Baloo: A Large-Scale Hybrid Soft Robotic Torso for Whole-Arm Manipulation

Curtis C. Johnson, Andrew Clawson, Marc D. Killpack

Abstract—Soft robotic actuators and their inherent compliance can simplify the design of controllers when operating in contact-rich environments. With such structures we can accomplish high-impact, dynamic, and contact-rich tasks that would be difficult using conventional rigid robots which might either break the robot or the object without careful modeling and design of high bandwidth controllers. In order to explore the benefits of structural passive compliance and exploit them effectively, we present a prototype robotic torso named Baloo, designed with a hybrid rigid-soft methodology, incorporating both adaptability from soft components and strength from rigid components. Baloo consists of two meter-long, pneumatically-driven soft robot arms mounted on a rigid torso and driven vertically by a linear actuator. We explore some challenges inherent in controlling this type of robot and build on previous work with rigid robots to develop a joint-level neural-network adaptive controller to enable high performance tracking of highly nonlinear, time-varying soft robot dynamics. We also demonstrate a promising use case for the platform with several hardware experiments performing whole-body manipulation with large, heavy, and unwieldy objects. A video of our results can be viewed at https://youtu.be/eTUvBEVGKXY.

Index Terms—soft robot, hybrid rigid-soft robots, whole-arm manipulation, adaptive control

I. INTRODUCTION

ROBOTIC manipulation has been studied extensively for decades and remains a challenging problem–especially when facing 'open-world' manipulation tasks with objects of varying shapes, sizes, and weights in vastly differing environments. Some objects in the open world can be very difficult to manipulate with the traditional end-effector-only paradigm without using a stronger and larger robot. This is especially that case for unknown, large, or heavy objects. In this work we advocate for a more general manipulation paradigm that includes the entire structure of the robot, instead of just the end effector. This style of manipulation is known as whole-arm or whole-body manipulation and has the potential to dramatically increase both the quantity and variability of manipulable objects in the open world.

When operating in open-world environments, a successful manipulation system should at least be robust to uncertainty. Better still is the ability to actively learn and adapt to unknown objects and environments. To accomplish this, it is often useful for the robot to exhibit compliant behavior. Compliant interactions can be implemented on rigid robots with active compliance/impedance control [1], [2], but we propose that
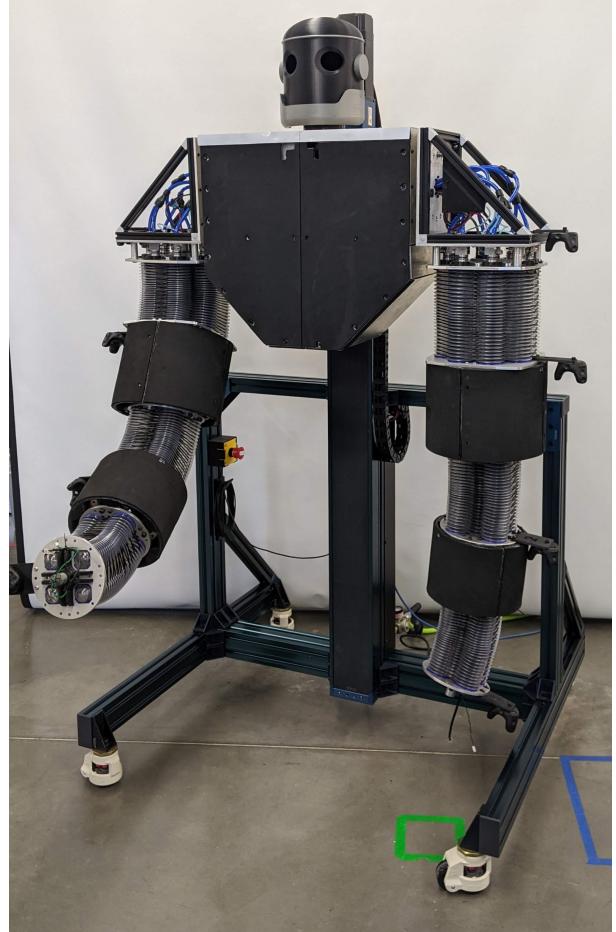
Fig. 1. Photo of Baloo with one arm extended forwards. HTC Vive trackers are mounted on each joint end plate for joint configuration estimation.

exploiting the passive compliance of soft materials has some additional unique benefits that are particularly useful in whole-arm, open-world manipulation tasks.

One approach is to add passive (i.e. mechanical) compliance to an otherwise rigid robot. The authors of [3] were among the first to recommend outfitting the WAM robot arm with high-friction, compliant material for whole-arm manipulation. Recent research supports this idea in several ways. The authors of [4], [5] use the rigid Kinova Jaco Gen2 robot arm outfitted with compliant pneumatic structures to accomplish whole-arm grasping of large objects. Other researchers have added passive compliance to the end effector itself [6] to accomplish tasks like bi-manual manipulation of a large box [7].

Another alternative approach is to soften the structure of

the robot itself. Because of the passive (i.e. mechanical) compliance built into the system, soft robots can excel in highly uncertain environments like contact-rich manipulation, human robot collaboration, or tasks involving impacts without the risk of damage to the robot or to the environment [8], [9]. Soft robots offer several advantages over rigid robots in these situations, often with surprising robustness [10]. This robustness is possible because a soft structure performs a similar role as a low-level impedance controller, but acts in a decentralized manner without any additional sensors or actuators. In addition, compliance-based control actions can have zero time delay. Soft structures can store potential energy for highly dynamic tasks [11] and can exhibit stable oscillations, which can be very useful for locomotion [12]. Collectively, this idea of outsourcing control, sensing, and computational functionality onto hardware is known as mechanical intelligence [4] or embodied intelligence [13].

As many state-of-the-art soft robot platforms are designed on the millimeter to sub-meter scale, there are still significant challenges with scale and size [14]. Recent research (see [9], [15]) suggests that incorporating both soft and rigid components into the structure of a robot can help solve essential manipulation-related problems, as the rigid components can carry heavier loads while the compliant components preserve passive compliance and adaptability.

While incorporating soft components into robotic systems brings several benefits, these advantages come with their own set of challenges. Soft materials often exhibit non-ideal properties such as nonlinear dynamical behavior and time-dependent wear and fatigue. These material characteristics change over time due to repeated use, environmental conditions, and inherent material limitations. Typical actuation methods for soft robots (e.g. fluidic, cable-driven, etc.) suffer from significant challenges in scale, modeling, control, and manufacturing [16].

To leverage the benefits of using soft materials we need control methods that can handle these challenges. Interest in developing control methods for soft robotics is growing. Some control techniques such as Model Predictive Control [17], [18] rely on precise models of the robot's dynamics [12], but obtaining accurate models for systems incorporating soft materials is particularly challenging due to their inherent variability and complexity. On the other hand, there are data-driven modeling and control techniques [19] which attempt to learn a model or controller directly from data using reinforcement learning [20], [21], machine/deep learning [22], [23], or the Koopman operator [24]. These approaches are difficult because of the temporal uncertainty of soft robots, which can necessitate gathering and re-gathering data frequently. Therefore, a control method that can tolerate both model inaccuracies and does not require large amounts of training data or training time is ideal.

Adaptive control methods can potentially solve the problem of time-varying dynamics by dynamically adjusting to the changing properties of soft materials. Such controllers can adapt in real-time to account for both static and dynamic variations in system behavior. This adaptability makes it particularly suitable for managing the unpredictable characteristics

of soft robots. There is some existing work on adaptive control for soft robots but much is still heavily model-based and requires an explicit formulation of a regressor [25], [26], [27], [28], which can be difficult to compute for higher degrees of freedom. Alternative approaches combining adaptive control with machine learning do not require analytical models, but require offline training [29], [30], [31] or manual data collection [32]. Continual learning [33] offers a method to mitigate some of these issues by learning online, but the authors report that retraining online takes up to one minute. This becomes cumbersome in contact-rich manipulation where dynamics can change quickly. An additional challenge with purely data-driven approaches is that data is often collected by commanding randomized inputs. On some platforms–ours included–this approach can easily damage the hardware. It is not yet clear how to gather sufficiently rich data while also guaranteeing safety.

In order to address the challenges discussed above, the contributions of this paper are as follows.

- We present our design for a large-scale robot platform named Baloo (see Fig. 1), designed with a hybrid rigid-soft methodology to combine the strength of rigid components with the adaptability of soft components.
- We present a lightweight neural-network based adaptive controller originally developed for rigid robots in [34] and show that it successfully compensates for nonlinearities and time-varying disturbances in real-time. We also release an open-source Python implementation of the controller[1].
- We analyze Baloo's hardware capabilities through various whole-body manipulation tasks using a variety of challenging objects. The accompanying video[2] and experimental data[3] are available online.

The remainder of the paper is structured as described next. Section II-A presents the mechanical design of the platform. Sections II-B and II-C provide details about the system integration and characterization. Section II-D derives the adaptive controller. Section III contains the results of the adaptive controller experiments as well as the whole-arm grasping trials. We finish with a discussion of the results and future work in Sections IV and V respectively.

II. METHODS

A. Design

In this section we present design details of each subsystem of Baloo (Fig. 1). We start with the compliant joints, then discuss how the joints are assembled to build the pneumatic arms and how the arms operate in the context of the entire torso assembly.

1) Compliant Joints: The interior, central structure, or spine of the compliant joints, highlighted in Fig. 2a, is composed of an alternating series of Polyethylene terephthalate glycol (PET-G) spacers (Fig. 3, #2) and open-cell polyethylene foam sheets (Fig. 3, #3) stacked around an aramid rope (Fig. 3, #11).

---

[1]https://github.com/byu-rad-lab/manipulator-adaptive-control

[2]https://youtu.be/eTUvBEVGKXY

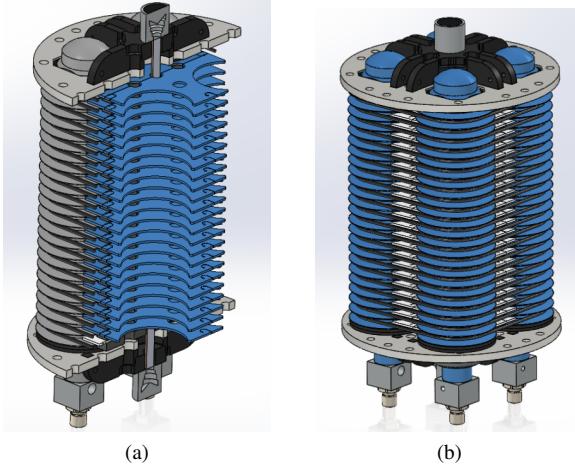[3]https://github.com/byu-rad-lab/baloo-data-analysis

Fig. 2. External and Internal views of the small soft pneumatic robot joints used in the arms. Fig. 2a shows a cross-sectional view of the joint with the constraining spacers highlighted in blue. Fig. 2b shows a fully-assembled small joint with the four blow-molded pneumatic actuation chambers highlighted in blue. Also visible is the rope and key mechanism that travels through the center of the joint and constrains the joint motion longitudinally.

In order to constrain the length of the joint, we rigidly attach an aluminum end fitting (Fig. 3, #7) to each end of the aramid rope. A steel ball bearing is embedded into the rope weave (Fig. 4a), inserted into the the end fitting (Fig. 4b), and epoxied together. We observed from experimental testing that addition of a steel ball increases the tensile strength of the spine assembly by about 50% over epoxy alone.

To actuate the joint, we use a set of blow-molded Polyethylene Terephthalate (PET) actuation chambers with an accordion-like structure that allows bending and elongation/compression (Fig. 3, #4 and highlighted in blue in Fig. 2b). Each of these chambers is assembled into the spine, with spacers along the spine aligning with grooves in the accordion structure. Once all four chambers are in place, they are outfitted with protective TPU gaskets (Fig. 3, #10), set into the holes of the bottom aluminum plate (Fig. 3, #1), and locked from underneath with a retaining ring 3D printed in PLA (Fig. 3, #8). The same procedure is repeated for the top plate, with the exception of the retaining ring, as the top of the plastic chamber does not have a groove to accept one.

The chamber/spine assembly is compressed to allow the rope end fittings to pass through the center of both aluminum end plates and lock in place with a key assembly. The removable key (Fig. 3, #6) is 3D printed in PLA with 100% infill and supported by the 3D printed structure shown in Fig. 3, #5. This structure spreads the load across the majority of the end plate. A section view of the resulting assembled joint after key insertion is shown in Fig. 2a. The result is an internal static equilibrium between the chambers which are in compression and the aramid rope held in tension by the key assembly.

Aluminum interface blocks (Fig. 3, #9) insert into the bottom of each pressure chamber and are sealed with an o-ring. The blocks allow access to the chamber for air flow via tube fittings (Fig. 3, #13) and for pressure sensing.

We designed three sizes of compliant joints to build an entire arm. Each of the three sizes shown in Fig. 5 uses a similar
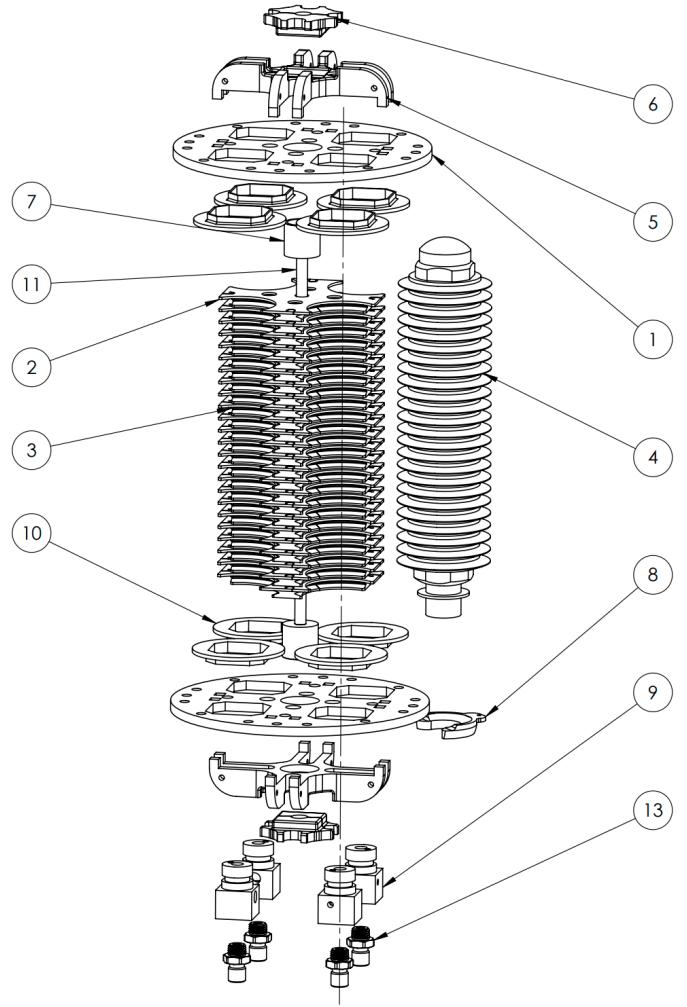

Fig. 3. Exploded assembly view of a single small joint. Three additional plastic actuation chambers (#4) are hidden to simplify the view.

Fig. 4. Steel ball embedded in aramid rope braid and inserted into end fitting.

design with a few small modifications. To provide a higher output torque, the medium joint is slightly wider than the small joint to increase the moment arm of the pressure chambers. In similar fashion, the large joint, which is responsible for lifting the entire weight of the structure, is even wider with two pressure chambers (as opposed to just one for the small and medium joints) acting along each actuation direction and controlled by a single valve.

Fig. 5. All three sizes of the passively compliant joints. Each size was selected based on torque requirements.
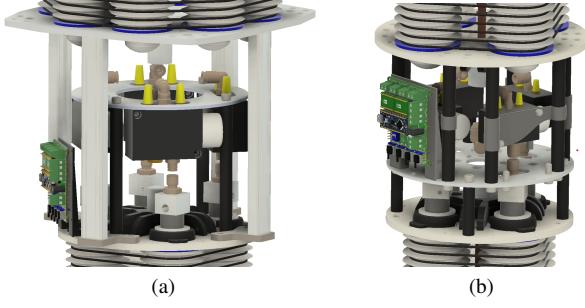

(a)   (b)

Fig. 6. Fig. 6a: Rigid link between large and medium joints, containing four valves which drive the chambers in the medium joint. Fig. 6b: Rigid link between medium and small joints, containing four valves driving the chambers in the small joint.

2) Pneumatically-Actuated Arms: Each of the compliant joints is actuated by individually filling or venting each of the pressure chambers. We use one Enfield LS 5/3 Proportional Directional valve to control a single chamber, for a total of 12 valves per arm. The valves are driven with the PneuDrive pressure control system [35]. Between the large and medium joints is a rigid link containing four LS-V15s valves and associated control electronics for the medium joint, shown in Fig. 6a. Between the medium and small joints is a similar rigid link, with four LS-V05s valves and electronics to control the small joint, shown in Fig. 6b. Four LS-V25s valves and electronics to control the large joint are located in the chest, shown in Fig. 7.

3) Base and Torso: Baloo is designed to have a workspace mostly in front of the robot, from the floor to approximately human height. Accordingly, the chest and arms are mounted on the Vention MO-LM-039-1530 Ball Screw Actuator, which is supported by a rigid structure of aluminum extrusion. The chest and arms are nominally at a height of 1.5 meters (as shown in Fig. 7) and can be lowered all the way to the floor.

The arms are mounted on the torso with ball bearings, which allow the arms to rotate outwards and reach in front of the robot. Each arm can be manually pinned into the desired configuration (i.e. a fixed angle relative to the chest) for manipulation tasks.

B. System Integration and Sensory Feedback

As previously mentioned, we use the modular pressure control system called PneuDrive [35] to control the four independent pressures for each joint. Each joint has an embedded

Fig. 7. Assembly of the entire bi-manual robot with base, chest, and both arms.

pressure control module running high-rate PD control on pressures. The modules are connected to a serial communication bus running through the middle of the joints into the chest, where an Odroid N2+ acts as a controller for a single arm. Air, power, and communication lines travel down the arm using the holes visible in the spacers in Fig. 2a. One of the Odroid computers also acts as controller for the elevator or linear actuator that moves the arms up and down.

The Odroid N2+ computers are both connected via ethernet, which allows the entire system to be controlled via ROS. The control structure of the entire system is shown in Fig. 8.

Sensory feedback for a single arm includes pressures in each of the pressure chambers $p \in \mathcal{R}^{12}$, joint positions $q \in \mathcal{R}^6$, and joint velocities $\dot{q} \in \mathcal{R}^6$, as shown in Fig. 8. The height of the chest h is another configuration variable.

We use HTC VIVE Motion Trackers to track the orientation of both end plates of each joint (see the black trackers in Fig. 1) and calculate joint angles using the constant-curvature, singularity-free kinematic parameterization from [37]. Future work will include embedding length sensors as shown in [37], [38] into the joints to eliminate the need for an external motion capture system and relax the constant-curvature assumption.

The pressures p are all reported via PneuDrive. It is important to note that the pressure feedback can be thought of as a built-in torque sensor. As the compliant chambers deform in response to an external load the pressures also change. By having access to the pressures at any given instant we have an estimate of the actuation torques. The pressures play a similar role as current sensing in traditional rigid robots. An added
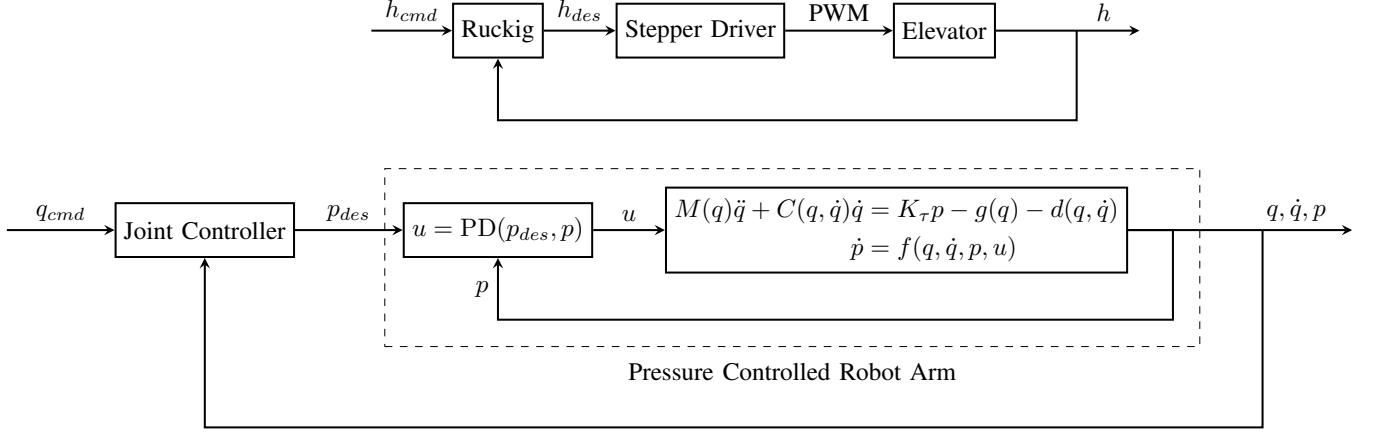
Fig. 8. Control Diagram of the Arms. We use the Ruckig Motion Planning [36] library to generate smooth elevator motions. The dynamics of an arm are given by the mass matrix M, the Coriolis matrix C, the pressure-to-torque mapping matrix K_τ, and the gravity torques g. d is a vector of unknown/poorly modeled disturbances (e.g. stiffness, damping, etc). The pressure dynamics are listed explicitly here as some function f of input and configuration variables.

benefit is that they do not require any extra hardware beyond what is needed to actuate the joint. Furthermore, the pressure-to-torque model is much more robust to inaccurate parameters than the current-to-torque model for traditional rigid robots.

Joint-space controllers provide pressure commands p_{des} given commanded joint angles q_{cmd}. To reduce undesirable mechanical vibrations for the elevator, we use the Ruckig [36] Motion Planning library to plan time-optimal jerk, acceleration, and velocity constrained trajectories h_{des} in real time, given a commanded height h_{cmd}.

C. System Identification

We performed two system identification tests to determine the dynamic characteristics of the compliant joints. The first test captures the free response of a single joint after an initial displacement. The result for a single small joint is shown in Fig. 9. This response is representative of the other joint sizes. We use the logarithmic decrement method [39] to estimate the damping ratio of the uncontrolled joint and found that it is approximately $\zeta = .0015$. The period of oscillation is approximately 0.125 seconds, which means that the damped natural frequency $\omega_d = 8$ Hz. For a second order system, the natural frequency is related to the damped natural frequency by $\omega_d = \omega_n \sqrt{1 - \zeta^2}$, which gives a natural frequency of approximately 8 Hz as well. ω_n and ζ together fully characterize a second order system. The highly underdamped nature of the joint is somewhat undesirable as it makes control difficult with pneumatics, which have a limited closed-loop control bandwidth. Future work will include improving the mechanical damping properties of the foam inside the joint to increase the damping ratio and keep parasitic torques low.

Since we parameterize each joint with two degrees of freedom [37], we assume that these dynamic parameters are the same for each joint angle, though in reality this is not true. Due to non-ideal material properties which arise from the blow molding manufacturing process, we qualitatively observed that the stiffness and damping are not uniform between the different plastic pressure chambers. The stiffness
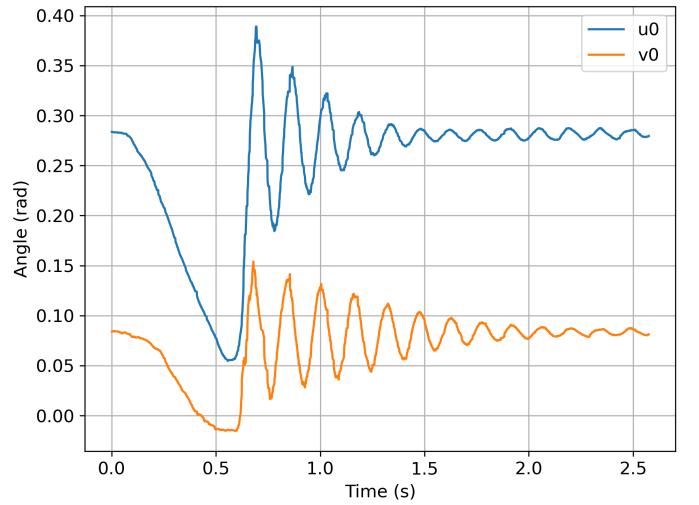

Fig. 9. Free response of a single small joint along each degree of freedom. The logarithmic decrement method counting times at successive peaks gives an estimated damping ratio of .0015 and a damped natural frequency of about 8 Hz.

and damping are also not linear with respect to q and $\dot{q}$ as is often assumed from Hooke's Law and viscous damping terms. These differences comprise part of the uncertainty that underscores the importance of the adaptive controller presented in the following sections.

The second test measures the mapping from pressure differential (i.e. torque) to joint angle and is shown in Fig. 10. We gathered data by commanding pressure differentials as a slow ramp and measuring the corresponding joint angle. The pressure differential is the difference in pressure between antagonistic pressure chambers. It is clear that there is a significant hysteresis loop present along each degree of freedom. We also observed that if the plastic chambers are deformed for long periods of time the position of the hysteresis loop shifts and introduces temporal uncertainty, causing fixed-gain control methods to fail over time. Again, this motivates the need for the adaptive controller presented in the following section.
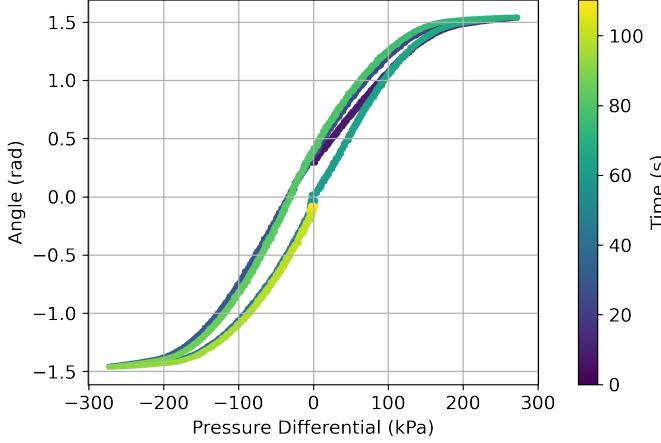
Fig. 10. Typical hysteresis loop of a single small joint over two cycles. The pressure differential is the difference in pressures between antagonistic pressure chambers and is the quantity which causes a bending torque.

D. Adaptive Controller Development

In this section we derive control and adaptation laws and analyze the stability of the resulting closed-loop system via Lyapunov analysis. We then share practical implementation details that we found to be important for achieving good performance.

In order to implement an adaptive controller suitable for Baloo, we adapt the controller originally presented in [40] to use Radial Basis Functions (RBFs) as a dynamics function approximator, similar to [34]. The authors of [34] formulate their adaptation mechanism to adapt RBF centers and widths in addition to the weights. We originally used this idea, but found that these additions made no impact on performance so we drop them from our formulation to minimize computational load.

An RBF Neural Network (RBFNN) attempts to approximate a nonlinear function f with some error ϵ as

$$f(x) = \Theta^T \Phi(x) + \epsilon(x) \quad (1)$$

where

$$x \in \mathcal{R}^N = \left[x_0, x_1, \ldots, x_{N-1} \right]^T$$

$$\Phi(x) \in \mathcal{R}^{P+1} = \left[\phi_0(x), \phi_1(x), \ldots, \phi_{P-1}(x), 1 \right]^T$$

$$\Theta^T \in \mathcal{R}^{n \times P+1} = \begin{bmatrix} \theta_{0,0} & \cdots & \theta_{0,P-1} & b_0 \\ \theta_{1,0} & \cdots & \theta_{1,P-1} & b_1 \\ \vdots & \ddots & \vdots & \vdots \\ \theta_{n-1,0} & \cdots & \theta_{n-1,P-1} & b_{n-1} \end{bmatrix} \quad (2)$$

$$f(x) \in \mathcal{R}^n = \left[f_0, f_1, \ldots, f_{n-1} \right]^T$$

with n outputs for each generalized coordinate in the dynamics, P RBF centers, and where $N = 4n$ inputs. Each scalar-valued RBF function ϕ_i is of the form

$$\phi_i(x) = \exp\left(-\frac{P}{d_{max}^2} ||x - m_i||^2 \right) \quad (3)$$

with a vector input x, a center vector $m_i \in \mathcal{R}^N$, and a width given by P/d_{max}^2. d_{max} is the maximum distance between any two RBF centers. The width is expressed in this way to heuristically maintain good coverage of the input space, with RBFs overlapping one another sufficiently.

We start with the robot dynamics as shown in Fig. 8, with $n = 6$ generalized coordinates:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} = K_\tau p - g(q) - d(q, \dot{q}) \quad (4)$$
$$\dot{p} = f(q, \dot{q}, p, u) \quad (5)$$

To simplify analysis, we assume that the closed-loop pressure dynamics $\dot{p}$ are much faster than the joint dynamics (i.e. $p_{des} = p$). This assumption is true if the joint trajectories are sufficiently slow. We measured the time constant of the pressure dynamics to be on the order of .1 seconds, so the commanded joint trajectories should have a time constant of at least 1 second.

Similar to [40], we start with a desired joint trajectory q_{des}, $\dot{q}_{des}$, and $\ddot{q}_{des}$, with the tracking error defined as $\tilde{q} = q - q_d$ and a virtual reference trajectory as

$$q_r = q_d - \Lambda \int_0^t \tilde{q}\, dt$$
$$\dot{q}_r = \dot{q}_d - \Lambda \tilde{q} \quad (6)$$
$$\ddot{q}_r = \ddot{q}_d - \Lambda \dot{\tilde{q}}.$$

We define an asymptotically stable surface as

$$s = \dot{\tilde{q}} + \Lambda \tilde{q} = \dot{q} - \dot{q}_r \quad (7)$$

whose value represents the velocity tracking error between the actual joint velocities and the virtual reference velocities. Note that if $s \to 0$ as $t \to \infty$ (i.e. trajectories converge to the sliding surface), then the tracking error $\tilde{q} \to 0$ as $t \to \infty$ as well.

Consider a Lyapunov Function candidate adapted from [34] of the form

$$V(t) = \frac{1}{2}\left[s^T M(q)s + \text{tr}(\tilde{\Theta}^T \Gamma^{-1}\tilde{\Theta}) \right] \quad (8)$$

where $\tilde{\Theta} = \hat{\Theta} - \Theta^*$ is the weight error between the estimated weights and the optimal (but unknown) weights of the RBFNN. $\Gamma > 0 \in \mathcal{R}^{(P+1) \times (P+1)}$ is a diagonal matrix of learning rates.

Differentiating V from Eqn. 8 with respect to time results in the following:

$$\dot{V}(t) = \dot{q}_r^T (K_\tau p - M\ddot{q}_r - C\dot{q}_r - g - d)$$
$$+ \text{tr}\left(\tilde{\Theta}^T \Gamma^{-1}\dot{\hat{\Theta}} \right). \quad (9)$$

For a full derivation of this result, please see the appendix. We approximate the robot dynamics using Equation 1 as

$$f(x) = M\ddot{q}_r + C\dot{q}_r + g(q) + d(q, \dot{q}) = \Theta^{*T}\Phi(x) + \epsilon(x) \quad (10)$$

where $x = [q, \dot{q}, \dot{q}_r, \ddot{q}_r]$. If we choose a control law shown in Fig. 11 as

$$K_\tau p_{des} = \hat{\Theta}^T \Phi(x) - K_D s \qquad (11)$$

and substitute (10) and (11) into (9) then we get

$$\dot{V}(t) = -s^T K_D s - s^T \epsilon(x)$$
$$+ \text{tr}\left(\tilde{\Theta}^T [\Phi(x)s^T + \Gamma^{-1}\dot{\hat{\Theta}}]\right). \qquad (12)$$

Again, please see the appendix for the intermediate steps. Choosing an adaptation law of

$$\dot{\hat{\Theta}} = -\Gamma \Phi(x) s^T \qquad (13)$$

eliminates the last term of $\dot{V}$. If the RBFNN is designed to be expressive enough, we can assume that the approximation error is bounded by some small constant $||\epsilon(x)|| < E$. An upper bound on $\dot{V}$ is then given by

$$\dot{V}(t) = -s^T K_D s - s^T \epsilon(x)$$
$$\leq -\lambda_{min}(K_D)||s||^2 + ||s||E \qquad (14)$$

where $\lambda_{min}(K_D)$ is the smallest eigenvalue of K_D. Thus $\dot{V} \leq 0$ when

$$||s|| \geq \frac{E}{\lambda_{min}(K_D)}. \qquad (15)$$

Equation 15 implies that trajectories of s are asymptotically stable to an arbitrarily small region around the origin. Inside this small region stability is not strictly guaranteed, but trajectories will remain globally ultimately bounded. In practice, choosing too large of a value for $\lambda_{min}(K_D)$ such that $\frac{E}{\lambda_{min}(K_D)} \to 0$ can destabilize the system. However, during experiments we were able to drive trajectories of s to within the noise threshold around the origin with relatively small values of $\lambda_{min}(K_D)$ (see Table I and Fig. 13).

From a theoretical perspective, Λ does not directly affect the stability of the system. Equation 7 shows that $||s||$ grows as Λ increases, but trajectories of s will still converge to the small region around the origin defined by $\frac{E}{\lambda_{min}(K_D)}$. Since s appears in both Equation 11 and 13, a larger Λ leads to larger control actions and adaptation rates and smaller tracking error. In practice Λ is a type of proportional gain which, if too high, can cause oscillations, jitter, or instability.

Γ also does not affect the stability of the system directly. A larger Γ results in increased adaptation speed and therefore lower tracking error. Intuitively, Γ is a type of integral term, which if it is too high, can cause overshoot, windup, or instability.

1) Controller Implementation Details: The RBFNN adaptive controller is implemented in Python with an average computation time per loop of 87 μs and is available for open-source use[1]. As noted above, we want the RBFNN to be as expressive as possible while still minimizing the computational load. The user specifies the P radial basis functions with their upper and lower bounds. This information is used to produce a set of Gaussian center vectors (m_i in Equation 3) that are generated with a Latin Hypercube Sampling (LHS) method. LHS helps the RBFNN be more expressive than uniform random sampling, especially in higher dimensions (e.g. $\mathcal{R}^N$ in our case), because it guarantees a degree of spread between the distributions, thereby avoiding gaps caused by high density clusters. d_{max} is then calculated by brute force (i.e. looping through all center vectors and finding the maximum distance between any two points). Calculating d_{max} in this way has an unfavorable time complexity of $\mathcal{O}(P^2)$ but this calculation needs to happen only once on startup. Additionally, we observed that above the range of $10 \leq P \leq 50$ there is no distinguishable performance improvement, so the startup computational cost is relatively minor. The weights $\theta_{i,j}$ and biases b_i are all initialized to be zero.

Our implementation of the RBFNN also includes bias terms (see Equation 1), unlike many other implementations. Although bias terms are not strictly necessary—as the RBFNN can approximate a function equally well without them if the weights θ are larger—we found that adding bias terms b allows the weights to be much smaller after reaching their optimal values. Intuitively this can be thought of as decoupling the function shape approximation via RBFNN weights from its location approximation via the bias terms. Physically, this means that any generalized torques that are not zero-centered (e.g. gravity torques or spring force offsets) are more explicitly estimated with the bias terms, as opposed to just implicitly through RBFNN weights. Consequently, for the same learning rate Γ, the weights converge faster with bias terms than without them. In other words, using bias terms enables us to achieve similar control performance with lower gains, compared to high gain control which introduces many well-known stability issues and sensitivities.

Another unique feature of our implementation is the feed-forward stiffness compensation shown in Fig. 11. The stiffness compensation term uses the desired joint angle signal q_{des} as opposed to the commanded joint angle q_{cmd}, as it provides a smoother input signal. Even with an extremely rough estimate of K, the addition of this term improves dynamic performance, decreases the load on the adaptation mechanism, and allows the gains to be lower and preserve the natural passive compliance during interactions [41]. This addition changes the control law to be

$$K_\tau p_{des} = \hat{\Theta}^T \Phi(x) - K_D s + K q_d. \qquad (16)$$

This addition does affect the stability of the closed-loop system because it adds another term to Equation 14:

$$\dot{V}(t) = -s^T K_D s - s^T \epsilon(x) + s^T K q_d \qquad (17)$$

Since the stiffness matrix $K > 0$, this new term can either enhance or diminish the stability of the closed loop system, depending on the signs of s and q_{des}. In the worst case, the stiffness feedfoward term can add a positive value to $\dot{V}$, which means that the new upper bound for $\dot{V}$ is

$$\dot{V}(t) \leq -\lambda_{min}(K_D)||s||^2 + ||s||E + \lambda_{max}(K)||s|| ||q_d||. \qquad (18)$$
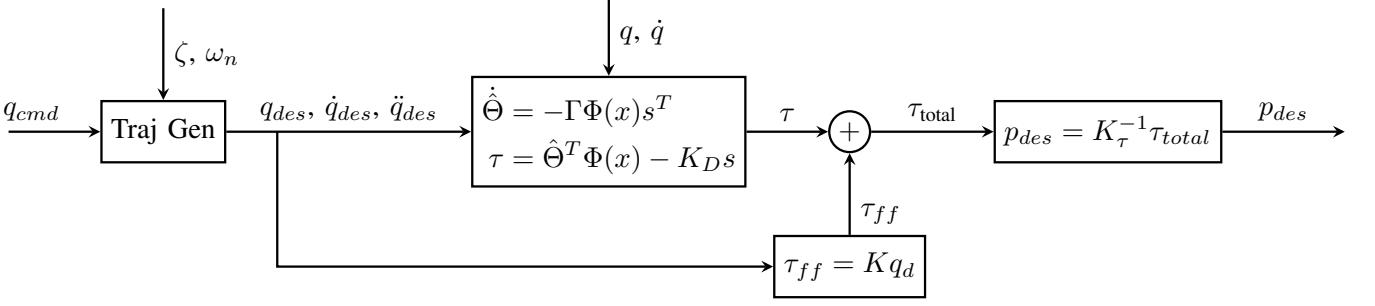
Thus $\dot{V} < 0$ if

Fig. 11. Adaptive Controller

$$||s|| \geq \frac{E + \lambda_{max}(K)||q_d||}{\lambda_{min}(K_D)}. \qquad (19)$$

Equation 19 highlights the design tradeoffs for the feedforward gain matrix K. A large K will improve transient response but could potentially destabilize the system.

III. HARDWARE EXPERIMENTS

A. Controller Performance

The control parameters we used for the following hardware experiments are recorded in Table I. The raw data, analysis, and code for all hardware experiments are publicly available on GitHub[3].

TABLE I
CONTROL PARAMETERS FOR EXPERIMENTAL TRIALS

Parameter	Value
Control Rate	500 Hz
Max Pressure	300 kPa
P	10
$\tau = \frac{1}{\omega_n \zeta}$	0.75 s
ζ	1.0
Λ	$diag(12.0, 12.0, 12.0, 12.0, 25.0, 25.0)$
Γ	$15.0 * I_P$
K_D	$2.5 * I_n$
K	$35 * I_n$
K_τ	$\begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}$

1) Step Response: To experimentally validate the performance of the adaptive controller, we sent a series of random joint angle commands q_{cmd} to the left arm controller of Baloo. A new command was sent every 10 seconds to allow us to examine the settling time and steady-state performance of the controller. The results of this experiment are shown in Fig. 12 and 13.

Fig. 12 shows the input q_{cmd} in blue, the output of the trajectory generator q_{des} in dotted green, and the actual joint trajectory q in orange. Note that the large amplitude oscillations at the beginning of the trajectory are, in part, due to the large pressure command sent at the beginning of each trial, when the arm pressure chambers fill from 0 kPa to a nominal 150 kPa. This large step induces high frequency oscillations similar to what is shown in Fig. 9. However, the adaptive
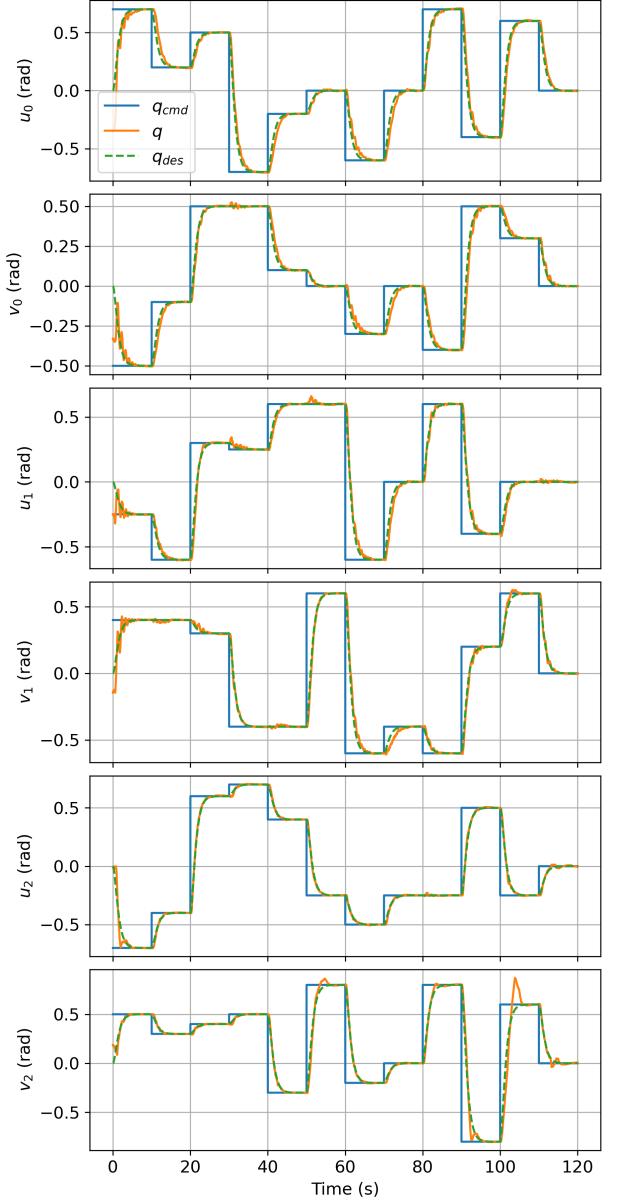

Fig. 12. Control performance on the entire arm. Average tracking errors for this trajectory are shown in Fig. 13.
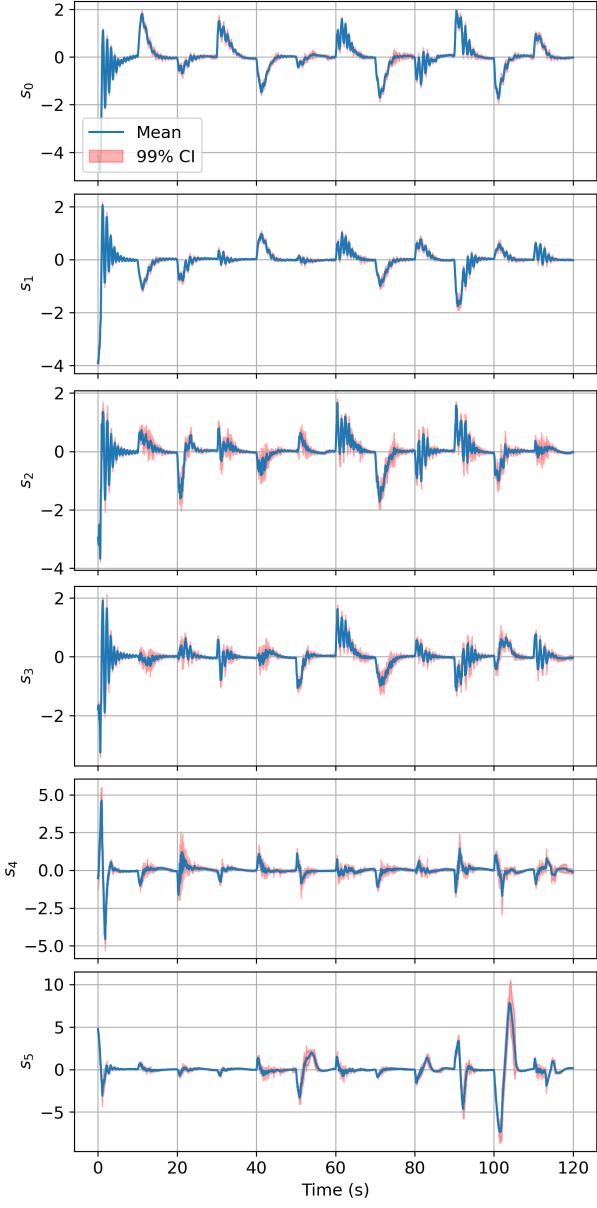
Fig. 13. Tracking error s signals for each degree of freedom for the trajectory shown in Fig. 12. Red shaded error is the 99% confidence interval over five trials.

Fig. 14. We added a 10 lb (4.5 kg) weight to a distal link of the left arm and restarted the same controller from scratch. No information about the weight was given to the controller.

controller that we developed is able to damp these out within a few seconds.

We repeated this trial five times and report the mean value of s for each degree of freedom in Fig. 13. Recall that s can be thought of as a weighted sum of the position and velocity tracking errors (see Equation 7). The red shaded region represents a 99% confidence interval on the mean of s for all five trials. This gives a sense of the repeatability and robustness of the control method, which is particularly important for robots with highly compliant members because they deform over time. We observed that the joints plastically deform during the data collection process for this experiment. When depressurizing all of the chambers, the joints did not return to the same configurations as before the data collection process.

2) Robustness to External Disturbances:

We also commanded the same step input trajectory with a 4.5 kg mass attached to a distal link of the arm, effectively doubling the link mass, as shown in Fig. 14. We did not provide any information about the added weight to the adaptive controller, which was initialized randomly as explained previously. Midway through the trajectory, at about $t = 55$ seconds, we released the velcro straps quickly and observed how the controller reacted to the significant and sudden change in the arm dynamics. The results of this test are shown in Fig. 15, where the red bar highlights the moment when we release the weight. A video of this experiment is available online[2].

B. Whole Arm Grasping

To evaluate the effectiveness of the hybrid compliant-rigid robot design in offloading some of the computational complexity of manipulation to the soft robotic hardware (i.e. mechanical intelligence), we perform whole-arm grasping experiments with several large, heavy, and unwieldy objects, listed in Table II and pictured in Fig. 16. We chose these objects to span a variety of sizes, textures, weights, and shapes. We use only an approximation of the geometric centroid to know where to start the elevator. No other object-specific information is used in any part of the trials.

As noted in Table II, the box has a nonuniform mass distribution because it is filled with weights that are free to move around. The mass is initially concentrated at the bottom center of the box, but the weights move around as the box tilts during manipulation.

To start the experiment we command the elevator to a height which leaves the torso roughly centered on each object. We place the object in the graspable workspace in front of the torso and between the arms. We varied the both the placement
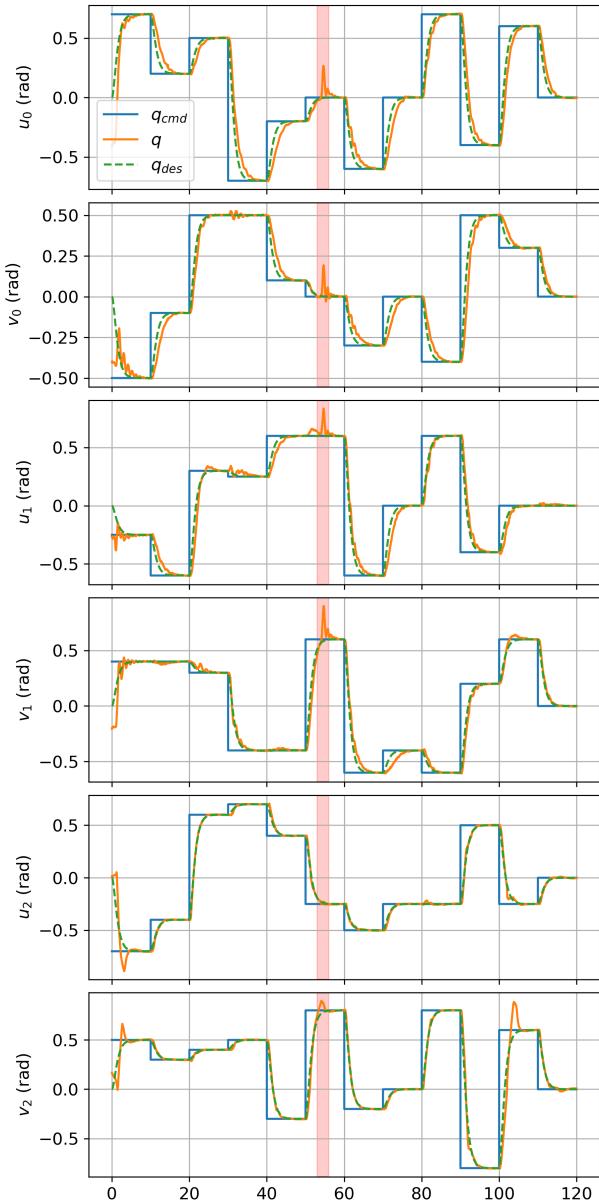
Item	BBox (m)	Mass (kg)	Challenges
Car Tire	0.8 x 0.8 x 0.2	21.9	Large, Heavy
Office Chair	0.7 x 0.7 x 1.0	11.9	Non-convex, Hinged
Ladder	1.2 x 0.6 x 0.2	15.6	Slippery, Heavy
Duffel Bag	0.4 x 0.4 x 0.8	6.75	Deformable
Kayak	1.9 x 0.7 x 0.3	8.5	Large, Unwieldy
Large Box	0.6 x 0.5 x 0.5	7.3	Nonuniform mass dist.

Fig. 15. 10 lb weight (4.5 kg) Dropped weight on red shaded area

Fig. 16. Objects used to perform whole arm grasping trials

and orientation of each object randomly, as can be seen in the video[2]. Then similar to the experimental procedures in [4] and [42], we slowly increment the pressure commands to move the joints into a grasping pose until a joint reaches the maximum allowed pressure of 300 kPa. This does not mean that the actuators are physically saturated. The failure pressure for these actuation chambers is approximately 615 kPa. Therefore, to maintain a safety factor of 1.5, we enforce a hardware-limited maximum pressure of 410 kPa, while only commanding pressures up to 300 kPa. This highlights the capability of compliant members to passively conform to a variety of geometries and weight distributions. Once the joints have all settled to their maximum allowable torques, the elevator is commanded to raise the object. We repeat this procedure five times for each object to test how well the hardware conforms to differences in the initial object pose. All of the objects were successfully picked up in all five of their trials. A single trial of each object is shown in Fig. 17. Recordings of all trials for all of the objects are also available online[2]. While we could have used the adaptive controller developed in Section II-D here, the purpose of this experiment is to observe the compliant behavior of the hardware during manipulation, not to control to a specific joint configuration. Thus we use only pressure commands (i.e. torque commands) and allow the arms to passively deflect.

IV. DISCUSSION

A. Adaptive Controller Performance

As can be seen in Fig. 11 and 14, the adaptive controller tracks the desired trajectories well, with and without external disturbances. There are occasional dynamic coupling effects between joints which appear as small disturbances to the adaptive controller (e.g. u_1 at $t = 30, 50$). As is expected, these effects occur most frequently on the large and medium joints which are carrying heavier loads with larger inertial parameters. As a result, the control gains for the last joint could be made larger to improve tracking performance and disturbance rejection without causing instability. Fig. 11 does indeed show excellent tracking on the last joint (u_2 and v_2). The exception to this are the large spikes most visible on v_2 at $t = 55, 95, 105$ seconds. We observed this to occur when, in certain configurations, the joint behaves like a bi-stable mechanism where the joint 'pops' into a nearby equilibrium configuration before being corrected by the adaptive controller. The behavior was very repeatable for all five trials, as shown with s_5 on Fig. 13. While we believe this is mechanical in
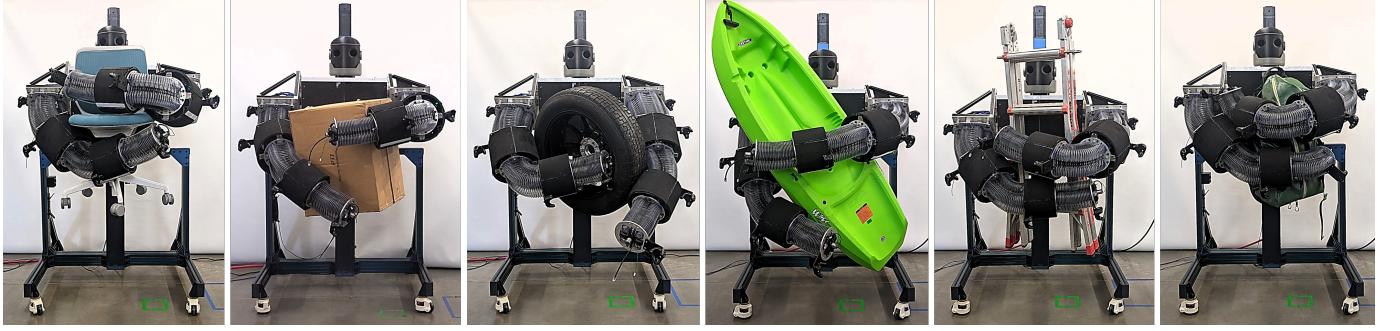
Fig. 17. All six objects successfully picked up using the same pressure trajectory. The final grasp pose of the arms vary significantly, which demonstrates the adaptability of passive compliance.

nature the specific cause is unknown and warrants further investigation.

On close inspection of Fig. 13, it is clear that there are high-frequency oscillations present during the transient portion of each step command. The controller is clearly robust to these oscillations and eliminates them quickly to converge to zero, but these oscillations are still undesirable. The overall effect is a generally shaky movement, which can be observed with a careful examination of the video[2]. This is a challenge of using purely reactive control with such a highly underdamped system. The oscillations could be eliminated by limiting the arms to move quasi-statically (e.g. $\tau >> 1$). They could also be eliminated by implementing some sort of predictive control mechanism to anticipate and compensate for these dynamics ahead of time, instead of simply reacting to them [28]. This, however, requires a model, which we do not use in this work. Lastly, this effect could be suppressed in hardware by choosing joint materials to increase the low damping ratio of the current joints (see Fig. 9). This capability is a unique feature of a hybrid rigid-soft robot design: we can more easily tune the hardware properties to improve control performance.

B. Whole-Body Manipulation

During the whole-body manipulation trials, recall that we placed each object in the reachable work space in front of Baloo but did not put any constraints on the initial pose of the object–only its approximate placement. The accompanying video[2] shows the variability in the initial placement of each object. This resulted in significant changes in the final grasping pose of the arms as shown in Fig. 17. The exact same pressure trajectory resulted in each of the different grasp poses shown, with a 100% success rate over all 30 trials. To the best of our knowledge, no other robotic manipulator–rigid or soft– has successfully manipulated such a wide variety of objects using only open-loop trajectories. This highlights the inherent adaptability of the hardware platform to conform to variations in size, geometry, and mass distribution while performing whole-body manipulation.

The ladder was particularly challenging because of its low surface friction. It was never dropped, but slipped occasionally as the object was lifted. The slippage stopped as the arms moved into contact with additional areas along the ladder in response to the changing pose of the object. Again, this highlights the benefit of a hybrid rigid-soft design as the soft elements act as high-bandwidth, no-delay slip controllers without any extra slip detection sensors.

V. CONCLUSION AND FUTURE WORK

This paper presents our prototype design of a bi-manual soft robot, built to explore the hybrid design philosophy of combining highly compliant structures with rigid ones. Since incorporating highly compliant materials into the structure of a robot introduces some unique challenges in regards to modeling and control, we develop a neural-network based adaptive controller and demonstrate with real-world experiments that it is able to compensate for the aforementioned challenges. We also showcase some of the unique capabilities of a hybrid-style robot design by offloading a significant portion of the control complexity to the hardware and successfully performing complicated and contact-rich manipulation with large and unwieldy objects.

Future work towards improving the performance of this prototype can include an in-depth investigation of specific material properties (e.g. stiffness and damping) to embed into the compliant joints to optimize its dynamic behavior. Refining the manufacturing process can also help to reduce the uncertainty we observed in material properties as well as kinematics.

We also depend on the HTC Vive motion tracking system for configuration estimation. This is not ideal, especially for whole-arm manipulation tasks, as occlusions and accidental contact with trackers is likely. For this work, we were careful to choose configurations that minimize this issue, but incorporating an on-board configuration sensing method would be valuable.

Adding a predictive control scheme can allow the arms to move more dynamically, faster than what the adaptive controller presented here is able to stabilize. While it does compensate for uncertainty well without any dynamic model, it is still purely reactive and inherently limited in dynamic performance for potential tasks like throwing or hammering. In some sense, the control method limits the capabilities of the hardware, which is capable of very dynamic movements. The combination of adaptive and predictive control is extremely promising for this type of robot, though the role of a model

needs to be investigated thoroughly. Work is currently underway to enable more dynamic manipulation tasks like throwing [11].

Additionally, future research on outfitting Baloo with tactile sensing skin to perform closed-loop control during whole-body manipulation tasks should enable more advanced and complex bi-manual and whole-body manipulation. We expect that using fabric-based tactile sensing in a closed loop [43], as opposed to the open-loop trajectories used in this work, will be essential to enabling improved manipulation capabilities.

We also anticipate that the addition of onboard cameras could be used for both proprioception (e.g. observing passive arm deformations) as well as exteroception (e.g. object pose estimation or grasp planning).

Implementing control loops or learned policies using these additional sensing modalities could open the door to a wide variety of open-world manipulation capabilities that have so far been impossible.

Appendix
Full Lyapunov Derivation

We start with the Lyapunov function candidate from Equation 8:

$$V(t) = \frac{1}{2}\left[s^T M(q)s + \mathrm{tr}\left(\tilde{\Theta}^T \Gamma^{-1} \tilde{\Theta} \right) \right]$$

The time derivative of V is derived as follows:

$$\dot{V} = \frac{1}{2}\left[\dot{s}^T M s + s^T \dot{M} s + s^T M \dot{s} \right]$$
$$+ \frac{1}{2}\mathrm{tr}\left(\dot{\tilde{\Theta}}^T \Gamma^{-1} \tilde{\Theta} + \tilde{\Theta}^T \Gamma^{-1} \dot{\tilde{\Theta}} \right)$$
$$= \frac{1}{2}\left[2s^T M \dot{s} + s^T \dot{M} s \right]$$
$$+ \frac{1}{2}\mathrm{tr}\left(\tilde{\Theta}^T \Gamma^{-1} \dot{\tilde{\Theta}} + \tilde{\Theta}^T \Gamma^{-1} \dot{\tilde{\Theta}} \right)$$
$$= s^T M \dot{s} + \frac{1}{2} s^T \dot{M} s + \mathrm{tr}\left(\tilde{\Theta}^T \Gamma^{-1} \dot{\tilde{\Theta}} \right)$$

Now substitute in the expression for $s = \dot{\tilde{q}}_r$ and $\dot{s} = \ddot{\tilde{q}}_r = \ddot{q} - \ddot{q}_r$:

$$\dot{V} = \dot{\tilde{q}}_r^T \left(M\ddot{q} - M\ddot{q}_r \right) + \frac{1}{2}\dot{\tilde{q}}_r^T \dot{M} \dot{\tilde{q}}_r + \mathrm{tr}\left(\tilde{\Theta}^T \Gamma^{-1} \dot{\tilde{\Theta}} \right)$$
$$= \dot{\tilde{q}}_r^T \left([\tau - C\dot{q} - g - d] - M\ddot{q}_r \right) + \frac{1}{2}\dot{\tilde{q}}_r^T \dot{M} \dot{\tilde{q}}_r$$
$$+ \mathrm{tr}\left(\tilde{\Theta}^T \Gamma^{-1} \dot{\tilde{\Theta}} \right)$$
$$= \dot{\tilde{q}}_r^T \left(\tau - C\dot{q} - g - d - M\ddot{q}_r \right) + \dot{\tilde{q}}_r^T \left(\frac{1}{2}\dot{M} - C + C \right) \dot{\tilde{q}}_r$$
$$+ \mathrm{tr}\left(\tilde{\Theta}^T \Gamma^{-1} \dot{\tilde{\Theta}} \right)$$
$$= \dot{\tilde{q}}_r^T \left(\tau - C\dot{q} - g - d - M\ddot{q}_r + C\dot{\tilde{q}}_r \right) + \mathrm{tr}\left(\tilde{\Theta}^T \Gamma^{-1} \dot{\tilde{\Theta}} \right)$$
$$= \dot{\tilde{q}}_r^T \left(\tau - C\dot{q} - g - d - M\ddot{q}_r + C\dot{q} - C\dot{q}_r \right) + \mathrm{tr}\left(\tilde{\Theta}^T \Gamma^{-1} \dot{\tilde{\Theta}} \right)$$
$$= \dot{\tilde{q}}_r^T \left(\tau - M\ddot{q}_r - C\dot{q}_r - g - d \right) + \mathrm{tr}\left(\tilde{\Theta}^T \Gamma^{-1} \dot{\tilde{\Theta}} \right)$$
$$= \dot{\tilde{q}}_r^T \left(K_\tau p - M\ddot{q}_r - C\dot{q}_r - g - d \right) + \mathrm{tr}\left(\tilde{\Theta}^T \Gamma^{-1} \dot{\tilde{\Theta}} \right)$$

By approximating dynamics with Equation 10 and choosing control law as in Equation 11 we get

$$\dot{V} = \dot{\tilde{q}}_r^T \left(\hat{\Theta}^T \Phi(x) - K_D s - \Theta^{*T}\Phi(x) - \epsilon(x) \right)$$
$$+ \mathrm{tr}\left(\tilde{\Theta}^T \Gamma^{-1} \dot{\tilde{\Theta}} \right)$$
$$= \dot{\tilde{q}}_r^T \left(\tilde{\Theta}^T \Phi(x) - K_D s - \epsilon(x) \right) + \mathrm{tr}\left(\tilde{\Theta}^T \Gamma^{-1} \dot{\tilde{\Theta}} \right)$$
$$= s^T \left(\tilde{\Theta}^T \Phi(x) - K_D s - \epsilon(x) \right) + \mathrm{tr}\left(\tilde{\Theta}^T \Gamma^{-1} \dot{\tilde{\Theta}} \right)$$
$$= -s^T K_D s - s^T \epsilon(x) + s^T \tilde{\Theta}^T \Phi(x) + \mathrm{tr}\left(\tilde{\Theta}^T \Gamma^{-1} \dot{\tilde{\Theta}} \right)$$
$$= -s^T K_D s - s^T \epsilon(x) + \mathrm{tr}\left(s^T \tilde{\Theta}^T \Phi(x) \right) + \mathrm{tr}\left(\tilde{\Theta}^T \Gamma^{-1} \dot{\tilde{\Theta}} \right)$$
$$= -s^T K_D s - s^T \epsilon(x) + \mathrm{tr}\left(\tilde{\Theta}^T \Phi(x)s^T \right) + \mathrm{tr}\left(\tilde{\Theta}^T \Gamma^{-1} \dot{\tilde{\Theta}} \right)$$
$$= -s^T K_D s - s^T \epsilon(x) + \mathrm{tr}\left(\tilde{\Theta}^T \Phi(x)s^T + \tilde{\Theta}^T \Gamma^{-1} \dot{\tilde{\Theta}} \right)$$
$$= -s^T K_D s - s^T \epsilon(x) + \mathrm{tr}\left(\tilde{\Theta}^T [\Phi(x)s^T + \Gamma^{-1} \dot{\tilde{\Theta}}] \right)$$

where we use the scalar property $\mathrm{tr}(a) = a$ and the cyclic property $\mathrm{tr}(ABC) = \mathrm{tr}(BCA)$.

References

[1] N. Hogan, "Impedance control: An approach to manipulation," in *1984 American Control Conference*, 1984, pp. 304–313.

[2] M. Schumacher, J. Wojtusch, P. Beckerle, and O. Von Stryk, "An introductory review of active compliant control," *Robotics and Autonomous Systems*, vol. 119, p. 185–200, Sep. 2019.

[3] K. Salisbury, W. Townsend, B. Ebrman, and D. DiPietro, "Preliminary design of a whole-arm manipulation system (wams)," in *1988 IEEE International Conference on Robotics and Automation Proceedings*, Apr. 1988, p. 254–260 vol.1.

[4] A. Goncalves, N. Kuppuswamy, A. Beaulieu, A. Uttamchandani, K. M. Tsui, and A. Alspach, "Punyo-1: Soft tactile-sensing upper-body robot for large object manipulation and physical human interaction," in *2022 IEEE 5th International Conference on Soft Robotics (RoboSoft)*. Edinburgh, United Kingdom: IEEE, Apr. 2022, p. 844–851. [Online]. Available: https://ieeexplore.ieee.org/document/9762117/

[5] M. Zhang and J. Barreiros, "Plan-guided reinforcement learning for whole-body manipulation."

[6] B. Zhang, Y. Xie, J. Zhou, K. Wang, and Z. Zhang, "State-of-the-art robotic grippers, grasping and control strategies, as well as their applications in agricultural robots: A review," *Computers and Electronics in Agriculture*, vol. 177, p. 105694, Oct. 2020.

[7] N. Dehio, Y. Wang, and A. Kheddar, "Dual-arm box grabbing with impact-aware mpc utilizing soft deformable end-effector pads," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, p. 5647–5654, Apr. 2022.

[8] J. Hughes, U. Culha, F. Giardina, F. Guenther, A. Rosendo, and F. Iida, "Soft manipulators and grippers: A review," *Frontiers in Robotics and AI*, vol. 3, Nov. 2016. [Online]. Available: https://www.frontiersin.org/articles/10.3389/frobt.2016.00069

[9] D. S. Shah, J. W. Booth, R. L. Baines, K. Wang, M. Vespignani, K. Bekris, and R. Kramer-Bottiglio, "Tensegrity robotics," *Soft Robotics*, vol. 9, no. 4, p. 639–656, Aug. 2022.

[10] A. Bhatt*, A. Sieler*, S. Puhlmann, and O. Brock, "Surprisingly robust in-hand manipulation: An empirical study," in *Robotics: Science and Systems XVII*. Robotics: Science and Systems Foundation, Jul. 2021. [Online]. Available: http://www.roboticsproceedings.org/rss17/p089.pdf

[11] S. N. T. Zwane, D. G. Cheney, C. C. Johnson, Y. Luo, Y. Bekiroglu, M. Killpack, and M. P. Deisenroth, "Learning dynamic tasks on a large-scale soft robot in a handful of trials," in *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, 2024.

[12] C. Della Santina, C. Duriez, and D. Rus, "Model-based control of soft robots: A survey of the state of the art and open challenges," *IEEE Control Systems Magazine*, vol. 43, no. 3, p. 30–65, Jun. 2023.

[13] H. Hauser, T. Nanayakkara, and F. Forni, "Leveraging morphological computation for controlling soft robots: Learning from nature to control soft robots," *IEEE Control Systems Magazine*, vol. 43, no. 3, p. 114–129, Jun. 2023.

[14] S. Li, S. A. Awale, K. E. Bacher, T. J. Buchner, C. Della Santina, R. J. Wood, and D. Rus, "Scaling up soft robotics: A meter-scale, modular, and reconfigurable soft robotic system," *Soft Robotics*, vol. 9, no. 2, p. 324–336, Apr. 2022.

[15] N. S. Usevitch, Z. M. Hammond, M. Schwager, A. M. Okamura, E. W. Hawkes, and S. Follmer, "An untethered isoperimetric soft robot," *Science Robotics*, vol. 5, no. 40, p. eaaz0492, Mar. 2020.

[16] G. Ambaye, E. Boldsaikhan, and K. Krishnan, "Soft robot design, manufacturing, and operation challenges: A review," *Journal of Manufacturing and Materials Processing*, vol. 8, no. 22, p. 79, Apr. 2024.

[17] P. Hyatt, D. Wingate, and M. D. Killpack, "Model-based control of soft actuators using learned non-linear discrete-time models," *Frontiers in Robotics and AI*, vol. 6, 2019. [Online]. Available: https://www.frontiersin.org/articles/10.3389/frobt.2019.00022/full

[18] Y. Huang, M. Hofer, and R. D'Andrea, "Offset-free model predictive control: A ball catching application with a spherical soft robotic arm," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2021, p. 563–570. [Online]. Available: https://ieeexplore.ieee.org/document/9636608/?arnumber=9636608

[19] Z. Chen, F. Renda, A. L. Gall, L. Mocellin, M. Bernabei, T. Dangel, G. Ciuti, M. Cianchetti, and C. Stefanini, "Data-driven methods applied to soft robot modeling and control: A review," *IEEE Transactions on Automation Science and Engineering*, p. 1–16, 2024.

[20] C. Alessi, D. Bianchi, G. Stano, M. Cianchetti, and E. Falotico, "Pushing with soft robotic arms via deep reinforcement learning," *Advanced Intelligent Systems*, vol. n/a, no. n/a, p. 2300899.

[21] T. G. Thuruthel, E. Falotico, F. Renda, and C. Laschi, "Model-based reinforcement learning for closed-loop dynamic control of soft robotic manipulators," *IEEE Transactions on Robotics*, vol. 35, no. 1, p. 124–134, Feb. 2019.

[22] C. C. Johnson, T. Quackenbush, T. Sorensen, D. Wingate, and M. D. Killpack, "Using first principles for deep learning and model-based control of soft robots," *Frontiers in Robotics and AI*, vol. 8, p. 654398, May 2021.

[23] D. Kim, S.-H. Kim, T. Kim, B. B. Kang, M. Lee, W. Park, S. Ku, D. Kim, J. Kwon, H. Lee, J. Bae, Y.-L. Park, K.-J. Cho, and S. Jo, "Review of machine learning methods in soft robotics," *PLOS ONE*, vol. 16, no. 2, p. e0246102, Feb. 2021.

[24] D. Bruder, X. Fu, R. B. Gillespie, C. D. Remy, and R. Vasudevan, "Data-driven control of soft robots using koopman operator theory," *IEEE Transactions on Robotics*, vol. 37, no. 3, p. 948–961, Jun. 2021.

[25] M. Trumic, C. Della Santina, K. Jovanovic, and A. Fagiolini, "Adaptive control of soft robots based on an enhanced 3d augmented rigid robot matching," in *2021 American Control Conference (ACC)*. New Orleans, LA, USA: IEEE, May 2021, p. 4991–4996. [Online]. Available: https://ieeexplore.ieee.org/document/9482817/

[26] M. Trumić, K. Jovanović, and A. Fagiolini, "Decoupled nonlinear adaptive control of position and stiffness for pneumatic soft robots," *The International Journal of Robotics Research*, vol. 40, no. 1, p. 277–295, Jan. 2021.

[27] *Adaptive Control of Large-Scale Soft Robot Manipulators With Unknown Payloads*, ser. Dynamic Systems and Control Conference, vol. Volume 3, Rapid Fire Interactive Presentations: Advances in Control Systems; Advances in Robotics and Mechatronics; Automotive and Transportation Systems; Motion Planning and Trajectory Tracking; Soft Mechatronic Actuators and Sensors; Unmanned Ground and Aerial Vehicles, 10 2019. [Online]. Available: https://doi.org/10.1115/DSCC2019-9037

[28] P. Hyatt, C. C. Johnson, and M. D. Killpack, "Model reference predictive adaptive control for large-scale soft robots," *Frontiers in Robotics and AI*, vol. 7, p. 132, 2020.

[29] Z. Chen, X. Ren, M. Bernabei, V. Mainardi, G. Ciuti, and C. Stefanini, "A hybrid adaptive controller for soft robot interchangeability," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, p. 875–882, Jan. 2024.

[30] M. S. Nazeer, D. Bianchi, G. Campinoti, C. Laschi, and E. Falotico, "Policy adaptation using an online regressing network in a soft robotic arm," in *2023 IEEE International Conference on Soft Robotics (RoboSoft)*, Apr. 2023, p. 1–7. [Online]. Available: https://ieeexplore.ieee.org/document/10121927/?arnumber=10121927

[31] Y. Li, X. Wang, and K.-W. Kwok, "Towards adaptive continuous control of soft robotic manipulator using reinforcement learning," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2022, p. 7074–7081. [Online]. Available: https://ieeexplore.ieee.org/document/9981335/?arnumber=9981335

[32] M. S. Nazeer, C. Laschi, and E. Falotico, "Rl-based adaptive controller for high precision reaching in a soft robot arm," *IEEE Transactions on Robotics*, vol. 40, p. 2498–2512, 2024.

[33] F. Pique, H. T. Kalidindi, A. Menciassi, C. Laschi, and E. Falotico, "A learning-based approach for adaptive closed-loop control of a soft robotic arm."

[34] M.-J. Lee and Y.-K. Choi, "An adaptive neurocontroller using rbfn for robot manipulators," *IEEE Transactions on Industrial Electronics*, vol. 51, no. 3, p. 711–717, Jun. 2004.

[35] C. C. Johnson, D. G. Cheney, D. L. Cordon, and M. D. Killpack, "Pneudrive: An embedded pressure control system and modeling toolkit for large-scale soft robots," in *2024 IEEE 7th International Conference on Soft Robotics (RoboSoft)*. San Diego, CA, USA: IEEE, Apr. 2024, p. 786–792. [Online]. Available: https://ieeexplore.ieee.org/document/10521958/

[36] L. Berscheid and T. Kröger, "Jerk-limited real-time trajectory generation with arbitrary target states," *Robotics: Science and Systems XVII*, 2021.

[37] T. F. Allen, L. Rupert, T. R. Duggan, G. Hein, and K. Albert, "Closed-form non-singular constant-curvature continuum manipulator kinematics," in *2020 3rd IEEE International Conference on Soft Robotics (RoboSoft)*, New Haven, CT, USA, May 2020, p. 410–416.

[38] C. Sorensen and M. D. Killpack, "Soft robot shape estimation: A load-agnostic geometric method," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 552–558.

[39] W. J. Palm, *System Dynamics*. McGraw-Hill, 2014.

[40] J.-J. E. Slotine and W. Li, "On the adaptive control of robot manipulators," *The International Journal of Robotics Research*, vol. 6, no. 3, p. 49–59, Sep. 1987.

[41] C. Della Santina, M. Bianchi, G. Grioli, F. Angelini, M. Catalano, M. Garabini, and A. Bicchi, "Controlling soft robots: Balancing feedback and feedforward elements," *IEEE Robotics & Automation Magazine*, vol. 24, no. 3, p. 75–83, Sep. 2017.

[42] N. Oh and H. Rodrigue, "Toward the development of large-scale inflatable robotic arms using hot air welding," *Soft Robotics*, vol. 10, no. 1, pp. 88–96, 2023. [Online]. Available: https://doi.org/10.1089/soro.2021.0134

[43] N. Day, J. Penaloza, V. J. Santos, and M. D. Killpack, "Scalable fabric tactile sensor arrays for soft bodies," *Journal of Micromechanics and Microengineering*, vol. 28, no. 6, p. 064004, 2018.