Vision-Based Proprioceptive Sensing: Tip Position Estimation for a Soft Inflatable Bellow Actuator

Peter Werner, Matthias Hofer, Carmelo Sferrazza and Raffaello D'Andrea

***Abstract*— This paper presents a vision-based sensing approach for a soft linear actuator, which is equipped with an internal camera. The proposed vision-based sensing pipeline predicts the three-dimensional tip position of the actuator. To train and evaluate the algorithm, predictions are compared to ground truth data from an external motion capture system. An off-the-shelf distance sensor is integrated in a second actuator of the same type, providing only the vertical component of the tip position and used as a baseline for comparison. The camera-based sensing pipeline runs at 40 Hz in real-time on a standard laptop and is additionally used for closed loop elongation control of the actuator. It is shown that the approach can achieve comparable accuracy to the distance sensor for measuring the linear expansion of the actuator, but additionally provide the full three-dimensional tip position.**

I. Introduction

Due to their intrinsic properties, inflatable soft robotic systems show promise in overcoming barriers encountered with classic rigid robotic systems [1]. Soft materials provide robots with intrinsic compliance and the ability to interact with their surroundings in a safer and more resilient way. However, these soft systems typically result in a high number of degrees of freedom [2]. Furthermore, modeling dynamic behavior is challenging due to the non-linear material properties. Therefore, sensory feedback is crucial for the control of soft robots [3].

Proprioception in robotics refers to sensing the robot's own internal state, and is an active field of research in soft robotic systems. The state of a soft robot can comprise a single point of interest (e.g. [4]) or the high-dimensional shape of a soft object (e.g. [5]).

A number of different approaches are investigated for retrieving the shape of a soft robot relying only on internal sensors [3]. In the context of optical sensors, stretchable strain sensors based on optical waveguides are employed, where the light transmission properties change when the waveguide is deformed [6]. The changing light intensity due to deflection is used in [7]. The idea is to attach a flexible circuit board, housing a light sensor and various photodiodes, to a soft object. When the object deforms it causes the flexible circuit board to bend and, as a consequence, the light intensity to change. A similar idea is proposed in [8], where an LED and a phototransistor are mounted on the opposing ends of an inflatable linear actuator. When the actuator is inflated, the light intensity measured by the photodiode decreases as a function of the elongation of the actuator.

The authors are members of the Institute for Dynamic Systems and Control, ETH Zürich, Switzerland. Email correspondence to Peter Werner wernerpe@ethz.ch.

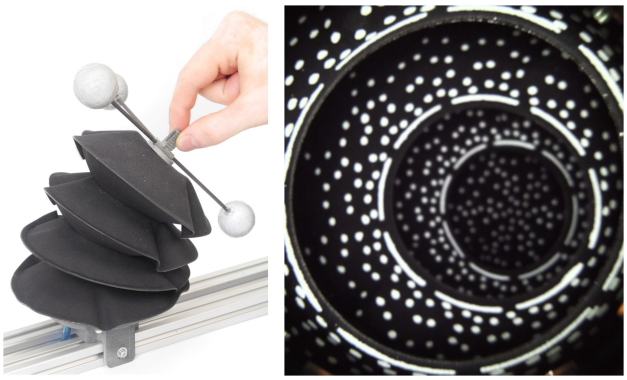

Fig. 1. Left: Manually deflected inflatable linear soft actuator with integrated camera. Markers for an external motion capture system are mounted on top for the acquisition of ground truth data. Right: Image from the integrated camera showing the employed pattern. The proposed sensing method is used to predict the tip position located on the grip, using only images from the integrated camera.

Vision-based sensing relying on a camera to measure the deformation of a soft material is a promising approach investigated actively in the field of tactile sensing. Rich visual information about the strain of the sensor's soft surface is provided by a camera tracking markers embedded in a soft material ([9], [10]) or observing the reflection of light on a deformed surface [11].

Vision-based sensing is promising for two reasons. Firstly, the sensor (i.e. the camera) is not required to mechanically interact with the soft material of the robot. Therefore, the sensor does not need to match the compliance of the soft material employed, in order to avoid stress concentrations or a degradation of the overall compliance of the system. Secondly, vision-based approaches provide high spatial resolution and minimal wiring [12]. However, the use of a camera generally leads to a bulkier structure and requires the points of interest to be in the field of view of the camera.

An approach that combines vision-based tactile sensing with pneumatic actuation is demonstrated in [13]. An internally mounted camera tracks markers attached to a soft membrane, which can be deformed by increasing the internal pressure. The authors of [14] present the three-dimensional shape reconstruction of soft objects based on a self-observing camera. Ground truth data from two external depth cameras is used to train an artificial neural network that runs on a GPU and predicts the object shape from the self-observing

camera images. A method to sense the two-dimensional displacement of the tip position of a soft link is presented in [15]. This also relies on a built-in camera.

In this paper, we present an approach for sensing the tip position of an inflatable, fabric-based, bellow actuator relying on the integration of an RGB camera into the actuator. A distinctive pattern is applied to the interior of the actuator during manufacturing. A number of computationally cheap features are extracted from the raw camera image and used as inputs to a support vector regressor (SVR), where ground truth data from a motion capture system are used to train the model. The proposed approach does not suffer from sensor dynamics that are challenging to model, such as hysteresis. Note that, as opposed to the system presented in [13], the inflatable bellow actuator inherently shows pattern occlusion, which hinders straightforward implementation of classical computer vision approaches. The method presented here can deal with such occlusions, and the lightweight nature of the features employed facilitates real-time implementation. Additionally, the SVR exhibits lower training complexity compared to end-to-end deep learning approaches.

The performance of the camera-based approach is compared to an infrared time-of-flight sensor serving as the baseline for measuring the linear expansion of the soft actuator. It is shown that the camera-based approach can track the tip position of the actuator with high accuracy. Finally, the camera-based state prediction is used in closed loop to control the elongation of the inflatable actuator. The pipeline runs in real-time at $40\,\mathrm{Hz}$ on the CPU of a standard laptop, showing the computational efficiency of the approach proposed. This work originates from the preprint version [16].

The remainder of this paper is organized as follows: The manufacturing of the actuator including the pattern and the integration of the camera is discussed in Section II. The feature extraction from the raw camera image and the applied machine learning pipeline is outlined in Section III. Experimental results of the real-time position estimation and the closed loop elongation control are presented in Section IV. Finally, a conclusion is drawn in Section V.

II. Hardware

The design and fabrication of the actuator, the integration of the sensors and the test setup used in this paper are presented in this section.

A. Actuator

The actuator consists of four circular cushions with a diameter of 140 mm when collapsed. Inflating these cushions causes the actuator to expand longitudinally. Two actuators are manufactured for the two sensors subject to comparison. A camera is integrated in the interior of the first actuator (denoted as actuator 1) and a visual pattern is applied to the fabric layers in its line of sight. A time-of-flight sensor is integrated into a second actuator (denoted as actuator 2) with no pattern applied to the interior surface. Three reflective markers (required by the motion capture system) are attached to a grip which is glued (using Loctite 4850) to the top bellow of each actuator. The assembled system with actuator 1 comprising the internal camera can be seen in Fig. 1.

Both actuators have the same dimensions. They are manufactured using the fabrication procedure described in [17]. To summarize: Each actuator is built from sheets of fabric material that have a sandwich structure. This material is composed of two layers of poplin fabric (polyester cotton blend 65/35) stacked above and below a layer of thermoplastic polyurethane (TPU) film (HM65-PA, 0.1 mm by perfectex) that are fused in a heat press. The resulting processed fabric is inextensible.

The four cushions of the actuator are composed of multiple disc pieces and a lid at the top end. The fabric pieces and additional TPU ring-shaped seam pieces are all cut using a laser cutter. The cushions are constructed by stacking the processed fabric parts with the TPU seams in-between and fusing them sequentially in a heat press. A more detailed description of the fabrication is given in [17] (Layered Manufacturing-Type I).

Before the single fabric pieces of actuator 1 (comprising the camera) are combined, a pattern is applied to the fabric layers on the interior of the actuator that are visible to the integrated camera. This is done to provide the camera with visual features to track. The pattern is applied with white textile spray paint (319921 textile spray paint by Dupli-Color) to provide a high contrast to the black fabric. First, the pattern is cut from adhesive stencil film (S380 by ASLAN) with a laser cutter. In a second step, the stencil is attached to the relevant fabric layers of the bellows and the pattern is applied in four successive, light coats. It is important to keep the paint layers thin, to prevent them from smearing in the heat press during assembly. The pattern consists of white dashed rings around the circular cut-outs and dots with a diameter of 2 mm scattered on the disk and lid pieces (approximately 150 dots for the disc pieces and 200 dots for the lid). The idea behind the rings is to make the individual bellows easily distinguishable from the camera's perspective. The dots are supposed to provide detailed information about the local bending of the fabric. The size of the dots is bounded below by the spray method employed. Compared to manually applying the dots with a brush, the applied approach is faster and leads to a reproducible result. The applied pattern can be seen in Fig. 2. While the pattern employed indeed captures relevant information about the actuator deformation (as will be shown in Section IV-B) it has not been optimized. There might be different patterns that allow for further improvement of the resulting prediction performance. The employed method relying on stencil film and spray painting provides a large design space.

A 3D printed flange (made from PA12, as all 3D printed parts) is glued to the first bellow (using Loctite 4850) for either actuator as an interface for the sensors. The camera and time-of-flight sensor are attached to a separate 3D printed fixture that is connected to the flange of the actuators with six screws to ensure airtightness, see Fig. 3 for the camera example. Pressurized air is supplied to the actuator through

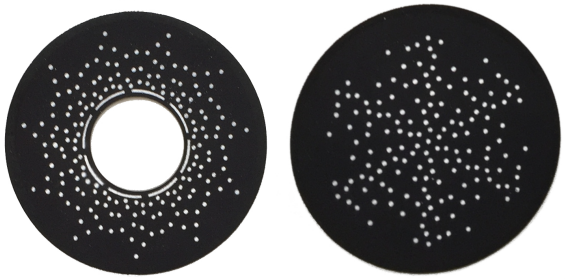

Fig. 2. Pattern applied to the fabric layers of the actuator, which are visible to the integrated camera before assembly. The left image shows one of the disc pieces with the applied pattern and the right image shows the lid piece with the applied pattern. White spray color is used to provide high contrast to the black fabric.

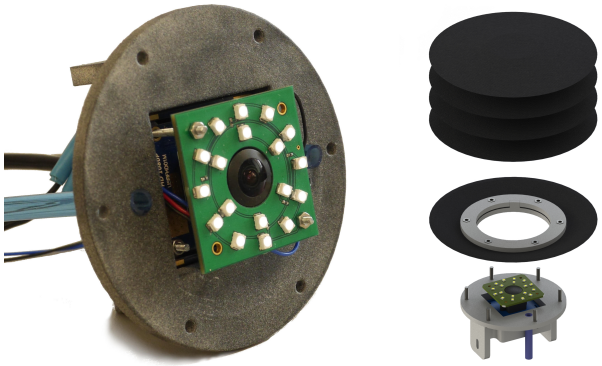

Fig. 3. Left: Camera fixture that is screwed to the bottom of the actuator. It features a USB camera with a 180° fisheye lens and a LED board to illuminate the interior of the actuator. Air to drive the actuator is supplied through the blue tubing. Right: Rendering of the actuator in exploded view. A two-part 3D printed flange is glued to the opening of the first bellow. The camera fixture is secured to it with six screws to ensure airtightness. When assembled, the camera does not protrude beyond the flange.

two blue hoses that are glued to openings in the camera fixture (using Loctite 4850).

B. Camera

The commercial camera used (USBFHD01M-L180 by ELP) is fixed-focus, has a 180° fisheye lens, a resolution of 640x480 and can provide up to 100 frames per second. An LED board is placed around the lens to control the illumination of the interior surface of the actuator and the pattern. Positioning this board on the same plane as the lens eliminates potential shadows. Note that both the camera and LED board are attached to the same end of the actuator, which simplifies integration (i.e. cabling and air tightness) compared to the case where either the lighting source is attached to the other end or multiple cameras are being used.

C. Time-of-flight Sensor

A time-of-flight distance sensor (VL6180X by STMicroelectronics) is integrated into actuator 2 for comparison. As mentioned before the actuator is constructed identically to the first one, with no pattern applied to the interior surface. The convergence time of a single measurement of the time-of-flight sensor (consequently also the sampling time of the sensor) depends on the amount of reflected light. A smaller convergence time is achieved if more light is reflected. Therefore, a piece of reflective tape (Scotchlite 7610 by 3M) is attached to the lid piece facing the time-of-flight sensor and forming the last cushion.

D. Test Setup

The test setup for the actuators includes all required peripherals used to measure and control the internal pressure as well as the motion capture system used as ground truth.

The pressure in the actuators can be controlled in two different ways. Manually, using an analog pressure regulator and automated, using a proportional flow control valve (MPYE-5-1/8-HF-010-B from Festo). The pressure is measured with pressure sensors (8230 from Bürkert) in the actuator and at the source. An embedded platform (consisting of an STM32 Nucleo-144 development board with STM32F413ZH MCU from STMicroelectronics) is used to interface the pressure sensors and valve by analog communication. It will also be used to execute the pressure controller discussed in Section IV-C. Communication between the embedded hardware and the host laptop is implemented over serial communication.

The camera can directly be connected to the host laptop by USB and the time-of-flight sensor is interfaced with a LabJack T7 Pro device. A motion capture system (using T40-S cameras by Vicon) with sub-millimeter accuracy is employed to obtain ground truth data at 200 Hz.

III. Method

The sensing pipeline based on the integrated camera is discussed in this section. First, a motivation of the general approach is provided in Section III-A and the extraction of features from the images is presented in Section III-B. The application of SVR is discussed in Section III-C and the data collection and model training are discussed in Sections III-D and III-E. The integration of the time-of-flight sensor is straightforward and requires little postprocessing. Therefore, it is briefly discussed in Section IV-A.

A. Motivation

The goal of the proposed camera-based sensing approach is to reconstruct in real-time the 3D tip position of the actuator using only images from the internal camera. This tip point is located on the grip in the center of the marker frame attached to the top cushion of the actuator (see Fig. 5, left image). An inertial Cartesian coordinate system I is introduced with the origin O directly above the camera lens. Let r denote the vector pointing from O to the point of interest, namely the tip position, and ${}_I r = (x, y, z) \in \mathbb{R}^3$ being its components in the inertial coordinate frame I, see Fig. 5.

As shown in Fig. 4, during the operation of the actuator the markers can become occluded by the bellows at different vertical coordinates. This is a consequence of the pattern chosen. As an example, using a single large dot on the top layer would solve the occlusion problem, but also reduce

the information about the local deformation. The availability of local information provides more flexibility to extend the current approach and potentially enables the tracking of multiple points and the full orientation of the actuator tip, whereas a pattern with a single dot would for example not be able to capture a rotation of the grip around the z-axis. As a consequence of the occlusion, the use of classical tracking algorithms, e.g. optical flow, would not be straightforward. On the other hand, the re-detection of single markers at each frame for further processing is not feasible due to the limited time and resources available in real-time. Therefore in this paper, a supervised learning approach is proposed, given the availability of ground truth data from the motion capture system.

Fig. 4. Example images recorded by the integrated camera with the tip position at different locations. The point of interest is located approximately in the middle of the center ring from the camera's perspective. It can be seen from the images that the number of dashed rings (indicating the different cushions) changes depending on the elongation state and the rings can be cut by the field of view of the camera. Some of the dots are occluded by the fabric layers closer to the camera depending on the state of the actuator.

We first perform a series of feature construction steps [18, Ch. 6] to extract computationally inexpensive data from the images, providing local information that is not impaired from occlusion. The features are computed by applying different computer vision filters (e.g. Canny edge detection, dilation, etc.) in parallel to the original image. Average pooling is applied to reduce the dimensionality from the camera resolution to a 3×3 grid of elements. In a second step, r is predicted by evaluating SVR models on the extracted features. Compared to end-to-end learning approaches, which bypass the feature engineering step, the SVR generally exhibits short training time and lower data requirements, while retaining real-time prediction capabilities. Note that this approach can deal with parts of the pattern being slightly out of focus, since this effect is inherently compensated for by the learning pipeline.

B. Feature Extraction

The raw image is first cropped to a square image with a resolution of 480x480 pixels. The following procedure is employed to obtain the features from a single image: First, the image is converted to a grayscale image $\mathcal{G}$. In a second step, $\mathcal{G}$ is passed through an array of five image filters producing a total of six images including $\mathcal{G}$. Fig. 6 illustrates how these filters are applied. An adaptive thresholding filter is applied to $\mathcal{G}$, producing $\mathcal{A}$. The result is then dilated and eroded, yielding $\mathcal{D}$ and $\mathcal{E}$. A Canny edge detector is applied to $\mathcal{G}$, resulting in $\mathcal{C}$. The last image $\mathcal{M}$ is obtained by using a binary thresholding filter on $\mathcal{G}$. The threshold of this filter is chosen to be the average of the pixel values of $\mathcal{G}$, denoted by $\bar{\mu}_{\mathcal{G}}$, plus an offset $b_{\mathcal{M}}$. The OpenCV[1] implementation of the above-mentioned filters is used with the parameters listed in the Appendix.

The features are then computed by splitting every image into 3×3 evenly-sized quadratic regions and averaging the pixel values across these regions (average pooling). The choice of the number of grid points can be considered as a tuning parameter. Different numbers of grid elements were investigated and it can be concluded that the prediction accuracy generally increases with an increasing number of elements. A grid of 3×3 elements turned out to be a good compromise between prediction accuracy and computational complexity.

The feature vector $\mu \in \mathbb{R}^{6 \cdot 3 \cdot 3}$ is obtained by concatenating the six individual feature values for each grid element. Finally, the individual entries of μ are normalized by subtracting the mean and dividing by the empirical standard deviation. In order to give a better understanding of the employed features, two exemplary features are shown over time with the corresponding ground truth position as a reference in Fig. 7. Note the different behavior of the two features when the actuator is moved in different directions.

Note that the average pooling step following the computation of the six filters is not equivalent to a direct subsampling of the original camera image. Evaluating the filters on the original, high resolution image and applying the average pooling in a second step provides more information than first subsampling the raw image and then applying different filters. An example for the preservation of the information content after the average pooling step can be seen in Fig. 5 by the clearly differing grid element intensity values resulting for two different features. The support vector regression model discussed in the next subsection relates such differences in the feature vector to different actuator states.

C. Support Vector Regression

To find a mapping between the features μ and r, kernelized support vector regression with a radial basis function (RBF) kernel is used. This is done by using three regressors to predict the three components of r separately. Every regressor has three hyperparameters that need to be tuned. Namely: ϵ, the parameter that controls the ϵ-insensitive loss function, K, the weighting factor that determines the relative cost between the loss function and the L_2-Regularization, and γ the kernel parameter that determines the width of the RBF. More details on support vector regression can be found in [19].

D. Data Collection

To evaluate, train and tune the proposed pipeline, two separate image data sets are collected. This is done by manually inflating the actuator to different elongations (using the analog pressure regulator) and manually moving the grip in the x- and y-directions (see Fig. 5) while simultaneously recording the images from the integrated camera and the

[1] https://opencv.org/

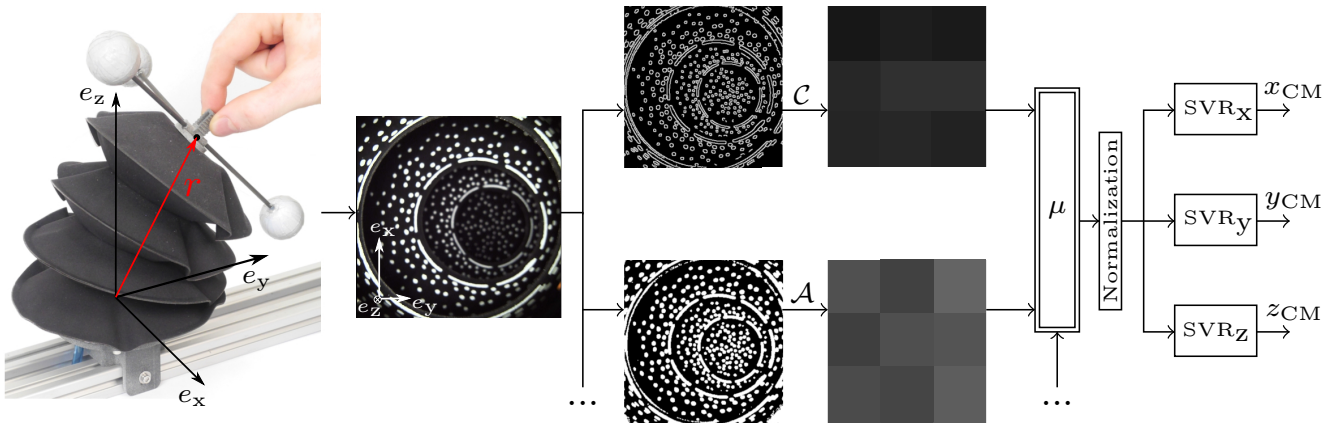

Fig. 5. Illustration of how the machine learning pipeline predicts the tip position r from the image captured by the integrated camera. First, features are extracted from images by applying an array of image filters and averaging the pixel values of the resulting images over a quadratic grid. The resulting intensity values are then concatenated yielding a feature vector μ. Two of the filters (Canny edge detection resulting in $\mathcal{C}$ and adaptive thresholding resulting in $\mathcal{A}$) are visualized above. The location r is then predicted by evaluating three SVR models on μ, to predict its components separately. The subscript CM denotes predictions from the camera-based pipeline.

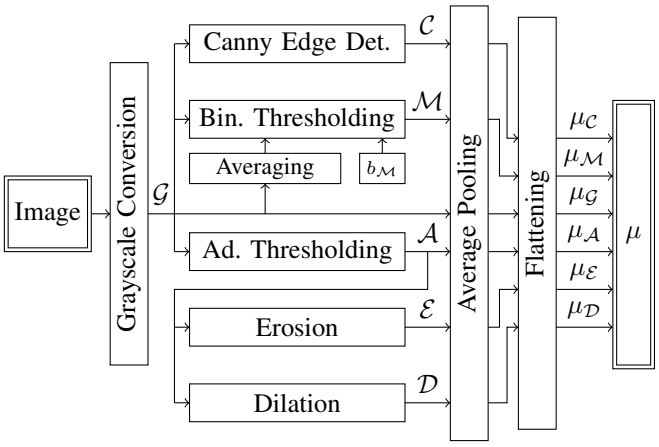

Fig. 6. Before predicting r with kernelized SVR, the raw camera image data is compressed. This is done by converting the images to grayscale and applying a collection of image filters. The resulting images are averaged across a 3×3 grid of evenly-sized quadratic regions (average pooling) and the resulting intensity values concatenated to the feature vector μ.

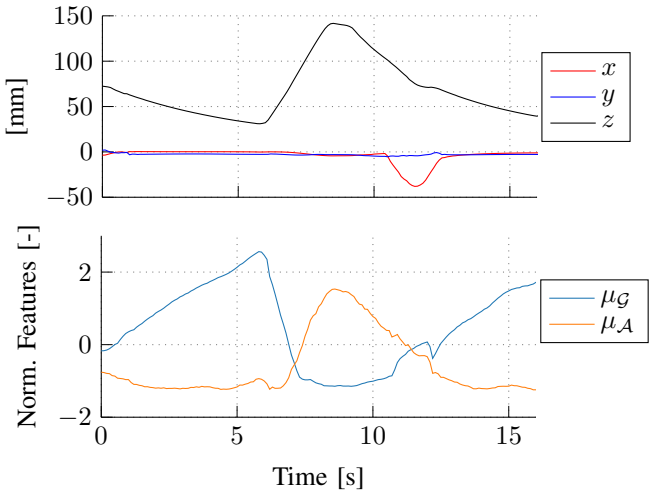

Fig. 7. Two normalized features (gray scale $\mathcal{G}$ in blue and adaptive thresholding $\mathcal{A}$ in orange) are shown as a function over time with the corresponding x-y-z ground truth position measurement. The features shown are extracted from the center element of the 3×3 grid. It can be seen that if the actuator is deflating (decreasing z-position) the orange curve stays approximately constant while the blue curve is increasing. During an inflation, the behavior is the opposite, with the orange curve increasing and the blue curve staying approximately constant. The movement in x-direction causes an increase in the blue curve and sharp decrease when the actuator returns back to the center position. The behavior of the two features indicates that they capture independent information about the actuator's state.

ground truth from the motion capture system at a rate of 20 Hz. Ten minutes of data are collected along a first trajectory corresponding to a data set of 12000 images and the associated ground truth information. This data set is called D_train. A second data set, D_val, is collected for validation over two minutes and consists of 2400 images and the corresponding ground truth.

E. Model Learning

In order to choose the subset of the six features yielding the best trade-off between accurate prediction and computational efficiency, a greedy forward feature selection algorithm is applied. We start with choosing the first feature. For each of the six features, the x, y and z-regressors are trained on the data set, D_train. The hyperparameters ϵ, K and γ for each of the regressors are optimized using 4-fold cross-validation on the training set. The hyperparameter space is extensively searched on a fixed grid. Training a set of three regressors takes about 30 seconds on the CPU of the employed laptop[2].

Once the optimal hyperparameters are determined, the models are evaluated on the validation data set, D_val, and the feature yielding the smallest validation error is chosen. The same procedure is repeated for the second feature where only the five remaining features are considered. Continuing this procedure gives the greedy forward feature selection for six candidate models relying on one to six features. The resulting validation errors are shown in Fig. 8. The model with five features is chosen because it gives a good trade-off between

[2]Intel Core i7-8550U @ 1.80 GHz

accuracy and computational efficiency. The final model uses the feature set $\{\mathcal{M}, \mathcal{G}, \mathcal{A}, \mathcal{E}, \mathcal{D}\}$ and the feature vector μ is adjusted accordingly. The optimal hyperparameters for the model with five features are summarized in Table I. The training of the SVR models is performed in Python with the scikit-learn library[3].

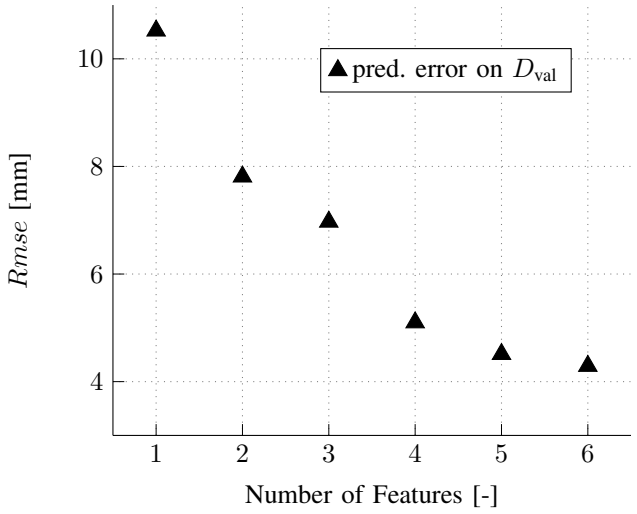

Fig. 8. Evaluation of the six candidate models on the validation data set. The models are obtained using the greedy forward feature selection algorithm. The combined $Rmse$ error shows a decreasing trend over the number of features used.

TABLE I

HYPERPARAMETERS FOR THE MODEL USING FIVE FEATURES

Regressor	ϵ	K	γ
x	2.5	120	0.01
y	2.5	120	0.01
z	2.5	160	0.01

IV. EXPERIMENTS

In this section the camera-based sensing approach is evaluated. A performance baseline is established with a time-of-flight distance sensor. The vision-based approach is then used in two different settings. First, the prediction of r is computed in real-time while the actuator is moved manually. In a second experiment, the proposed sensing approach is used for closed loop elongation control of the actuator. The root-mean-squared error is used as the evaluation metric for the experiments. All the data presented in this section has not been seen during training. In the following the subscript GT refers to ground truth data from the motion capture system.

A. Performance Baseline

The time-of-flight sensor is chosen as a reference, because it presents a straightforward solution to measuring the one-dimensional elongation of a linear actuator in a non-interacting fashion. The sensor is first calibrated by finding an approximate mapping from the raw sensor readings to the ground truth data using linear regression. The calibrated measurements from the time-of-flight (TF) sensor are denoted by

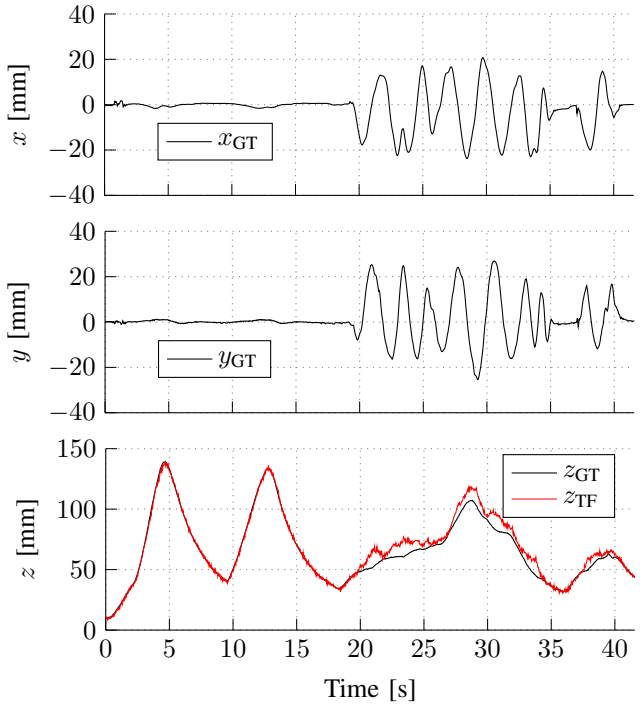

Fig. 9. A time-of-flight distance sensor is integrated into an actuator and calibrated to measure the z-component of r. The calibrated measurements, z_{TF}, are compared to ground truth data from the motion capture system (z_{GT}). Two scenarios can be observed in the plots. First the actuator is left undisturbed. After 19 seconds the grip on the actuator is manually moved in x- and y-directions. The $Rmse$ in z-direction is $1.37\,\text{mm}$ on the first part of the trajectory (undisturbed) and $6.19\,\text{mm}$ on the remainder of the trajectory (disturbed).

z_{TF}. The actuator is not moved in x- and y-directions during calibration as the sensor can only measure distance, without discerning between horizontal and vertical displacement. The sensor is then evaluated on a separate trajectory shown in Fig. 9. The first part of the trajectory (time $\in [0, 19\,\text{s})$) only includes vertical movement of the actuator leading to an $Rmse$ of $1.37\,\text{mm}$. During the second part of the trajectory (time $\in [19, 42\,\text{s})$), the actuator is also moved laterally. As expected, it can clearly be seen that the performance degrades significantly, yielding an $Rmse$ of $6.19\,\text{mm}$.

B. Real-Time Prediction

To observe the performance and the real-time capability of the camera-based sensing approach, the actuator is inflated (using the analog pressure regulator) and moved manually in the x- and y-directions. The reader is referred to the video attachment to gain an impression of the experiments conducted.

The resulting sensing pipeline runs reliably at $40\,\text{Hz}$ on a laptop[4]. The resulting performance on a sample trajectory is shown in Fig. 10. Predictions from the camera-based pipeline are denoted by the subscript CM. The $Rmse$ in the individual components are $4.01\,\text{mm}$ in x, $4.52\,\text{mm}$ in y and $2.56\,\text{mm}$ in z-direction. It can be seen that the camera-based approach can also reliably predict the 3D tip position, when the actuator is moved laterally. The performance is

[3] https://scikit-learn.org/

[4] Intel Core i7-7600 @ 2.80 GHz

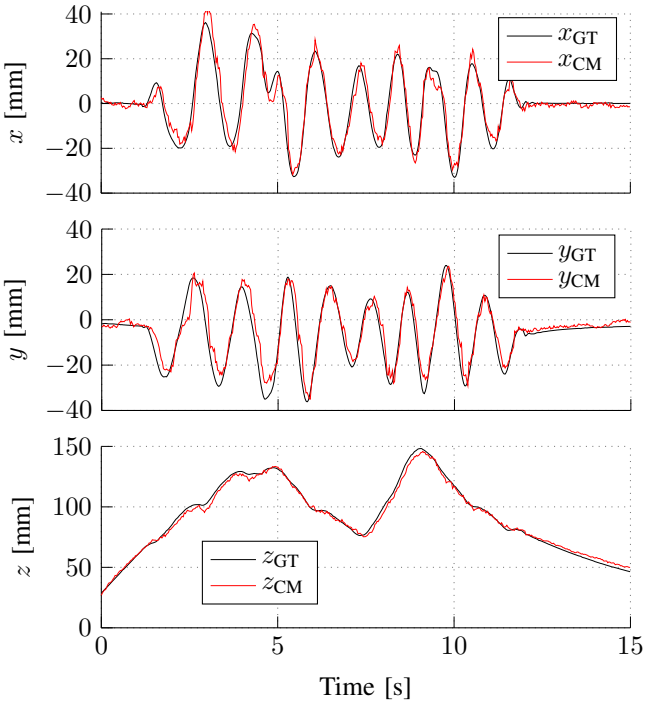

Fig. 10. Prediction of r computed at 40 Hz from camera images while the actuator is moved laterally. The predictions are plotted against ground truth data (subscript GT). The $Rmse$ in the individual components are 4.01 mm in x, 4.52 mm in y and 2.56 mm in z-direction.

considerably reduced if the actuator is elongated less than 20 mm. Upon inspection of training images recorded at or below this elongation, it is seen that the lighting conditions are drastically different to the images recorded at elongations above 20 mm. This issue is a current limitation of the approach.

C. Elongation Control

The linear elongation of the actuator (in z-direction) is controlled using the camera-based prediction as sensory feedback. The x- and y-components of the tip position of the linear bellow actuator can not be controlled by adjusting the internal pressure of the actuator and therefore, are disregarded for this experiment. A cascaded control architecture is used that separates the faster pressure dynamics from the elongation dynamics (see Fig. 11). A proportional-integral (PI) position controller with a quadratic feed forward component computes a pressure setpoint based on the elongation prediction by the camera and a given desired elongation. The pressure setpoint is then tracked in an inner control loop by a proportional-integral-derivative (PID) controller. Based on pressure feedback from a sensor connected to the actuator, it adjusts the spool position of the valve. Since the required pressures to cause an elongation of the actuator are very close to ambient pressure, a bypass is installed between the valve and the actuator, which releases air to the environment. This increases the required output pressure of the valve slightly and therefore simplifies pressure control (see [20] for a detailed discussion of the topic). The position controller is implemented in C++ and executed at 50 Hz in the same thread as the sensing pipeline. The predictions of the x- and

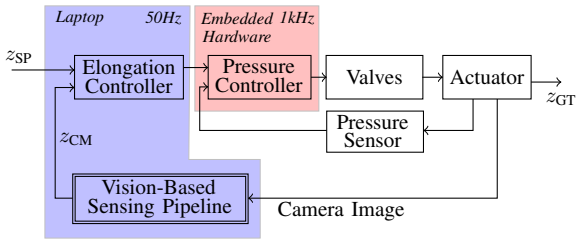

Fig. 11. The elongation of the actuator is controlled using a cascaded control architecture. The faster pressure dynamics are controlled in an inner control loop running at 1 kHz on an embedded hardware (indicated by the light red area). A proportional-integral-derivative (PID) controller adjusts the valve position to track the pressure setpoint computed in the outer control loop. The elongation is controlled in the outer control loop running at 50 Hz. The camera-based sensing pipeline predicts the current elongation (z_{CM}) which is fed to a proportional-integral (PI) controller. Given a desired actuator elongation of the actuator, the controller computes a pressure setpoint which is sent to the inner control loop. The sensing pipeline and the elongation controller are executed on a laptop computer (indicated by the light blue area). A motion capture system provides ground truth measurement of the elongation (z_{GT}).

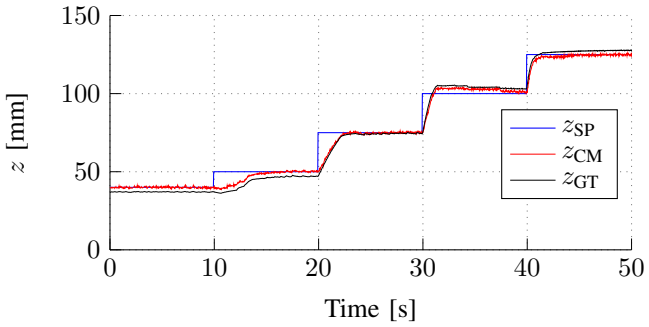

Fig. 12. Elongation control using the camera-based sensing pipeline for feedback. The z-component z_{CM} of the position r is predicted from camera images at 50 Hz and used for feedback control. Note that z_{SP} denotes the setpoint trajectory. The $Rmse$ between z_{GT} and the prediction z_{CM} is 2.49 mm.

y-components of r are disabled to increase the sampling rate of the z prediction (from 40 to 50 Hz). The pressure controller is implemented in C and executed at 1000 Hz on the embedded platform. Data is logged in a separate thread running at 100 Hz and the serial communication with the embedded hardware is also implemented in a separate thread.

The results of a series of elongation steps are shown in Fig. 12. It can be seen from the results that the setpoint trajectory can reliably be tracked with the camera-based sensing pipeline as feedback. Note that the slight mismatches between predictions and ground truth, which are elongation dependent, contribute to a similar $Rmse$ as in the real-time experiment shown in Fig. 10.

Smaller elongations in the range of 20 mm to 40 mm can also be measured by the camera-based sensing pipeline. However, the required pressures for controlling the elongation in this range are too close to ambient pressure for reliable tracking. Note that this is not a limitation of the sensing approach, but of the valves employed, and could be addressed by using a dedicated pressure control valve for a very small range.

V. Conclusion

A camera-based sensing approach for an air-driven linear soft actuator has been presented as a proof of concept. The proposed sensing pipeline first extracts features from images generated with an integrated camera using classical image filters and average pooling. The resulting average intensity values are then used as input features for three SVR models that predict the tip position of the actuator, r. The proposed approach performs similarly to a performance baseline made with an off-the-shelf distance sensor used to measure the linear extension. Additionally, the camera-based approach benefits from the rich visual information of the pattern, which allows it to predict the full 3D tip position. Moreover, it was demonstrated that the pipeline presented can successfully be used for closed loop elongation control of the actuator.

In order to extend the feasible range of the camera-based prediction to values below 20 mm, the lighting conditions for such elongations should be adjusted. This could for example be done by using an adaptive lighting strategy at these ranges (currently the LED light intensity is fixed) and is subject for future work.

As a proof of concept, the sensing approach presented in this work has been applied to track a single point of interest. However, since the internal pattern provides rich visual information spanning all the bellows, the approach is promising for the simultaneous tracking of multiple points, assuming that ground truth data is available for each of these points. Future work will investigate the generalization to multiple points, the prediction of the orientation beside the position, the application of similar approaches to different actuator types (e.g. angular or twisting actuators), as well as the integration of the sensing approach into a complete system such as used in [21]. The white pattern applied to the interior surface of the actuator was arbitrarily designed and its optimization with respect to a tailored metric may provide additional information about the quantities of interest and hence further improve the prediction performance, especially for a wider range of applications.

At the current state the sensing algorithm is only robust to interactions applied on the grip. General types of interactions were not considered in this paper and will be the subject of future work.

Acknowledgment

The authors would like to thank Michael Egli and Matthias Müller for their contribution to the development of the prototype and the test setup.

Appendix

A. Filter Parameters

The OpenCV framework was employed for the image processing with the following image filter parameters. The adaptive thresholding filter ($\mathcal{A}$) uses *maxValue* of 255, *adaptiveMethod* Gaussian, *thresholdType* Binary, *Blocksize* of 57 and a C of 2. The binary thresholding ($\mathcal{M}$) uses *maxValue* of 255, $b_{\mathcal{M}}$ of 100, *Threshold* of $\bar{\mu}_{\mathcal{G}} + b_{\mathcal{M}}$ where $\bar{\mu}_{\mathcal{G}}$ is the average across all entries of $\mu_{\mathcal{G}}$ (see Fig. 6) and a *Type* of Binary. The Canny edge detection filter ($\mathcal{C}$) has the *lowThreshold* set to 100, the *highThreshold* set to 130 and a *kernelSize* of 3. The dilation filter ($\mathcal{D}$) uses a *kernelShape* square and *kernelSize* of 5×5. The erosion filter ($\mathcal{E}$) also uses a *kernelShape* square and *kernelSize* of 5×5.

References

[1] P. Polygerinos, N. Correll, S. A. Morin, B. Mosadegh, C. D. Onal, K. Petersen, M. Cianchetti, M. T. Tolley, and R. F. Shepherd, "Soft Robotics: Review of Fluid-Driven Intrinsically Soft Devices; Manufacturing, Sensing, Control, and Applications in Human-Robot Interaction," *Advanced Engineering Materials*, vol. 19, no. 12, p. 1700016, 2017.

[2] D. Rus and M. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, pp. 467–475, 2015.

[3] H. Wang, M. Totaro, and L. Beccai, "Toward Perceptive Soft Robots: Progress and Challenges," *Advanced Science*, vol. 5, no. 9, p. 1800541, 2018.

[4] M. T. Gillespie, C. M. Best, and M. D. Killpack, "Simultaneous position and stiffness control for an inflatable soft robot," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1095–1101.

[5] Z. Zhang, J. Dequidt, and C. Duriez, "Vision-based sensing of external forces acting on soft robots using finite element method," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1529–1536, 2018.

[6] H. Zhao, K. O'Brien, S. Li, and R. F. Shepherd, "Optoelectronically innervated soft prosthetic hand via stretchable optical waveguides," *Science Robotics*, vol. 1, no. 1, 2016.

[7] M. K. Dobrzynski, R. Pericet-Camara, and D. Floreano, "Contactless deflection sensor for soft robots," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011.

[8] H. D. Yang, B. T. Greczek, and A. T. Asbeck, "Modeling and Analysis of a High-Displacement Pneumatic Artificial Muscle With Integrated Sensing," *Frontiers in Robotics and AI*, vol. 5, p. 136, 2019.

[9] C. Chorley, C. Melhuish, T. Pipe, and J. Rossiter, "Development of a Tactile Sensor Based on Biologically Inspired Edge Encoding," *Advanced Robotics, ICAR*, 2009.

[10] C. Sferrazza and R. D'Andrea, "Design, Motivation and Evaluation of a Full-Resolution Optical Tactile Sensor," *Sensors*, vol. 19, no. 4, 2019.

[11] W. Yuan, S. Dong, and E. H. Adelson, "GelSight: High-Resolution Robot Tactile Sensors for Estimating Geometry and Force," *Sensors*, vol. 17, no. 12, 2017.

[12] Z. Kappassov, J.-A. Corrales, and V. Perdereau, "Tactile sensing in dexterous robot hands," *Robotics and Autonomous Systems*, vol. 74, pp. 195–220, 2015.

[13] B. W. McInroe, C. L. Chen, K. Y. Goldberg, R. Bajcsy, and R. S. Fearing, "Towards a Soft Fingertip with Integrated Sensing and Actuation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[14] R. Wang, S. Wang, E. Xiao, K. Jindal, W. Yuan, and C. Feng, "Real-time Soft Robot 3D Proprioception via Deep Vision-based Sensing," vol. abs/1904.03820, 2019.

[15] J. Oliveira, A. Ferreira, and J. C. Reis, "Design and experiments on an inflatable link robot with a built-in vision sensor," *Mechatronics*, vol. 65, p. 102305, 2020.

[16] P. Werner, M. Hofer, C. Sferrazza, and R. D'Andrea, "Vision-based proprioceptive sensing for soft inflatable actuators," *arXiv preprint arXiv:1909.09096*, 2019.

[17] H. D. Yang and A. T. Asbeck, "A New Manufacturing Process for Soft Robots and Soft/Rigid Hybrid Robots," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[18] H. Liu and H. Motoda, *Feature selection for knowledge discovery and data mining*. Springer Science & Business Media, 2012, vol. 454.

[19] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, no. 3, 2004.

[20] M. Hofer and R. D'Andrea, "Design, Modeling and Control of a Soft Robotic Arm," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[21] M. Hofer, L. Spannagl, and R. D'Andrea, "Iterative learning control for fast and accurate position tracking with an articulated soft robotic arm," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 6602–6607.