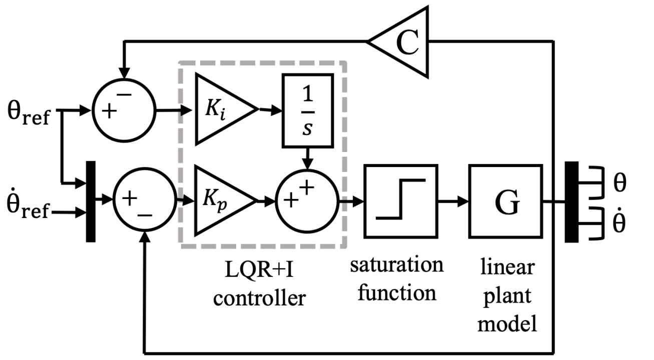

Figure 5: A high level representation of the block diagram of the LQR + Integral action control law made in simulation with the linearized model of the system. The block diagram provides context for the proportional and integral gain matrices (K_p, K_i) calculated in the methods section and represents the control law as it was implemented into the multiple input single output system. The user-defined values include the angular position, θ_{ref}, and velocity, $\dot{\theta}_{ref}$, of the system. These values are compared to the true state of the system, θ and $\dot{\theta}$, and the difference is sent to the LQR+I controller gain matrices calculated in the methods section. Multiplying by C reduces the total states to the angular position. As with the SISO system, a saturation function is applied to the command signal to limit the PWM output from the controller. These values are sent to the linearized plant model, G.

In addition to the angle, the angular velocity, angular acceleration, and biceps charge information were included as additional observables to describe the state of the system:

$$X = \begin{bmatrix} \theta \\ \dot{\theta} \\ \ddot{\theta} \\ \mathrm{PWM_{biceps}} \end{bmatrix}$$

These data vectors were stacked up and pushed into the DMDc architecture again using MATLAB. DMDc resulted in the following continuous dynamic and control input matrices:

$$A = \begin{bmatrix} -1.53 & 0.85 & 0.22 & -0.13 \\ -894.58 & -120.65 & 13.98 & -68.29 \\ 3.75 & 0.0279 & -70.53 & 6.03 \\ \sim 0 & \sim 0 & \sim 0 & \sim 0 \end{bmatrix}$$

$$B = \begin{bmatrix} -0.13 & -0.02 \\ 36.09 & -15.93 \\ 68.84 & -0.94 \\ \sim 0 & \sim 0 \end{bmatrix}$$

The step responses of these matrices were plotted to understand the relationship between rows and states. From the step responses, it is evident that the first three states represent angular position, angular velocity, and biceps charge, respectively, and thus, the state corresponding to the row of near zeros is angular acceleration and can be omitted with negligible effect. The step responses also show that the first and second column in the input matrix represent the biceps and triceps input signal, respectively. Following an attempt of solely using LQR synthesis, integral action was determined to be critical in effectively controlling the system.

Hence, we used an LQR + Integral action (LQR+I) controller to close the loop of the MISO system [9] using the matrices $Q = diag(100, 0.01, 0, 800)$ and $R = diag(0.01, 0.01)$. Matrix Q represents a penalty on the error in state, where the last value corresponds to the penalty on the integrator error, while matrix R determines a penalty on actuator efforts. The gain matrices were calculated following the procedure in [9] and the values corresponding to the angular position and velocity were used in the control architecture depicted in Fig. 6, as the biceps error signal was zero.

$$K_p = \begin{bmatrix} 82.509 & 0.909 \\ -34.149 & -0.376 \end{bmatrix} \qquad K_i = \begin{bmatrix} 261.346 \\ -108.158 \end{bmatrix}$$

Again, results of the system with feedback were simulated in MathWorks' Simulink with added saturation to ensure that the controller accounted for the limited range of allowable PWM signals. Finally, the control law was implemented in ROS following the architecture depicted in Fig. 5.

V. Results

To validate the controllers developed, the user generated desired reference angles of the lever arm with discrete step size inputs using the XBOX controller. Upon generating a reference orientation, the system was demonstrated to track the desired angle and maintain the desired angle while experiencing disturbances, such as loading, unloading, and sudden perturbations.

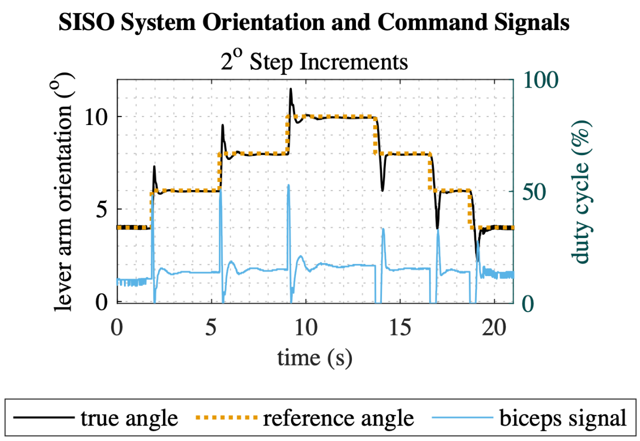

Figure 6: The left axis provides the value of the true orientation (black) of the single input single output system in response to the user-defined reference angle (orange) which step up and down by 2-degree increments. The right axis displays the percentage of duty cycle of PWM signal to the biceps channel that was commanded by the controller.

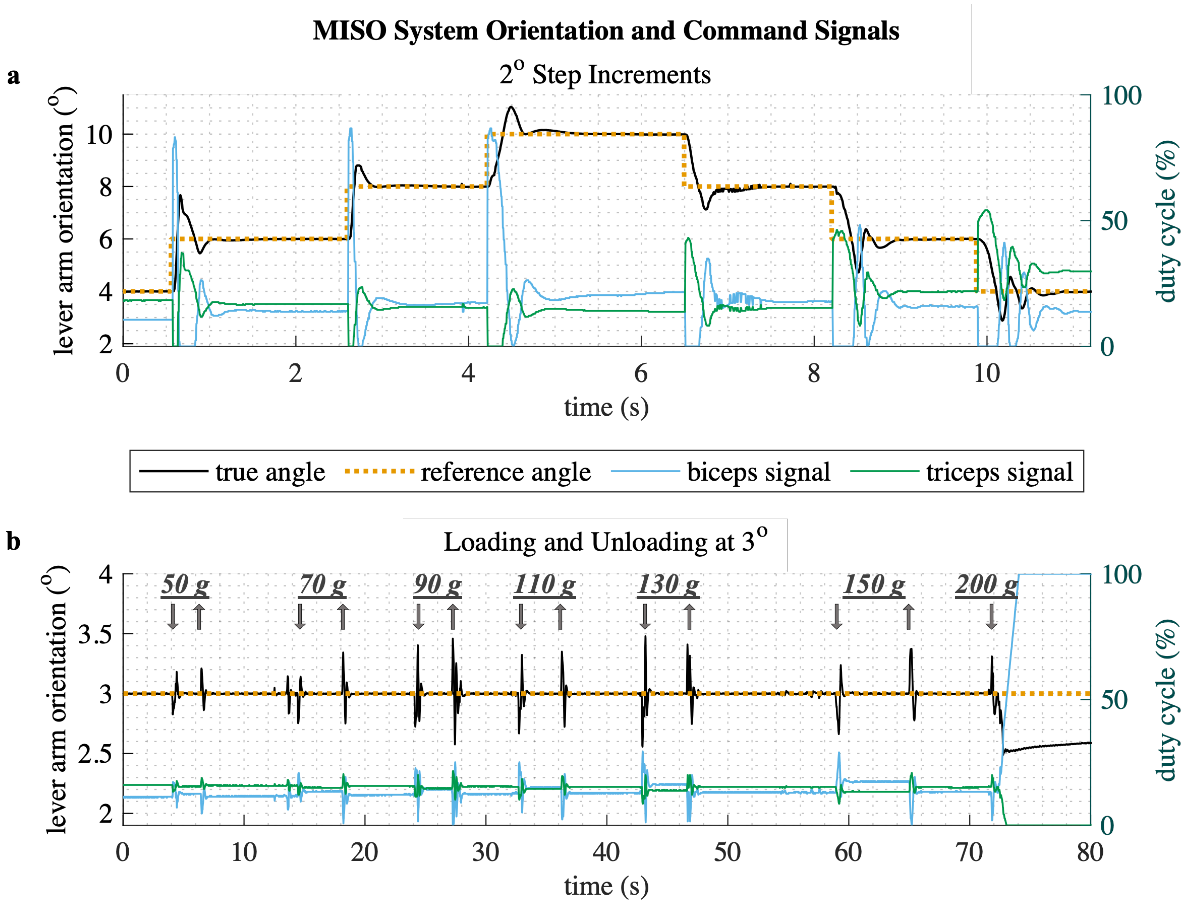

Figure 7: Results of the multivariable controller demonstrating the responses of the multiple input multiple output system to step inputs (a) and under conditions of loading and unloading (b). The left axis describes the orientation prescribed to the system by the user (orange) and the true orientation of the system (black). The right axis provides the percentage of duty cycle of PWM signals to the biceps (blue) and triceps (green) channels. In the top plot (a) 2-degree incremental step inputs are commanded to the system by the user. In the bottom plot (b), the system is commanded to maintain an angle of 3 degrees by the user, and weights (50 - 200 grams) are added to the lever arm (downward-facing arrow) and removed from the lever arm (upward-facing arrow) by the user.

A. Results from PID Controller for SISO System

Fig. 6 illustrates how the SISO system with closed-loop PID control responded to 2-degree incremental step inputs. The step responses varied, due to the varying conditions of the HASEL actuators in different operating ranges of actuation. Analyzing the step from 4 to 6 degrees, the SISO system attained a rise time of 0.05 seconds, a settling time of 1.9 seconds, an overshoot of 65% and a steady state error of 2%.

B. Results from LQR+I Controller for MISO System

Fig. 7a illustrates how the multivariable system with LQR+I control responded to 2-degree incremental step inputs. Again, the step responses varied as the conditions of the HASEL actuators changed throughout the operating ranges of actuation. Again, inspecting the step response from 4 to 6 degrees, the antagonist muscle apparatus integrated with an LQR+I control law achieved a rise time of 0.02 seconds, a settling time of 0.8 seconds, an overshoot of 80%, and a steady state error of 0.25%. The figure shows that the PWM signals from the antagonist muscle pair oscillate 180 degrees out of phase from each other. In addition to tracking a reference input, the system was tested under external disturbances as depicted in Fig. 7b from loading and unloading conditions. We demonstrated that closed loop control maintained an angle within 0.5 degrees of the reference angle up to a 150-gram weight and returned to within 0.1 degrees of the reference angle in under one second each time. Note that at 200 grams the actuators are unable to maintain the reference angle and the biceps attempt to contract more and the charge input to the triceps drops to zero. Responses of the closed-loop multivariable system, including the system's response to sudden human-induced perturbations, can be found in the supplementary video.

VI. Discussion and Future Work

Through the controller design procedure presented in this paper, we have demonstrated that we can synthesize and implement an effective controller for a HASEL-driven robotic arm. Both the PID and multivariable controllers proved satisfactory when tracking a reference input. The multivariable system was advantageous in comparison to the SISO system as the antagonist muscle pair worked together to dampen the system and help attain the desired position more quickly. The multivariable controller proved to be effective against sudden disturbances and loading and unloading conditions, as well.

Work needs to be done to determine the best choice of time interval, Δt, for the DMDc algorithm. Minimal changes in the time intervals severely hindered the model and resulted in unrealistic step responses. Further investigation also needs to be conducted on expanding the observable space and understanding when a model contains an adequate number of states to describe the nonlinearities of the system properly. Since the controllers proved to be effective, expanding the dictionary of observables was not explored in this research. The controllers integrated into the biceps-triceps mechanism were deemed effective, but to optimize the overall performance of the closed-loop system, modifications should be made in the following three areas: mechanical design, modelling, and controller synthesis.

Since the controller synthesis framework developed during this project does not require any information about the system a priori, it can be expanded and adapted to future morphologies of HASEL-driven robots, including those with more degrees of freedom or alternative actuator geometries. More complex soft-robotic system designs should be explored in future work to validate this, including those with extensive state measurements so that the reduction aspect of DMDc can be exploited.

Future work can use extensions of DMD, such as eDMD, to better capture the intricate nonlinear dynamics of the HASEL actuator systems and thus, optimize the controller. Attaining a more accurate linear model of the nonlinear physical plant will promote the use of the standard controller synthesis methods used in this work, so that the performance estimates are representative of the closed-loop system.

Furthermore, future work should utilize other model-based controller design methods, such as h-infinity synthesis and μ-analysis that will provide added measures of robustness. The innate compliance of the soft materials that make up HASEL actuators require robust control laws that can compensate for any inconsistencies in the performance of the actuators over time or from system to system.

References

[1] El-Atab, Nazek, Rishabh B. Mishra, Fhad Al-Modaf, Lana Joharji, Aljohara A. Alsharif, Haneen Alamoudi, Marlon Diaz, Nadeem Qaiser, and Muhammad Mustafa Hussain. 2020. "Soft Actuators for Soft Robotic Applications: A Review." *Advanced Intelligent Systems* 2 (10): 2000128.

[2] Kim, Sangbae, Cecilia Laschi, and Barry Trimmer. 2013. "Soft Robotics: A Bioinspired Evolution in Robotics." *Trends in Biotechnology* 31 (5): 287–94.

[3] Thuruthel, Thomas George, Yasmin Ansari, Egidio Falotico, and Cecilia Laschi. 2018. "Control Strategies for Soft Robotic Manipulators: A Survey." *Soft Robotics* 5 (2): 149–63.

[4] Acome, S. K. Mitchell, T. G. Morrissey, M. B. Emmett, C. Benjamin, M. King, M. Radakovitz, and C. Keplinger. 2018. "Hydraulically Amplified Self-Healing Electrostatic Actuators with Muscle-like Performance." *Science* 359 (6371): 61.

[5] Rothemund, Philipp, Sophie Kirkman, Christoph Keplinger. "Dynamics of electrohydraulic soft actuators". Proceedings of the National Academy of Sciences 117(28) 16207-16213 (2020).

[6] Tu, Jonathan H., Clarence W. Rowley, Dirk M. Luchtenburg, Steven L. Brunton, and J. Nathan Kutz. 2014. "On Dynamic Mode Decomposition: Theory and Applications." *Journal of Computational Dynamics* 1 (2): 391–421.

[7] Proctor, Joshua L., Steven L. Brunton, and J. Nathan Kutz. 2016. "Dynamic Mode Decomposition with Control." *SIAM Journal on Applied Dynamical Systems* 15 (1): 142–61.

[8] Åström, Karl Johan, and Richard Murray. 2008. "Chapter Eleven: Frequency Domain Design." In *Feedback Systems*, 315-46. Princeton University Press.

[9] Reis, Francisco, R.P. Torrico-Bascope, and Marcus Costa. 2011. "LQR Control with Integral Action Applied to a High Gain Step-Up DC-DC Converter." In , 256–61.

[10] Skogestad, Sigurd, Ian Postlethwaite, and A. Cemal Eringen. 1996. *Multivariable Feedback Control: Analysis and Design*. New York; Chichester: Wiley

[11] Doyle, John Comstock, Bruce A. Francis, and Allen Tannenbaum. 1992. *Feedback Control Theory*. New York; Toronto: Macmillan Pub. Co.

[12] Kellaris, Nicholas, Vidyacharan Gopaluni Venkata, Garrett M. Smith, Shane K. Mitchell, and Christoph Keplinger. 2018. "Peano-HASEL Actuators: Muscle-Mimetic, Electrohydraulic Transducers That Linearly Contract on Activation." *Science Robotics* 3 (14): eaar3276.

[13] Mitchell, Shane K., Xingrui Wang*, Eric Acome*, Trent Martin, Khoi Ly, Nicholas Kellaris, Vidyacharan Gopaluni-Venkata, Christoph Keplinger. "An Easy-to-Implement Toolkit to Create Versatile and High-Performance HASEL Actuators for Untethered Soft Robots". Advanced Science 6(14) 1900178 (2019).

[14] Johnson, Brian K., Vani Sundaram, Mantas Naris, Eric Acome, Khoi Ly, Nikolaus Correll, Christoph Keplinger, James Sean Humbert, and Mark E. Rentschler. 2020. "Identification and Control of a Nonlinear Soft Actuator and Sensor System." *IEEE Robotics and Automation Letters* 5 (3): 3783–90.

[15] Schunk, Cosima, Levi Pearson, Eric Acome, Timothy G. Morrissey, Nikolaus Correll, Christoph Keplinger, Mark E. Rentschler, and J. Sean Humbert. 2018. "System Identification and Closed-Loop Control of a Hydraulically Amplified Self-Healing Electrostatic (HASEL) Actuator." In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 6417–23.

[16] Li, Qianxiao, Felix Dietrich, Erik M. Bollt, and Ioannis G. Kevrekidis. 2017. "Extended Dynamic Mode Decomposition with Dictionary Learning: A Data-Driven Adaptive Spectral Decomposition of the Koopman Operator." *Chaos: An Interdisciplinary Journal of Nonlinear Science* 27 (10): 103111.

[17] Bruder, Daniel, Xun Fu, R. Brent Gillespie, C. David Remy, and Ram Vasudevan. 2021. "Data-Driven Control of Soft Robots Using Koopman Operator Theory." *IEEE Transactions on Robotics* 37 (3): 948–61.