

State Estimation for Compliant and Morphologically Adaptive Robots

Valentin Yuryev^{1*}, Max Polzin¹ and Josie Hughes¹

Abstract—Locomotion robots with active or passive compliance can show robustness to uncertain scenarios, which can be promising for agricultural, research and environmental industries. However, state estimation for these robots is challenging due to the lack of rigid-body assumptions and kinematic changes from morphing. We propose a method to estimate typical rigid-body states alongside compliance-related states, such as soft robot shape in different morphologies and locomotion modes. Our neural network-based state estimator uses a history of states and a mechanism to directly influence unreliable sensors. We test our framework on the GOAT platform, a robot capable of passive compliance and active morphing for extreme outdoor terrain. The network is trained on motion capture data in a novel compliance-centric frame that accounts for morphing-related states. Our method predicts shape-related measurements within 4.2% of the robot’s size, velocities within 6.3% and 2.4% of the top linear and angular speeds, respectively, and orientation within 1.5°. We also demonstrate a 300% increase in travel range during a motor malfunction when using our estimator for closed-loop autonomous outdoor operation.

I. INTRODUCTION

For robots to support outdoor industries and research activities, such as animal monitoring, climate surveillance, and agriculture, they require the capability to operate within and traverse extreme terrain conditions [1][2][3]. While animals demonstrate such capabilities, and can operate on challenging and varied terrains, typical state-of-the-art robotic solutions, such as rigid-body quadrupeds, are primarily confined to challenging but man-made terrain [4]. This can be partially attributed to their reliance on a singular mode of locomotion and an inability to physically adapt to the wide range of conditions found in outdoor environments [5][6]. Conversely, locomoting animals exploit passive and active compliance to achieve adaptability and beneficial interaction with the environment. For example passive adaptation enables tumbleweeds to roll in the wind, cockroaches to squeeze through tight spaces, and kangaroos to store energy in their legs[7][8]. Active morphological adaptability is exemplified in nature by armadillos or hedgehogs, which can modify their shape to defend themselves or leverage terrain to their advantage [9][10].

This biological inspiration has inspired a paradigm shift in robot design toward systems that leverage passive compliance and adaptive morphologies[11], resulting in a new robot design space for traversing varied terrain [12][13]. To fully exploit the autonomy of such robots, reliable motion

controllers and high-level decision-making modules are required, which are dependent on reliable state estimation. For compliant, morphing robots this becomes challenging due to kinematic changes from active morphology and the lack of rigid-body assumptions. Our goal is to produce a reliable state estimator that accommodates the challenges from robots with passive compliance and active morphological adaptability. For this paper, we focus our efforts on the GOAT robot, a elastic-rod shelled robot capable of driving, rolling and swimming by actively changing its morphology [14].

State estimation for rigid-body robots is a widely researched topic [15][16]. For legged robots, modern actuation and sensor advancements have enabled precise model-based state estimation [17]. Similarly, in autonomous driving, deep-learning methods have provided reliable state estimation in various dynamic environments [18]. While robust and industry-proven, these methods rely on the rigid-body assumption. For compliant, soft robots, the information encoded in the relationship between the robot-centric frame and the rest of the robot’s kinematics—such as the location of the sensors relative to the motors—becomes inconsistent and unobservable. The lack of a rigid frame further complicates state estimation as the computational and sensor payloads can experience large impacts and perturbations.

With the increasingly prevalence of soft robotics, there have been state-estimation methods developed for compliant systems both on the algorithm side such as kinematic-based filtering [19] and sensor side such as IMU arrays and piezoelectric sensors [20]. However, these primarily focus on minor deformations, not significant morphological change during operation. Recent AI advances have produced learning-based approaches that accommodate active morphological variations within a single locomotion mode, such as walking, but not for robots capable of multiple modes like driving and rolling [21]. To the best of our knowledge, no state estimation method exists that combines different locomotion modalities with robot compliance.

We propose a state estimation approach for compliant robots capable of active morphological adaptation, shown on Fig. 1. Our method utilizes a machine learning-based history belief encoder, trained on motion capture data, to approximate a robot’s typical rigid-body states (e.g., linear and angular velocities) and states unique to compliant and morphologically adaptive robots (e.g. outer shell shape). To accommodate passive compliance, we introduce a compliance-centric frame based on the overall robot shape, rather than a classical robot-centric frame tied to sensor positions. To support various locomotion modalities, we augment a feedforward PID controller with soft-body-specific

¹These authors are with with CREATE Lab, Swiss Federal Institute of Technology Lausanne (EPFL), Lausanne, Switzerland.

*Correspondence to: V.Yuryev (valentin.yuryev@epfl.ch)

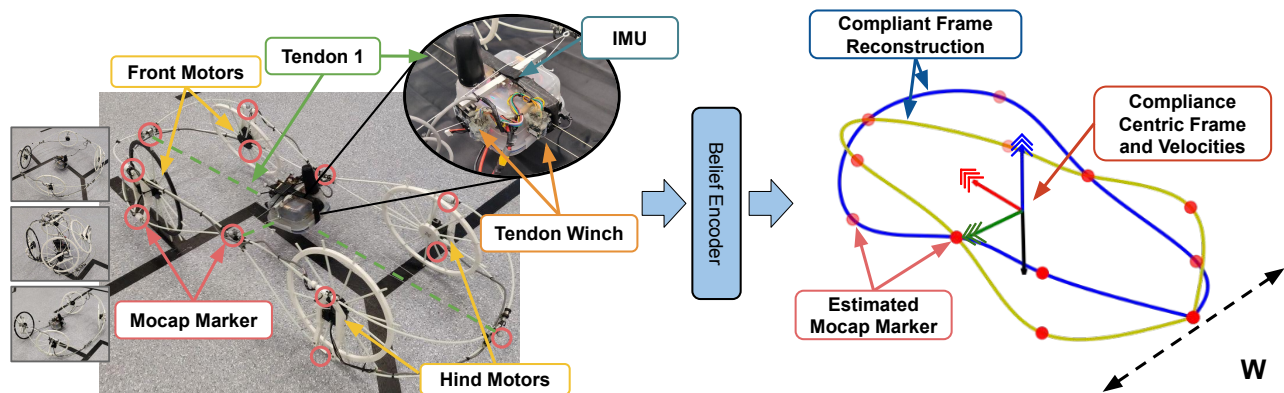


Fig. 1: Overview of our state estimation framework for compliant and morphologically adaptive platforms. Left: placement of motors, sensors and motion capture for data collection. Right: reconstruction of compliance-centric frame, shape and width.

information from the state and shape estimator.

We test our method on the GOAT platform, which features passive terrain compliance and active morphological changes for different locomotion modalities like driving and rolling [14]. We demonstrate state estimation for both rigid-body states (e.g., linear and angular velocities) and morphing-robot-specific states (e.g., outer shell shape) across varied morphological scenarios. Model performance is validated against motion capture ground truth. We showcase improved control via a morphologically adaptive PID controller that enhances velocity tracking during driving. To demonstrate robustness, we test the history belief encoder with malfunctioning morphology sensors in unforeseen corner cases. Finally, we stress-test our closed-loop controller, which relies on our state estimator, in an outdoor scenario against an open-loop baseline.

The remainder of this paper is structured as follows. First, we introduce our method. We then present the experimental setup and results. Finally, we summarize our findings and propose future research directions.

II. METHODS

This section details the methodology for frame and state estimation of a compliant, morphologically adaptive robot. We introduce the compliance-centric frame, our state estimation framework, data collection procedures, noise resistance, and the feedforward PID morphing controller that utilizes our estimator.

1) *Compliant, Morphing Robot Platform*: We develop and deploy our method on the GOAT robot [14], a platform designed for extreme outdoor traversal, shown in Fig. 1. The robot features a compliant elastic rod frame with four direct drive motors. Two winches and tendons enable the frame to be morphed and folded. By shortening or elongating the tendons, GOAT modulates its forward mobility and turning ability, and can morph into a ball to roll—an effective method for traversing steep terrain. The robot operates on grassy, rocky, and sloped terrains, and can swim. Its payload, which contains the compute and sensors, is suspended within the compliant frame via the tendons.

Autonomous control requires reliable state estimation to determine the robot’s velocities and orientation. Shape estimation is also critical to achieve target morphologies reliably, as relying solely on tendon length is error-prone due to encoder inaccuracies, tendon slippage, or malfunction. We propose estimating the state and shape using only onboard sensing: an IMU in the central payload, and motor data from the wheels (current, measured velocities, and commanded velocities) and winches (estimated tendon length). Although demonstrated on the GOAT platform, our state-estimation methodology generalizes to other compliant or morphing robots.

2) *Compliance-Centric Frame*: Compliance and adaptive morphology challenge velocity and configuration estimation due to fundamental kinematic changes and significant base frame oscillation. Typically, state estimation for rigid robots uses a body- or sensor-centric reference frame. For compliant, morphing robots, we propose a compliance-centric frame whose position and orientation depend on key points along the compliant structure.

For GOAT, this frame is constructed by defining a unit X-vector from the “front” and “back” points (Fig. 1). A unit Y-vector is similarly defined from the “left” and “right” points. The Gram-Schmidt method ensures orthogonality between the X and Y components. The Z-vector is the cross product of X and Y. This approach minimizes oscillation noise and simplifies estimator training.

A. RNN and Belief Encoder

Our main contribution is a state and shape estimator for compliant, morphologically adaptive robots. For GOAT, this estimates the compliance-centric frame’s velocities, orientation, and key outer frame points from which we can then estimate the frame shape (labeled as motion capture markers in Fig. 1) using only an IMU, motor and winch measurements.

To address changing morphology, robot compliance and sensor uncertainties we employ a history-based belief encoder (Fig. 2), which has proven effective for estimating terrain from noisy perception data [6]. The belief encoder

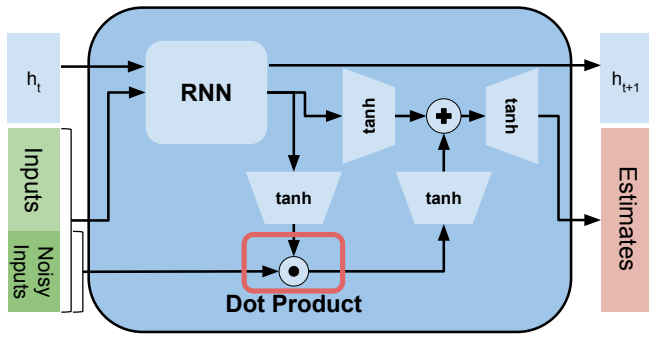


Fig. 2: Belief encoder architecture. Hidden states are passed through onto the next iteration. Inputs are fully fed into the RNN while a subset of the inputs that are considered untrustworthy are further adjusted by the network.

enables the RNN to directly influence untrustworthy measurements via the dot product multiplication. This network compensates for unreliable measurements by incorporating historical data via a recurrent neural network (RNN). The RNN output can directly interact with untrustworthy sensor inputs, enabling real-time error correction. We adapt this approach to handle generally unreliable sensor data.

In our implementation, we replace the self-attention layer with a dot product between estimated and error-corrected components. We use hyperbolic tangent activation units, as it is meant to work better with contentious input sensor data that revolves near zero similar to networks implemented in [22].

B. Data Collection and Training

To collect ground truth data for training the shape and state estimator we use a motion capture system. We manually commanded the robot to drive, turn, and morph, collecting data of it driving, morphing, and rolling within a motion capture system spanning approximately 3×4 m. To ensure a balanced dataset, we included an even mixture of static cases, where the robot held various shapes without commanded motion, and dynamic cases, where it drove forward or turned at various speeds.

To estimate the shape of GOAT’s outer compliant frame, we estimate frame points along the two elastic rods, matching the motion capture marker locations in Fig. 1. We transform the world centric motion capture to a compliance-centric frame via a homogeneous transformation matrix to provide the target training data for our belief encoder: the frame point locations (X, Y, Z) in the compliance-centric frame. To reconstruct and visualize the 3D shape, we fit two cubic splines (one per elastic rod) to the estimated frame points, as shown in Fig. 1.

For velocity estimation, ground truth velocities are computed in the compliance-centric frame using the finite difference method, and an exponentially moving average (EMA) filter applied to reduce the noise. The belief encoder is trained to estimate these smoothed velocities. The network’s final output is the gravity vector expressed in the compliance-centric frame.

The network is trained via supervised learning to minimize the RMSE between ground truth and predicted values, as shown in Eq. (1). Due to a large quantity of near-zero outputs, we augmented the cost function during training in a style similar to mean absolute percentage error (MAPE), as shown in Eq. (1), giving more weight to values near zero [23].

$$\mathcal{L}(\theta) = \min_{\theta} \sum_{i=0}^N \frac{(f(x_i; \theta) - y_i)^2}{1 + |y_i|} \quad (1)$$

where x is the model’s inputs, θ is the model’s parameters, $f(x; \theta)$ is the model’s prediction function and y is the target value from the training data.

C. Robustness via Noise Addition

During outdoor operation, the compliant frame often leads to significant oscillations and sporadic noise in sensor readings. To ensure robustness to these perturbations, we augment our normalized training data by adding Gaussian (STD of 0.1) and sinusoidal noise (amplitude 0.1 and frequency ranging from 0 to 1). Furthermore, sensor alignment and controllability (e.g., the tendon connection between the frame and payload) can be compromised. To ensure robustness to such cases, we also augment the data with time-consistent measurement offsets where we offset all data by a constant that ranges from 0 to 1.

D. Feedforward PID Controller for Morphologically Adaptive Robot

To exploit the state-estimator, we implemented a feed-forward velocity-tracking PID controller (Eq. (3)) for the forward linear and yaw angular components. The integral term of the PID includes a wind-down coefficient $\alpha < 1$ to prevent saturation:

$$e_k = \hat{v}_k - v_k \quad (2)$$

$$\hat{u}_k = \hat{v}_k + K_p e_k + K_i \sum_{i=0}^k e_i * \alpha^{k-i} + K_d (e_k - e_{k-1}) \quad (3)$$

where e_k is the error at control timestep k , \hat{u}_k is the desired velocity of the PID controller \hat{v} and v are desired and measured velocities, K are the PID gains, k is the control timestep.

We then utilize the linear forward component \hat{u}_x and angular yaw \hat{u}_ϕ component of the PID Eq. 3 output \hat{u} to compute the motor velocities v_{left} and v_{right} :

$$v_{\text{left}} = \hat{u}_x - \frac{W}{2} \hat{u}_\phi \quad v_{\text{right}} = \hat{u}_x + \frac{W}{2} \hat{u}_\phi \quad (4)$$

However, a morphologically adaptive robot such as the GOAT does not have a constant width W , shown in Fig 1, as the robot can change and morph its shape during operation. Thus, we use our shape estimation framework to estimate W as the Euclidean distance between the averaged points p_i , as shown in Eq. 5, attached to the motors in Fig. 1:

$$W = \|p_{\text{left}} - p_{\text{right}}\|_2 = \left\| \frac{1}{N} \left(\sum_{i=1}^N p_{\text{left},i} - \sum_{i=1}^N p_{\text{right},i} \right) \right\|_2 \quad (5)$$

This dynamic assessment of W is then used in Eq. 4 to determine the appropriate wheel velocity.

III. EXPERIMENTAL SETUP

Our state estimation framework for compliant robots is applied to a modified version of the GOAT robot shown in Fig. 1 [14]. We retrofitted the robot with Dynamixel XM430-W350 motors to collect reliable proprioceptive data, including motor velocities and currents. Additional sensor data for state-estimation comes from an IMU mounted in the central, tendon suspended payload which reports linear accelerations, angular velocities, and orientation estimates. As a potentially noisy input, as shown in Fig. 2, we use the lengths of the tendons connecting the robot's payload to its compliant morphing outer frame, which is estimated from the open loop winch motors encoders. To collect ground truth training data, we use motion capture system OptiTrack PrimeX22 cameras with sub-millimeter accuracy [24]. To sync the data from the GOAT platform and the motion capture system, we do a post-processing run, where we manually match the change in marker positions to a timestep where a command is given to the robot, such as to drive forward or to morph.

For training, we utilize a mean absolute percentage error loss objective with truncated backpropagation through time for 50 timesteps. All inputs and outputs are normalized. We train our model using a PyTorch-based framework on an RTX 5090 GPU. The total amount of data trained on is approximately 40 minutes of operation at 20Hz. Each data set was separately augmented with gaussian noise, constant offsets and sinusoids, resulting in 4 times the data amounting to 160 minutes. The latent space and RNN hidden dimension are both set to 32. The network input dimension is 21, consisting of: the IMU gravity vector (3), IMU linear acceleration (3), IMU angular velocity (3), left and right wheel motor velocity commands (2), measured motor currents (4), and measured motor velocities (4). The output dimension is 45, corresponding to: 12 frame points (each 3D), linear velocities (3), angular velocities (3), and the gravity vector (3) in the wheel-centric frame.

For computation onboard the robot, we use a Raspberry Pi 5 (8GB) running a PyTorch JIT-traced policy. The system processes inputs and outputs at 20 Hz with negligible inference speed. All inputs and outputs are normalized and denormalized at this frequency to match the training conditions.

For linear velocity tracking experiments, we utilized P, I, and D gains of 1, 0.0, and 0.1, respectively. We tracked the desired yaw rate using P, I, and D gains of 1, 0.01, and 0.01, respectively. All gains were tuned empirically on the robot.

All reported results are a mixture of offline (computed from a recording) and online (computed on the robot) data in no way used during training, representing unbiased estimator's performance in a real-world scenario.

IV. EXPERIMENTAL RESULTS

In this section, we first report the performance of the state estimation, before demonstrating the role of this state

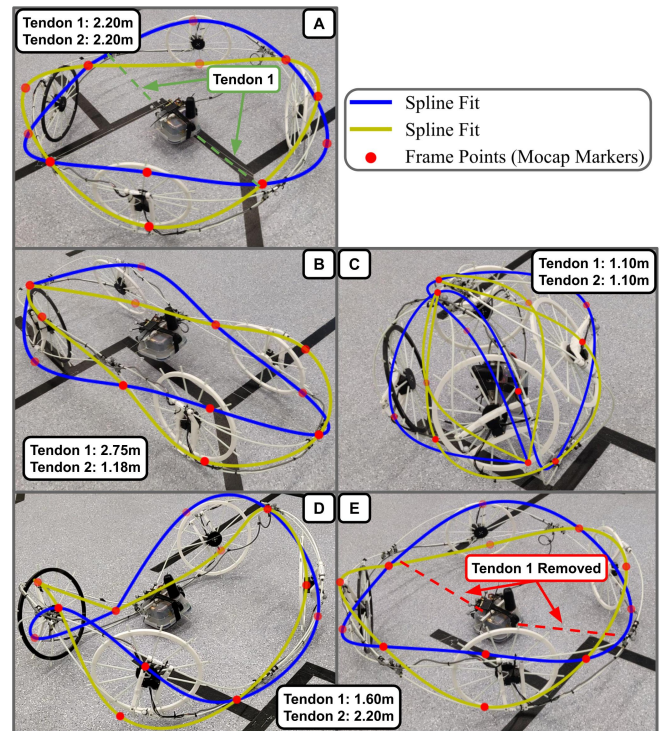


Fig. 3: Overlay of the reconstructed frame shape over the actual robot image for scenarios in Table I. Note that tendon lengths estimated specified in case (D) and (E) are the same. Yet the framework is capable of reconstruction the shape.

estimation for real-time control.

A. State Estimation

To obtain a quantitative assessment of the state estimation, the root mean square error (RMSE) between the estimation and motion capture ground truth was determined. This includes the compliance-centric frame points (12 motion capture points), angle between true gravity and estimated gravity, linear and angular velocities, shown for static and dynamic scenarios. The results are shown in Table I. Static scenarios indicate cases where the robot does not move and are primarily used to validate the shape reconstruction. Dynamic cases, such as robot driving or turning, are when there is a desired velocity command given to analyze gravity and velocity estimation.

1) *Static Shape Reconstruction*: To evaluate shape reconstruction performance, we selected various static scenarios where the robot maintains a fixed shape without moving or morphing. The selected shapes include a circle (A), rover (B), sphere (C), and intermediate pose (D, E), depicted in Fig. 3. Cases (A-C) represent the main operating modes of the GOAT robot, namely turning, driving, and rolling. For visualizing the estimated shape, we apply a twice differentiable cubic spline to the estimated points. The RMSE is calculated from 5 seconds of time series data at 20Hz. For the standard configurations in 3 (A-C), the RMSE error is given in Table I. The mean point reconstruction of the main operating modes (A - C) are in 2 to 5 cm range, which from visual qualitative assessment in Fig. 3 provides a good visual

TABLE I: RMSE of reconstructed robot frame points, gravity vector and velocity components across different scenarios

Scenario	Frame Points (mm)	Gravity ($^{\circ}$)	Linear Velocity (mm s^{-1})				Angular Velocity ($^{\circ} \text{s}^{-1}$)			
			u_x	u_y	u_z	Mean	u_{ψ}	u_{θ}	u_{ϕ}	Mean
(A) Static Circle	26	1.1	3	1	2	2	0.4	0.1	0.2	0.2
(B) Static Rover	44	2.3	4	7	2	5	0.2	0.2	0.6	0.4
(C) Static Sphere	50	1.1	2	11	3	5	0.6	0.3	0.6	0.5
(D) Static Intermediate w/ Tendon	53	1.4	14	6	4	8	0.6	0.5	1.0	0.7
(E) Static Intermediate w/o Tendon	76	2.1	8	6	1	5	0.3	0.2	0.5	0.4
Driving Forwards	28	1.7	26	30	4	20	1.1	0.8	1.6	1.1
Yawing in Place	40	1.3	16	11	6	11	1.0	1.0	4.4	2.2
Driving Forwards w/ dist.	16	0.8	33	16	6	18	1.4	1.1	2.4	1.7
Mean	42	1.5	13	11	4		0.7	0.5	1.4	
Std Dev.	18	0.5	10	8	2		0.4	0.4	1.3	

reconstruction of the shape of the robot. Over all scenarios, the error remains approximately 4.2% of the overall span of the robot that has a diameter of 1 meter when in circular form.

To evaluate the performance of the shape estimation in scenarios where there is error or disturbance in the tendon measurements or configuration, we simulate a scenario where one tendon controlling the morphology is removed, as shown in Fig. 3 scenario (E). We then compare the resulting estimated shape against case (D), where the tendon estimated measurements is the same but the tendon is still attached, resulting in frame deformation and the saddle shape. The overall shape remains circular and closer to the real circular shape of the robot (A), rather than potential intermediate shape (D), that the network would estimate if it only relied on tendon measurements. Frame point reconstruction RMSE for that scenario is 76mm, which is higher than the main operating modes (A - C) but is still visually representative of the robot’s shape. This indicates that our belief encoder adjusts its estimates and discards the inaccurate tendon length measurements provided to it, showcasing the belief encoder-based network capability to adapt to corner-case situations where certain sensor measurements are untrustworthy.

To evaluate improvement of compliance-centric frame over base frame, we calculated the standard deviation of the frame pose during transformation from circle (A) to sphere (C). We observed compliance frame location standard deviation to be 55 [mm] and base frame standard deviation to be 80 [mm], indicating that compliance-centric frame provides a more stable frame to do reconstruction against.

2) *Dynamic Shape Reconstruction*: The shape estimation can also be used “live” throughout the robot’s morphing process. Fig. 4 shows the time series response of the robot morphing from a flat spherical scenario (C) to a circular scenario (A) (Fig. 3), estimated width W of the robot and the RMSE of frame point estimation throughout the process. We observe that the static shapes are easier to predict, while the RMSE of the shape estimator increases during the intermediate morphing phases. This may be explained by the belief encoder not fully trusting the estimated tendon lengths until the IMU information has provided more confirming data, or due to lack of training data in these specific scenarios as they are very transient.

The estimation of the robots width W , also shown in Fig.4, is directly reliant on the performance of our frame point

reconstruction and influences the PID controller. While the intermediate shape error results in incorrect calculation of the width, the final stable configuration RMSE, such as the rover, is low enough that we can reliably use its estimates for adjusting our low level wheel controller described in Eq. (4).

3) *Gravity*: While gravity estimation is not directly used by our controller, it is a valuable component of state estimation in scenarios where the robot changes its shape and rolls. Note that IMU linear acceleration estimates this value almost directly. As such, we expect the measurement to be accurate in static cases, but not exact since we estimate the gravity frame in our compliance-centric frame as we defined in Section II-2. All gravity error values are below 3° , representing adequately accurate gravity vector estimation for estimating robot orientation.

To demonstrate how our method would perform in an outdoor rolling locomotion mode, we have also performed a short rolling of a hill experiment to demonstrate real-world estimation of the gravity vector in a dynamic cases, which are harder to read from the linear accelerations of the IMU. Fig. 5 qualitatively shows, that we can approximate the number of rolls and orientation whilst in a sphere shape rolling down a hill. Note that, while the estimator was trained on normalized gravity vector, nothing is enforcing this behavior for the output of our estimator.

4) *Linear & Angular Velocities*: While we estimate all linear and angular velocities, the relevant components for operating the GOAT robot are the linear forward velocity (u_x) and the angular yaw rate (u_{ϕ}). Therefore, we focus our analysis primarily on these components. The estimation results for all other components across different scenarios can be found in Table I.

For the static cases (A - C), we collected 5 seconds worth of data where the robot does not receive any desired velocity or morphing commands and does not move. In these scenarios the linear and angular velocity error is significantly lower than dynamic cases since the estimates are all close to 0, the easiest case to predict. Since the network was trained on some rolling and non static data, the estimates are never exactly zero, resulting in minor estimation error.

In dynamic scenarios, we provide a linear or angular velocity command to the robot and observe the step responses. The compliance-centric frame is computed from the ground truth motion capture data, which is then used to calculate ground

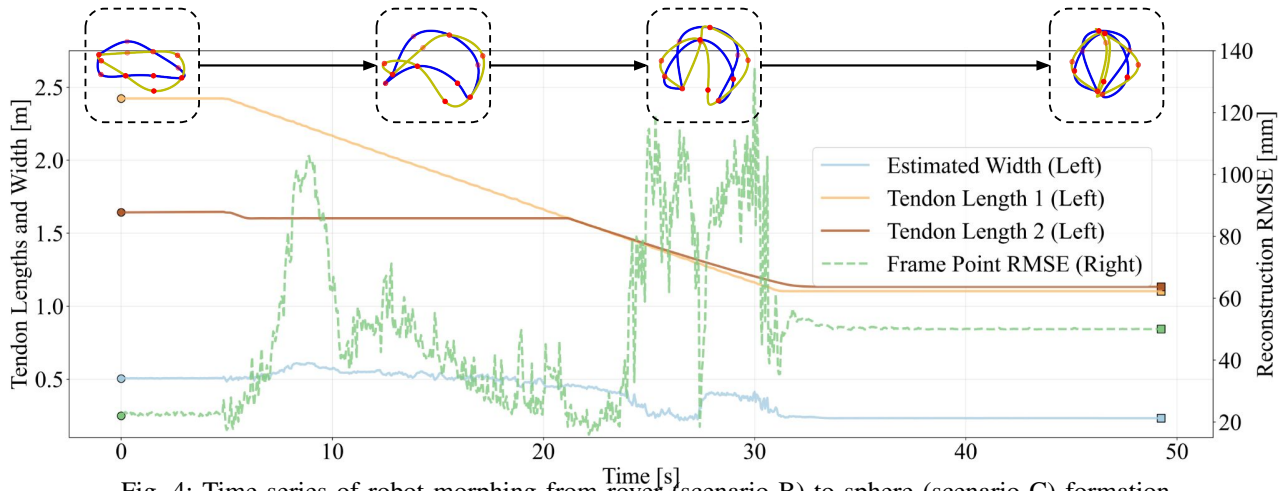


Fig. 4: Time series of robot morphing from rover (scenario B) to sphere (scenario C) formation.

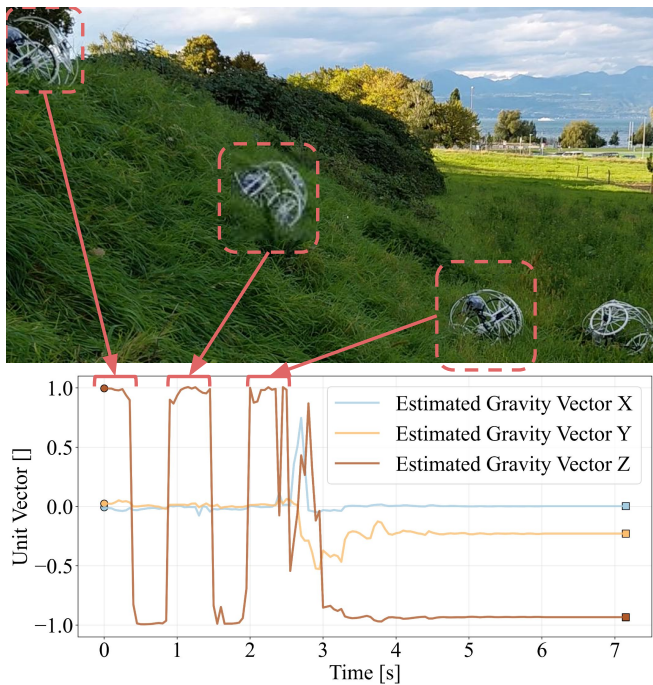


Fig. 5: (Top) Time lapse images of the GOAT robot rolling down a hill and (Bottom) the estimated gravity vector time series.

truth velocities via the finite difference method. We expect the dynamic cases to be harder to predict due to noise from the difficulty of integrating sensor data (IMU only provides linear accelerations), sensor payload disturbances (oscillation due to compliance) while moving and wheels slipping. To evaluate our linear forward u_x velocity estimation, we executed a pure forward motion command within the motion capture facility with and without a simulated disturbance of a broken wheel, shown in Fig. 6. The error for driving forwards is 26mm for nominal scenario (Fig. 6.A) and 33mm for the scenario with a simulated "broken" wheel (Fig. 6.B). Given maximum commanded velocity of 400mm s^{-1} , this results in error of 6.5% and 8.3%, respectively.

To validate our yaw rate estimates, we performed a

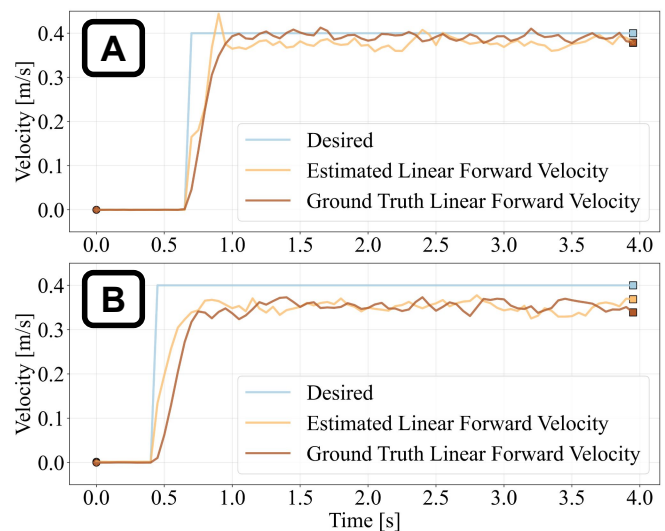


Fig. 6: Estimation of linear tracking while driving forwards with closed loop controller with disturbance (A) and with simulated disturbance of a broken wheel (B).

pure yawing motion inside the motion capture system for comparison against the smoothed finite-difference velocities derived from the ground truth data. The RMSE of the yaw estimation was calculated to be 4.4°s^{-1} , as shown in Fig. 7 which is 2.4% of the maximum commanded yaw rate of 1rad s^{-1} . The resulting estimates are close to the ground truth data and raw IMU data. Note that the figure incorporate IMU readings, which were processed through an exponential moving average (EMA) filter with $\alpha = 0.3$. While the IMU measurements provide a rough quality check, the IMU frame is located inside the robot's payload (as shown in Fig. 1) and does not align with our compliance-centric frame, calculated from the points attached to the motors.

B. Morphing Feed Forward PID Control

In this section, we discuss the validation of our morphing feedforward PID controller and its use of state estimation.

To showcase the benefits of online state-estimation of the compliant frame for robot control, more specifically the

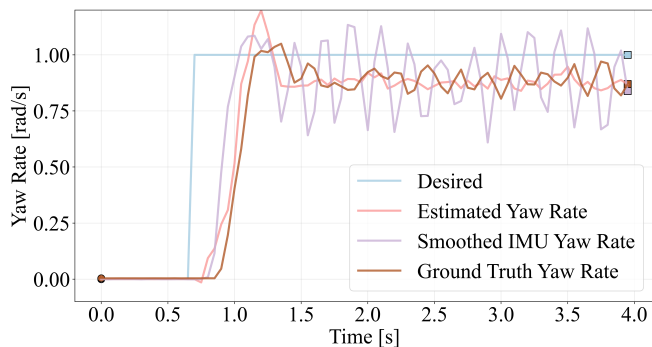


Fig. 7: Estimation of yawing rate during pure yawing command tracked with our PID controller. IMU data is based on IMU installed inside the robot payload and does not fully correspond to the yaw rate that is estimated.

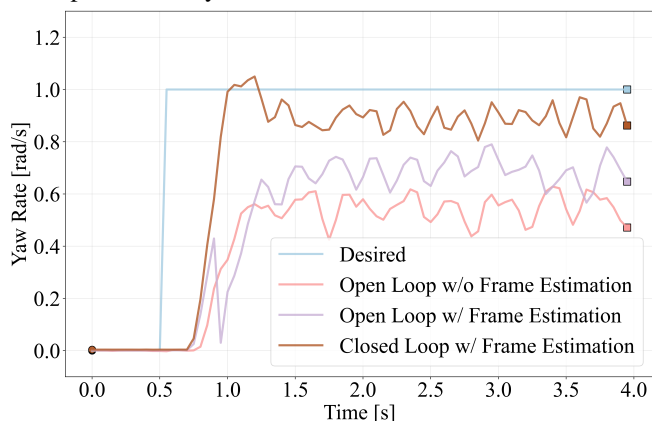


Fig. 8: Motion capture based ground truth yaw rate tracking for scenarios with frame estimation disabled, enabled in open and closed loop scenarios.

dynamic width estimation, we performed a pure yaw rate experiment comparing three configurations: a constant frame width estimated at 0.5m (the width in rover form), dynamic frame estimation without PID control, and the full system with dynamic frame width estimation and the feedforward PID controller. We then compared the ground truth yaw rates from the motion capture system. In Fig. 8, it can be seen that the desired yaw rate tracking steady-state error improves with dynamic frame estimation from 0.45rad s^{-1} to 0.31rad s^{-1} (31% improvement) and is further enhanced to 0.11rad s^{-1} (65% improvement) with the addition of the PID controller.

To test whether our state estimation is sufficient for controlling the robot's desired velocities, we performed a straight-line driving experiment inside the motion capture system. The target forward linear velocity \hat{u}_x was set to a constant 0.4 m/s without the additional PID control from Eq. (3), while the desired yaw rate was set to 0° s^{-1} . To further stress-test the framework, we multiplied the commanded velocity of the front-right wheel by a factor of 0.75 to simulate a motor malfunction. Without feedback, the robot's open-loop controller causes it to veer to the right, as shown in Fig. 9. In contrast, the closed-loop controller—relying on our state estimation—compensates for the yaw-rate error and maintains a straighter trajectory. The final average deviation

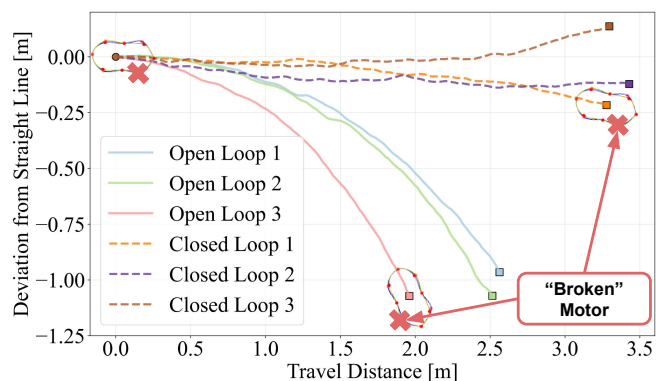


Fig. 9: Top-down view of motion capture based open loop vs closed loop forward driving performance when encountering a malfunctioning front-right motor.

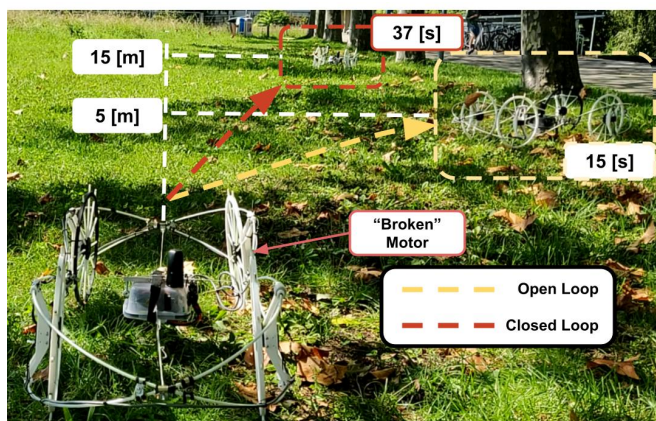


Fig. 10: Travel distance and time before going significantly off-course of an outdoor driving experiment with simulated malfunctioning front-right motor.

for the open loop and closed loop was 0.74 [m] and 0.07[m], respectively. This demonstrates our framework's capability to provide accurate measurements that enable a controller to generate reliable wheel commands.

To validate our system further, we conducted a similar straight-line experiment in an outdoor environment. The desired velocity for the front-right wheel was once again multiplied by a constant factor of 0.75 to simulate a malfunction. The resulting final positions of the robot after driving until significantly veering off-course, along with their estimated trajectories and distance travel projected onto the starting position direction, are shown in Fig. 10. The results show that runs utilizing the PID controller with our state estimation managed to travel 300% farther while maintaining a straighter trajectory in the outdoor environment. For the closed-loop scenario, the final position in Fig. 10 is due to accumulation of tracking error over longer period of time that can be fixed with a global referencing such as SLAM. Additionally, unlike the motion capture facility in Fig. 9, without the ability to orient the front wheels (akin to a car), rover mode faces difficulties to turn via only the left and right wheel velocity difference in grassy terrain.

V. DISCUSSION & CONCLUSION

To fully utilize compliant, morphologically adaptive robots, we must exploit their autonomy and control capabilities. However, the key component of any controller is state estimation. Estimating the state of a compliant, morphologically adaptive robot is challenging due to the failure of the rigid-body assumption, compliance with external disturbances, and constant changes in morphology. In this paper, we propose a state estimation algorithm that can overcome these challenges and function under highly uncertain and dynamic robot configurations. We demonstrate that our state estimation framework improves the robot's controllability and increases the system's robustness to sensor noise and mechanical failures.

Our method is capable predicting the shape and velocities of a robot such as GOAT, and the estimation is accurate in domains that we have trained on. However, as can be seen in Fig. 4, there is increased reconstruction error in transient morphing states of the robot. To overcome some of the challenges with unseen data, we would need to record truly dynamic data in a larger motion capture facility such as rolling down a hill and swimming. Another limitation of this method is the syncing between robot and motion capture data, as well as ensuring that the motion capture quality is high enough for training. While we have managed to collect enough data, often one minute of training data required up to 10 minutes of post processing to ensure temporal and spatial consistency. Another venue to train estimators is to augment the data with rigid-body simulations of spheres and driving.

For future work, we plan to build upon this framework to develop a fully autonomous ecosystem. This would include a motion controller, a high-level morphology selector, and a high-level navigation module, enabling deployment in challenging outdoor terrain for tasks that rigid-body robots are incapable of tackling.

VI. ACKNOWLEDGMENTS

We would like to thank the SYCAMORE Lab and Professor Kamgarpour for lending us the motion tracking facility and providing technical support on the use of the equipment to make this research possible.

REFERENCES

- [1] F. Angelini, P. Angelini, C. Angiolini, S. Bagella, F. Bonomo, M. Caccianiga, C. D. Santina, D. Gigante, M. Hutter, T. Nanayakkara, P. Remagnino, D. Torricelli, and M. Garabini, "Robotic Monitoring of Habitats: The Natural Intelligence Approach," *IEEE Access*, vol. 11, pp. 72 575–72 591, 2023.
- [2] M. Dunbabin and L. Marques, "Robots for Environmental Monitoring: Significant Advancements and Applications," *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 24–39, Mar. 2012.
- [3] A. King, "Technology: The Future of Agriculture," *Nature*, vol. 544, no. 7651, pp. S21–S23, Apr. 2017.
- [4] P. Ramdya and A. J. Ijspeert, "The neuromechanics of animal locomotion: From biology to robotics and back," *Science Robotics*, vol. 8, no. 78, p. eadg0279, May 2023.
- [5] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning Quadrupedal Locomotion over Challenging Terrain," *Science Robotics*, vol. 5, no. 47, p. eabc5986, Oct. 2020, arXiv:2010.11251 [cs].
- [6] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science Robotics*, vol. 7, no. 62, p. eabk2822, Jan. 2022, arXiv:2201.08117 [cs].
- [7] K. Jayaram and R. J. Full, "Cockroaches traverse crevices, crawl rapidly in confined spaces, and inspire a soft, legged robot," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 113, no. 8, pp. E950–957, Feb. 2016.
- [8] D. L. Morgan, U. Proske, and D. Warren, "Measurements of muscle stiffness and the mechanism of elastic storage of energy in hopping kangaroos," *The Journal of Physiology*, vol. 282, pp. 253–261, Sept. 1978.
- [9] C. Stapp, "Defensive diversity: Factors influencing the interspecific variation in the armor of armadillos (cingulata)," phd, 2022, iSBN: 9798374413137 Pages: 47.
- [10] A. Berger, L. M. F. Barthel, W. Rast, H. Hofer, and P. Gras, "Urban Hedgehog Behavioural Responses to Temporary Habitat Disturbance versus Permanent Fragmentation," *Animals*, vol. 10, no. 11, p. 2109, Nov. 2020, publisher: Multidisciplinary Digital Publishing Institute.
- [11] F. Stella and J. Hughes, "The science of soft robot design: A review of motivations, methods and enabling technologies," *Frontiers in Robotics and AI*, vol. 9, Jan. 2023, publisher: Frontiers.
- [12] B. T. Wilcox, J. Joyce, and M. D. Bartlett, "Rapid and Reversible Morphing to Enable Multifunctionality in Robots," *Advanced Intelligent Systems*, vol. 7, no. 7, p. 2300694, 2025, eprint: <https://advanced.onlinelibrary.wiley.com/doi/pdf/10.1002/aisy.202300694>.
- [13] W. Kim, J. Byun, J.-K. Kim, W.-Y. Choi, K. Jakobsen, J. Jakobsen, D.-Y. Lee, and K.-J. Cho, "Bioinspired dual-morphing stretchable origami," *Science Robotics*, vol. 4, no. 36, p. eaay3493, Nov. 2019, publisher: American Association for the Advancement of Science.
- [14] M. Polzin, Q. Guan, and J. Hughes, "Robotic locomotion through active and passive morphological adaptation in extreme outdoor environments," *Science Robotics*, vol. 10, no. 99, p. eadp6419, Feb. 2025.
- [15] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, Mar. 1960.
- [16] T. D. Barfoot, *State Estimation for Robotics: Second Edition*. Cambridge University Press, Feb. 2024, google-Books-ID: qN7oEAAAQBAJ.
- [17] M. Bloesch, M. Hutter, M. A. Hoepflinger, S. Leutenegger, C. Gehring, C. D. Remy, and R. Siegwart, "State estimation for legged robots-consistent fusion of leg kinematics and IMU," *Robotics*, vol. 17, pp. 17–24, 2013, publisher: The MIT Press.
- [18] P. Ghorai, A. Eskandarian, Y.-K. Kim, and G. Mehr, "State Estimation and Motion Prediction of Vehicles and Vulnerable Road Users for Cooperative Autonomous Driving: A Survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 16 983–17 002, Oct. 2022.
- [19] F. Stella, C. D. Santina, and J. Hughes, "Soft Robot Shape Estimation With IMUs Leveraging PCC Kinematics for Drift Filtering," *IEEE Robotics and Automation Letters*, vol. 9, no. 2, pp. 1945–1952, Feb. 2024.
- [20] S. J. Woodman and R. Kramer-Bottiglio, "Stretchable Shape Sensing and Computation for General Shape-Changing Robots," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 8, no. Volume 8, 2025, pp. 1–23, May 2025, publisher: Annual Reviews.
- [21] Z. Luo, Y. Dong, X. Li, R. Huang, Z. Shu, E. Xiao, and P. Lu, "MorAL: Learning Morphologically Adaptive Locomotion Controller for Quadrupedal Robots on Challenging Terrains," *IEEE Robotics and Automation Letters*, vol. 9, no. 5, pp. 4019–4026, May 2024.
- [22] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, p. eaau5872, Jan. 2019, arXiv:1901.08652 [cs].
- [23] A. Botchkarev, "Performance Metrics (Error Measures) in Machine Learning Regression, Forecasting and Prognostics: Properties and Typology," *Interdisciplinary Journal of Information, Knowledge, and Management*, vol. 14, pp. 045–076, 2019, arXiv:1809.03006 [stat].
- [24] O. Krumpek, O. Kroeger, and S. Mohr, "Reproducible Optical Tracking Precision: Evaluating a Static, Near-Parallel Support Structure for OptiTrack PrimeX22 Cameras," Feb. 2025, arXiv:2502.21189 [eess] version: 1.