Bootstrapping the Dynamic Gait Controller of the Soft Robot Arm

Rudolf Szadkowski[1], Muhammad Sunny Nazeer[2], Matteo Cianchetti[2], Egidio Falotico[2], and Jan Faigl[1]

***Abstract*— In this paper, we propose a novel dynamic gait controller for the repetitive behavior of soft robot manipulators performing routine tasks. Compliance with soft robots is advantageous when the robot interacts with living organisms and other fragile objects. However, predicting and controlling repetitive behavior is challenging because of hysteresis and non-linear dynamics governing the interactions. Existing prior-free methods track the dynamic state using recurrent neural networks or rely on known generalized coordinates describing the robot's state. We propose to model the interaction induced by the repetitive behavior as gait dynamics and represent the dynamic state with Central Pattern Generator (CPG) tracking the motion phase and thus reduce the complexity of the robot's forward model. The proposed method bootstraps an ensemble of the forward models exploring multiple dynamic contexts that are expanded as it searches for repetitive motion producing the target repetitive behavior. The proposed approach is experimentally validated on a pneumatically actuated soft robot arm I-Support, where the method infers gaits for different targets.**

I. Introduction

The elastic construction of soft robots actuated by pressure [1] or tendons [2] provides infinite degrees of freedom, inherent dexterity[3], and naturally compliant interaction with the environment [4]. The soft robot properties expand the robotic domain by deployments where careful interaction is needed [5] or in wet environments [6]. Elasticity has direct advantages for locomotion, where the effectors are less susceptible to damage and can be non-disruptive to the environment [7], [8]. However, the behavior of complex elastic bodies is non-linear. It induces dynamic phenomena, such as time delay or hysteresis, which represents a great challenge for dynamic gait control of soft robots [9].

The soft robot control is underactuated and depends on the dynamic interaction between the elastic body and the operational environment. Dynamic control requires describing the robot's dynamic state, which is generally unknown. The dynamic state can be inferred from sufficient sensory observations, such as pose, velocity, and torque of the robot's joints [10]. Alternatively, with prior knowledge about the physical nature of the state, it is possible to track the system state with the Kalman filter [11]. Without prior assumption about the state or sufficient sensory observations, a dynamics model can be trained using the *Recurrent Neural Networks* (RNN) [12], [13], thus representing the dynamic state internally.

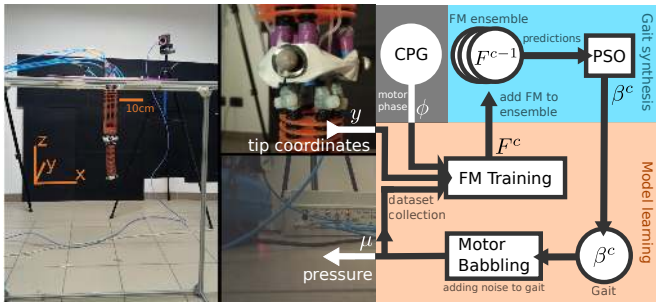

Fig. 1. The I-Support platform and the bootstrapping algorithm diagram. A network of 8 cameras provides the tip coordinates for motion reconstruction.

The RNN can be trained for arbitrary dynamics but is computationally and memory intensive. Updating the RNN for each new data is time intensive, and the model may catastrophically forget earlier learned experiences. Even more, the computationally expensive capacity of the RNN may be unnecessary for a subclass of general dynamics, where the robot interacts with the environment using repetitive motion, the gait dynamics. On the other hand, in the continual learning paradigm [14], [15], we assume the robot continually experiences various dynamic contexts, so the robot updates its knowledge continually.

Therefore, we propose to explore the soft robot dynamics by bootstrapping algorithm expanding an ensemble of dynamic models used to control the gait motion. The proposed method uses a biologically inspired concept, the *Central Pattern Generator* (CPG), which is a neural oscillator involved in animal gait locomotion studied in biomimetic controllers for gait locomotion [16]. We use the CPG as a dynamic state estimator, thus decoupling the gait control into phase control and amplitude control [17], [18]. The *Forward Model* (FM) is then composed of a simple CPG model tracking the gait phase, while the sensory-motor interaction is modeled by a *Feed-forward Neural Network* (FNN).

The ensemble of FMs is generated by the bootstrapping algorithm that explores multiple dynamic contexts for a given sensory target. In this work, the dynamic context is determined by the performed gait, and we show that the

*This work was not supported by any organization

[1]Faculty of Electrical Engineering, Czech Technical University in Prague, Technická 2, 166 27 Prague, Czech Republic {szadkrud|faiglj}@fel.cvut.cz

[2]The BioRobotics Institute, Scuola Superiore Sant'Anna, Italy {MuhammadSunny.Nazeer|Egidio.Falotico|matteo.cianchetti}@santannapisa.it

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 730994 (TERRINet). The work has been supported by the Czech Science Foundation (GAČR) under research project No. 21-33041J and the OP VVV funded project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics." The authors also acknowledge the support of the Brain Inspired Robotics Lab and BioRobotics Institute for the provided robotic platform, experimental equipment, and experimental facility.

perturbation dynamics of each gait differ from the other. Thus, multiple FMs must be learned to successfully synthesize such gait that produces sensory targets. We demonstrate the proposed method on the soft robot arm, the I-Support [1] depicted in Fig. 1, which is tasked with reaching a sequence of given coordinates. The experiments show that the CPG-based model encodes dynamics sufficiently to reach coordinates reachable only by dynamic motion using open-loop control. Moreover, we show that the bootstrapped ensemble can be reutilized for other novel tasks.

II. Related Work

Soft robotics presents many features but challenges for dynamics modeling and control that the current research tackles. The published solutions can be categorized by: (i) the domain to kinematic or dynamic; (ii) prior knowledge as model-free or model-based; (iii) control approach to direct policy learning or model-based control. In the presented short overview of the existing methods, we focus on the category related to the proposed solution based on learning the dynamic model (or control) without prior knowledge about the robot dynamics. For a review of other categories, we encourage the reader to study [9], [4].

Dynamic motion control can be learned indirectly by learning the FM utilized in control or directly by learning the inverse model providing the control policy for each input. In [2], the authors compared multiple schemes for foam robot hand control such as *Reinforcement Learning* (RL) or supervised neural network learning using the learning data collected from the finite element simulation. The RL method was adapted for gait control in [18], where the CPG-based controller decomposes the closed trajectory into optimized amplitude and phase. The FM can be directly integrated into the feedback loop using model predictive control as demonstrated for trained pneumatic arm controller in [11]. If the FM is differentiable, the distal learning [19] can exploit feedback error back-propagation to regulate the control [20]. Both direct control policy learning and FM-based control rely on online sensory feedback to infer the control policy.

Combining inverse and forward model learning allows the FM to train the control policy offline. In [12], the RL obtains a controller for the pneumatically actuated soft robotic arm, where the controller is trained in an environment simulated by the FM implemented as an RNN. A similar approach to the RNN-based FM training is used to develop an I-Support controller [21]. The controller can be trained by FMs continually, as shown in [15], where the inverse model learns novel control policies without forgetting previous policies. The task of learning of inverse model and FM are complementary.

The FM has another essential role in ensemble dynamics. The ensemble of specialized models has physiological evidence [22] where the internal models are activated by their relevancy to the task. The relevancy assessment is modeled in [23], where each FM provides performance prediction or assesses the current estimation error, which regulates the output of the whole ensemble [24]. The proposed bootstrapping algorithm builds an ensemble to be further utilized in control; hence, learning the FM is necessary.

The dynamics FM must be able to model the dynamic phenomena, and without sufficient dynamic state observation, it depends on a sequence of inputs. Thus many dynamic FM learners implement RNNs [12], [21]. The RNN architecture can be adapted to the robot morphology as shown in [13], where each RNN models a part of the soft robot. However, modeling RNN capabilities require large datasets and are computationally expensive. An RNN must unroll to predict whole trajectories (instead of just the next state), which causes slower prediction and control inference than in the case of FNNs. The FNN inverse model has continual learning capabilities, in [15]; however, the RNN-based FM is retrained every time, forgetting previously learned dynamics. We show that the dynamics FM can also be implemented as the FNN and used with a continual learning scheme.

In the present work, we focus on dynamics where the behavior results from repetitive motion, the gait dynamics. The gait dynamics model relaxes the task of the general dynamics model while keeping the dynamics complex enough, allowing the robot's mobility [7], [8], [25]. We show that the CPG and FNN are sufficient to encode the gait dynamics and control the soft robot arm to reach the desired behavior and targets.

III. Problem statement

For the given T-periodic *sensory target* $y^*(t) = y^*(t+T) \in \mathbb{R}^N$, the open-loop controller must produce periodic *motor command* $u(t) = u(t+T) \in \mathbb{R}^M$ resulting in the *sensory observation* $y(t)$ that is close to the sensory target $y(t) \approx y^*(t)$. We aim for model-based control with a dynamic forward model estimating the motor command effects

$$\frac{\mathrm{d}x}{\mathrm{d}t} = g(x(t), u(t)), \tag{1}$$

$$y(t) = f(x(t), u(t)), \tag{2}$$

where the dynamic state x is unknown and partially observed by the sensors y. The dynamic component of the model corresponds to the dynamic nature of the soft robot, where hysteresis and non-linear behavior can be expected. Without prior knowledge, it is hard to identify functions g and f describing the soft robot arm dynamics. Furthermore, the dynamic state x can contain infinite-dimensional information relevant to the robot-environment interaction.

In the presented work, we focus on modeling the repetitive dynamic behavior induced by repetitive motion, the gait dynamics. During undisturbed gait, the motor commands repeat with the T-period, which entrain the sensory observations to be T-periodic. We can distinguish two components of the dynamic state x that influence the sensory-motor relation, entrained dynamic state x' and dynamic context c. First, the *entrained state* x' describes an unknown dynamic variable that is entrained by the motor, and thus it is also T-periodic. Besides, the *dynamic context* c reflects a dynamic situation relevant to the sensory-motor interaction that persists over multiple gait cycles. Since we focus on the dynamic context

determined by an average gait motion, we can assume the entrained state and dynamic context represent the hidden state sufficiently. Their computational representations are described in the following section.

IV. Proposed Method

The proposed method iteratively generates forward models that estimate dynamics induced by perturbing a given gait. The dynamics are estimated by the FNNs combined with the CPG providing the state phase awareness. Each generated model is added to the ensemble, which is used to synthesize a new gait. The particular building blocks are described in detail in the following parts of the section.

A. Gait Dynamics Forward Model

During undisturbed gait, the motor command trajectory $u(t) = u(t+T)$ creates a closed loop, $\mathbf{u}_0 \subset \mathbb{R}^M$; likewise, the entrained state x' follows a single loop denoted $\mathbf{x'}_0$. The position on the loop $x' \in \mathbf{x'}_0$ can be uniquely represented by the *state phase* $\phi \in [0, 2\pi)$. Thus, as long as the state x' stays in the vicinity of the loop $\mathbf{x'}_0$, the entrained state x' of the unknown dimensionality can be represented by a one-dimensional state phase ϕ. For the state phase estimation, we can use the simplified mathematical model of the CPG

$$\frac{\mathrm{d}\phi}{\mathrm{d}t} = \omega + p(t)\sin(\phi), \tag{3}$$

where ω is the angular velocity of the estimated phase, and $p(t) = 0$ denotes the external input to which the CPG can synchronize.

The state phase ϕ is sufficient to describe the state only when the entrained state x' is on the loop $x' \in \mathbf{x}_0$; however, if the motor command temporarily changes $u(t) \notin \mathbf{u}_0$, the trajectory x' can change as well. In practice, the legged agent sustains its gait by biomechanic properties of the body or reflexes. Thus, after the regular gait motion is restored, the state x' returns to the loop $\mathbf{x}_0$. Hence, we focus on such cases where the following assumption holds.

Assumption 1 (J-Gait Closure Property): A transient motion perturbation has a transient effect on the sensory observation that vanishes within J periods.

Assuming J-gait closure, the deviation from the loop $\mathbf{x}_0$ at the time t is a result of the J previous gaits $U(t) = \{u(\tau)|\tau \in (t-TJ, t]\}$. The state phase $\phi(t)$ and the history of J gaits of the motor commands $U(t)$ describes the current state $x'(t)$ that is observed by the sensory measurement $y(t)$.

We approximate $U(t)$ by taking discrete samples from it to keep the FM computationally tractable. The signal is sampled at the state phases $\Phi_k = 2\pi k/K; k \in \{1, \ldots, K\}$ with the *granularity* K. The *sampling time* of the kth sampling phase Φ_k can be defined as

$$\tilde{\Phi}_k(t) = \max\{\tau|\tau \in (t-T, t] : \phi(t) = \Phi_k\}. \tag{4}$$

We further define *motor embedding* as a sequence of commands sampled by the estimated phase $\mu(t) = (u(\tilde{\Phi}_k))_k^K \in \mathbb{R}^{M\times K}$. The motor history U is then represented by the window of the last J gait embeddings, $\hat{U}(t) = (\mu(t-Ti))_i^J \in \mathbb{R}^{M\times K\times J}$.

We estimate the sensory observation with a phase-aware FM

$$\hat{y}(t) = F(\hat{U}(t), \phi(t)) = \sum_k^K \varphi_k(\phi(t)) F_k(\hat{U}(t)), \tag{5}$$

$$\varphi(\phi)_k = \begin{cases} 1 \text{ if } k = \arg\min_{k'}(\phi - \Phi_{k'})^2, \\ 0 \text{ else,} \end{cases} \tag{6}$$

where $F_k : \mathbb{R}^{J\times K\times M} \to \mathbb{R}^N$ is a model for the state phase Φ_k. The phase model F_k is implemented as the FNN. The gait dynamics FM denoted F is then implemented as multiple FNNs switched by the CPG estimated state phase ϕ.

B. Learning the Forward Model

The FNNs of the FM denoted F are trained by datasets collected during perturbing the given *base gait* $\beta \in \mathbb{R}^{M\times K}$ that induces effects in sensory observations. For each gait cycle, during motor babbling, random noise is added to the base gait resulting in perturbed gait $\mu \in \mathbb{R}^{M\times K}$. The resulting motor embedding μ can be transformed into a motor command by

$$u(t) = \sum_k^K \varphi'_k(\phi(t))\mu[k], \tag{7}$$

$$\varphi'_k(\phi) = \frac{\exp(\phi - \Phi_k)^2}{\sum_l^K \exp(\phi - \Phi_l)^2}, \tag{8}$$

where ϕ is the state phase estimated by the CPG, $\mu[k] \in \mathbb{R}^M$ denotes the motor command at the kth phase, and $\varphi'_k(\phi)$ is a softmax weight that smooths the transition between the command phases. The result motor command $u(t)$, sensory observation $y(t)$, and estimated phase $\phi(t)$ are stored into recording $R = (u(t), y(t), \phi(t))_{t_0}^{t_1}$, where t_0 and t_1 denote the start and end of the motor babbling, respectively.

The recording R is used for learning the FM denoted F. After computing the sampling times (4), the signals $y(t)$ and $u(t)$ are embedded into the set of triplets $D = \{(\hat{U}_i, y_i, \Phi_{k,i})\}_i^{|D|}$. Each kth phase model F_k is trained by the dataset $\{(\hat{U}, y)|(\hat{U}, y, \Phi) \in D : \Phi = \Phi_k\}$. The FM denoted F is then added into ensemble $\mathscr{F}$.

C. Gait Synthesis

The problem is to find such a gait β^* that results in the behavior with the sensor observation $y^*(t)$. First, we define the evaluation of the gait β as the squared difference between the target and predicted outcome of J-times repeated gait, $(\beta)_j^J$, estimated by the FM denoted F: $L(\beta; F) = \sum_k^K ||F_k((\beta)_j^J) - y^*(\tilde{\Phi}_k)||^2$, where $y^*(\tilde{\Phi}_k)$ is the sensory target at the state phase Φ_k. Let $\mathscr{F}$ be a set of the forward models; then, the loss function can be defined as the average evaluation

$$\mathscr{L}(\beta) = \sum_F^{\mathscr{F}} w_F L(\beta; F), \tag{9}$$

where $w_F \in \{0, 1\}$ is a parameter set by a model selection strategy, detailed in Section V. We use *Particle Swarm Optimization* (PSO) to minimize the loss function $\mathscr{L}$ and synthesize a new gait β'.

D. Bootstrapping Forward Model Ensemble

The FM denoted F^c is optimized for the prediction in the dynamic context c in which the model was trained. In this work, the dynamic context is determined by the base gait β^c. Therefore, each model F^c is optimized for predicting sensory-motor interaction during its respective base gait β^c. The FM ensemble $\mathscr{F} = \{F^c\}_c^C$ is bootstrapped by repeating two stages: (i) Gait synthesis and (ii) Model learning.

At the first iteration $c = 1$, the base gait is set to zero $\beta^1 = \mathbf{0}$. Then, the motor babbling is added during the model learning, and the FM denoted F^1 is trained. The newly trained model is added into the ensemble $\mathscr{F}$, synthesizing the next base gait β^2. In consequent iterations, we generate gait and model sequence $(\mathbf{0}, F^1) \rightarrow (\beta^2, F^2) \cdots \rightarrow (\beta^C, F^C)$.

On the time scale of bootstrapping iterations, the algorithm behaves as an iterative closed-loop controller, where the record R^c is the feedback that updates the ensemble by adding a new FM denoted F^c resulting in a new base gait β^c. Since every gait synthesis is performed with respect to the sensory target $y^*(t)$, we hypothesize that the sequence of the base gaits β^c gradually improves the real performance. Moreover, the output of the bootstrapping algorithm is then the FM ensemble $\mathscr{F}$ of the size C that can be utilized for different sensory targets, as shown in the experimental evaluation presented in the following section.

V. Results

The proposed method was deployed on the I-Support arm and evaluated in three experiments: (i) trajectory following; (ii) dynamic reaching for a given point; and (iii) multiple reaching motion inference. The first two experiments demonstrate the bootstrapping of multiple CPG-based forward models that encode and exploit the dynamics of their respective contexts, see Section V-C and Section V-D, respectively. In the last experiment described in Section V-E, the trained bootstrapped ensemble is reused in various tasks showing that the ensemble has generalization qualities.

A. Experimental Setup

I-Support is a soft robotic arm composed of two connected, individually pneumatically actuated modules with three pneumatic control signals each; see Fig. 1. Each module is composed of three coupled pneumatic chambers [1], [26] that inflate or deflate depending on the pressure set by the control signal in the range 0 kPa to 90 kPa. The change in the chambers' volume results in a change in the effector pose.

The pose of the I-Support arm was tracked by the Vicon motion capture system consisting of eight cameras employed to capture different perspectives of the robot for shape reconstruction and actuator tracking. The tip of each module is tracked via three Vicon markers placed in isosceles triangular form. Four additional markers are placed on the support frame to establish a reference coordination system. Thus, ten markers were observed by the Vicon's cameras. The marker was then post-processed into module tip spatial coordinates; see Fig. 1 for the coordinate axes.

B. Preliminary Experiments and Hyperparameters Settings

We empirically searched for the controller and bootstrapping hyperparameters in preliminary experiments. The embedding granularity was set to $K = 8$ and the CPG angular velocity to $\omega = 4.5\,\mathrm{rad\,s^{-1}}$ (measured gait-cycle duration: $T \approx 1.4\,\mathrm{s}$), which produced resonant behavior of the sensory observation y. The gait-closure parameter $J = 3$ representing the sensory estimation computed from the three last gaits was found sufficient for the estimation and control inference.

Regarding the 3-gait closure, during preliminary experiments, we observed that as the arm performs periodic motion, the Z coordinates of both segment centroids slowly decrease over the entire session. Because the non-periodic behavior is outside of the scope of the proposed model, the Z coordinate is omitted in the experiments.

The phase models, F_k^c, are implemented[1] as the FNN with the input $M \times K \times J = 144$ and output $N = 4$ sizes. The FNN has one hidden layer of the size 100 units with the hyperbolic tangent activation function. The neural network is trained by `Adam` with 1000 epochs. The training data was gathered using motor babbling for 80 gait cycles. The base gaits, $\beta^c \in \mathbb{R}^{48}$, are synthesized by the PSO[2] that searched for the optimum for 500 iterations. The `select-last-three` models strategy, $w_{FC-3} = w_{FC-2} = w_{FC-1} = 1$, was used for computation of the loss value (9), where C is the bootstrapping iteration. One bootstrapping iteration consisting of the PSO gait synthesis, data collection, and FM learning was measured to take 156 s on average[3].

C. Double-loop Experiment

The I-Support is tasked to reach a set of target locations, where each must be reached at a certain phase during the gait cycle; see Fig. 2. For each phase, k, the target XY coordinates are set for both segments, depicted as green markers in Fig. 2. Therefore, each segment tip should follow its target trajectory loop using a periodic motor command, a gait.

We ran the experiment six times for ten bootstrapping iterations. The base gaits and their result trajectories are shown for the performed double-loop experiment in Fig. 2, where we show the early, best, and last iteration of the one selected experiment. The result of the early iteration shows no apparent pattern in the command, and only a few observed trajectory points are close to the target. The later iterations show a pattern resembling antagonist commands between chambers of the same segment (u_1, u_2 or u_5, u_6), where the positive peak of one signal co-occurs with a negative peak of the other. The result trajectories fit better with the target; the observed trajectory points are close to the target points of their respective phase.

Bootstrapping improves the base gait after ten iterations in all six experimental runs. The gradual improvement of the *performance error* $K^{-1}\sum_k^K ||y(\tilde{\Phi}_k) - y^*(\tilde{\Phi}_k)||^2$ was

[1] `Sklearn` library implementation of the multi-layer perceptron regressor from the `https://scikit-learn.org/stable/` is used.

[2] `Pyswarms` implementation from `https://github.com/ljvmiranda921/pyswarms` was utilized.

[3] Using Intel Core i7 with 16 GB RAM.

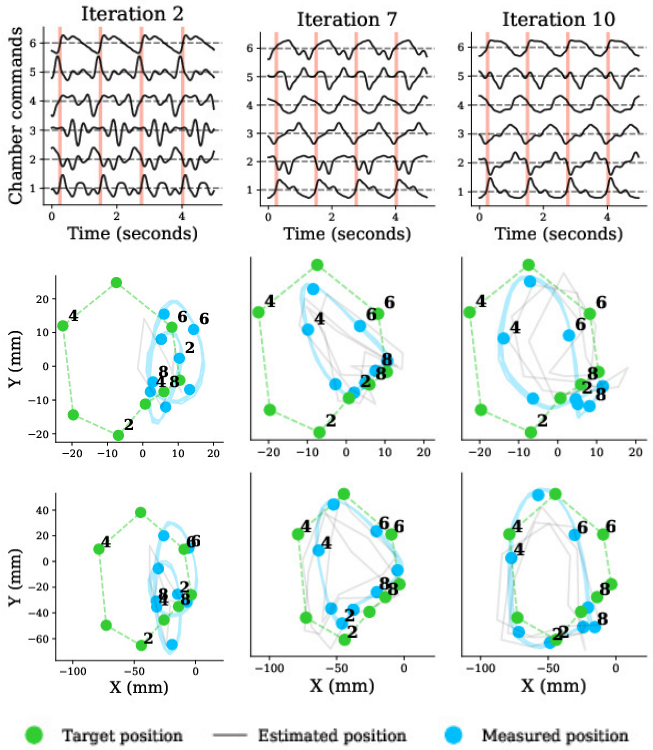

Fig. 2. Double-loop experiment motor-sensory overview for early, best, and last iterations. Motor commands on top with gait-cycle starts are highlighted in red. Below are sensory observations, targets, and estimations with phases indicated by numerals.

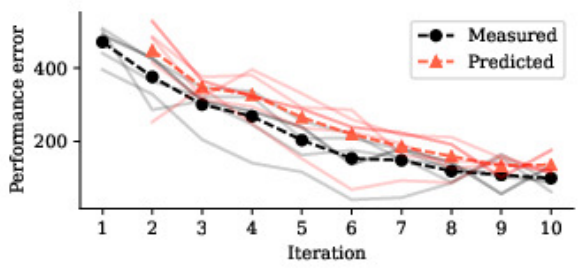

Fig. 3. The predicted and measured performance error of the double-loop experiment.

observed in all six experiments; see Fig. 3. Interestingly, despite the PSO minimizes the *predicted performance error* $K^{-1}\sum_k^K ||\hat{y}(\tilde{\Phi}_k) - y^*(\tilde{\Phi}_k)||^2$, the predicted performance error is on average higher than the observed performance error.

The *estimation error* $K^{-1}\sum_k^K ||y(\tilde{\Phi}_k) - \hat{y}(\tilde{\Phi}_k)||^2$ grows with the difference of the dynamic context. The cross-evaluation between each cth iteration model F^c and the c'th iteration dataset $D^{c'}$ is visualized in a matrix of the mean squared estimation errors in Fig. 4. For each model-dataset pair (c, c'), we evaluate the estimation error and the squared difference between the base gaits β^c and $\beta^{c'}$. Both indicators are projected into the plot on the right in Fig. 4. The growing trend, where the estimation error grows with the dynamic context difference, is apparent. Across all six experimental runs, the average correlation between the error and context difference is 0.80 ± 0.04.

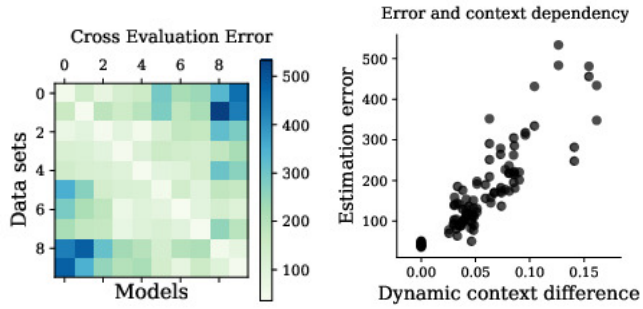

Fig. 4. Relation between the dynamic context difference and estimation error. The mean model estimation error in its dynamic context is 45.24 with the variance 2.61. The correlation between the dynamic context distance and estimation error is 0.80 with the variance 0.04.

D. Single-point Experiment – Dynamic reaching

The I-Support was tasked to reach one point with the distal segment at a particular phase of the gait motion. The tip of the distal segment should create a trajectory loop, which reaches the given point in the second phase. For three different points, the arm was able to optimize its performance, which can be observed from the results depicted in Fig. 5.

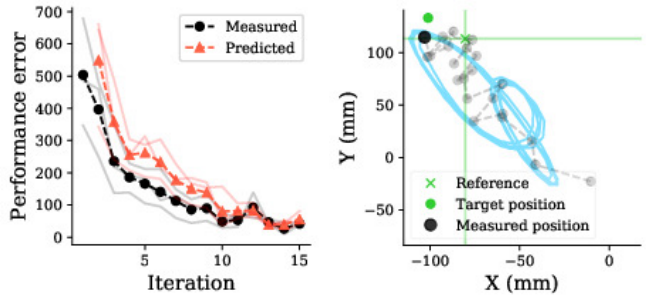

Fig. 5. Single-point experiment performance progress (left). The progress of the trajectory point at Φ_2 goes beyond the reference point (right). The final trajectory is visualized in blue.

The exploitation of the dynamic model is demonstrated by the task that can be solved only by a dynamic motion as follows. First, a reference point that the I-Support can reach with static control was measured. Two chambers of the proximal segment and one chamber of the distal segment were given maximal pressure (90 kPa), and after the arm settled, the centroid of the distal segment was taken as the reference point. Then, the arm was manually stretched further from the resting position, and the tip position was captured as the target point. The assumption is that reaching the target point requires dynamic movement, and thus the model must encode the system dynamics. After 22 iterations, the base gaits reached the vicinity of the target point, Fig. 5.

We examine the capability of the trained FM denoted F^1 to reproduce dynamic phenomena, particularly the time delay between the command and the sensory effect. For the selected phase k of the base gait β^1, the kth command was randomly changed, and F^1 predicted sensory change for each consequent phase. The experiment was repeated ten times, and the averages of the consequence magnitude for each sensory phase are shown in Fig. 6, where we can

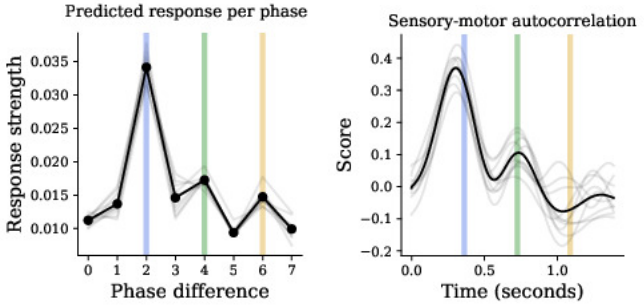

Fig. 6. Comparison between predicted and measured time delays. The color bands highlight the phase delays and the corresponding time delays.

see that the highest sensory response is predicted to occur after two phases, which corresponds to the time delay of 0.35 s. Comparing the auto-correlation between the norms of $u(t)$ and $y(t)$ ground truth signals, which show covariance between the control and sensor signals at different times, the correlation peaks fit the predicted response delay.

E. Single-point Experiment – Ensemble Generalization

Fig. 7. Three different setups reutilizing the trained models.

We examined whether the FMs trained in Section V-D can be reused for other tasks. We implemented the `select-competent-three` models strategy used in loss (9) computation, which selects three models from the entire ensemble. For each model, we evaluate the *sensory expectation* as $\tilde{y}^c(\tilde{\Phi}_k) = F_k^c(\beta^c)$, and we choose three models that have the sensory expectation closest to the given sensory target. Such a heuristic allows the gait synthesis to access the entire ensemble and pick the best FM that knows the dynamic context. Different manually set targets were tried, and after the gait synthesis, the robot approached the target point, as seen in Fig. 7.

VI. Discussion

The I-Support production results in embodiments with specific dynamics that further change with the soft-material exhaustion, thus incremental dynamics learning is essential. The gait dynamics estimated by the proposed CPG-based FM encode the dynamics sufficiently for synthesizing a gait that approaches the target trajectory in open-loop control. Because the optimized gait motion changes each iteration and the context change increases the estimation errors of the already learned models, bootstrapping new models is necessary. Bootstrapping shows performance improvement in each iteration, and the resulting FM ensemble can be reutilized for new tasks.

The reutilization of the ensemble opens the possibility of adding FMs learned during various tasks, thus improving the overall performance incrementally without forgetting the previous experience. The strategy of the FM selection is essential; each FM models its respective dynamic context, some more relevant to the current task than others. The two proposed strategies, `select-last-three` and `select-competent-three` (introduced in Sections V-B and V-E, respectively), are intuitive heuristics selecting the three relevant models, where the relevancy is given either by novelty or expectation. The heuristics performance and other strategies in multi-task scenarios should be further examined.

The model training data are generated by motor babbling, where each iteration uses the same babbling parameters. Therefore, the dynamics are sampled with the same coarseness, resulting in repeatedly unprecise FMs whose estimations are shown in Fig. 2. Adaptive regulation of the babbling parameters can improve FM performance.

Adding feedback to the control time scale, as opposed to the bootstrapping time scale, might improve the performance three-fold. First, since the FM has a simple FNN architecture, it can be integrated with a feedback control scheme such as the distal learning [19] tuning the control online [20]. Second, the estimation error of the FM can be assessed online, and thus the relevant FM can be selected by the current best estimator [24]. Thirdly, the feedback can be integrated into the CPG, thus synchronizing the estimated phase with observations [17]. The estimated phase can be shifted, or the angular speed can change through variable $p(t)$, see (3). Therefore, the dynamic context, gait amplitude, and phase would adapt to the situation online by adding feedback. We plan to investigate the benefits of sensory feedback in our future work.

VII. Conclusion

The proposed algorithm for bootstrapping CPG-based FMs explored the dynamics of I-Support, soft robot arm, and synthesized gaits approaching a given trajectory. As the newly synthesized gaits change the dynamic context, the estimation error grows for the current models, and a new FM adapts to the new context. This iterative adaptation improves the overall performance of the ensemble. The result CPG-based FMs capture dynamic properties, such as time delay, sufficiently to synthesize a gait dynamically reaching beyond the farthest steady state point. Moreover, the bootstrapped ensemble can be reused for novel tasks, where the gait synthesis selects FMs corresponding to the given target and generates novel gaits approaching the target. In future work, we aim to examine ensemble learning in multi-task scenarios and integrate the sensory feedback into the gait controller.

REFERENCES

[1] M. Manti, A. Pratesi, E. Falotico, M. Cianchetti, and C. Laschi, "Soft assistive robot for personal care of elderly people," in *IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, 2016, pp. 833–838.

[2] C. Schlagenhauf, D. Bauer, K.-H. Chang, J. P. King, D. Moro, S. Coros, and N. Pollard, "Control of tendon-driven soft foam robot hands," in *IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, 2018, pp. 1–7.

[3] J. Zhou, X. Chen, U. Chang, J.-T. Lu, C. C. Y. Leung, Y. Chen, Y. Hu, and Z. Wang, "A soft-robotic approach to anthropomorphic robotic hand dexterity," *IEEE Access*, vol. 7, pp. 101 483–101 495, 2019.

[4] T. George Thuruthel, Y. Ansari, E. Falotico, and C. Laschi, "Control strategies for soft robotic manipulators: A survey," *Soft Robotics*, vol. 5, no. 2, pp. 149–163, 2018, pMID: 29297756. [Online]. Available: https://doi.org/10.1089/soro.2017.0007

[5] J. Fathi, T. J. Oude Vrielink, M. S. Runciman, and G. P. Mylonas, "A deployable soft robotic arm with stiffness modulation for assistive living applications," in *International Conference on Robotics and Automation (ICRA)*, 2019, pp. 1479–1485.

[6] M. Cianchetti, M. Calisti, L. Margheri, M. Kuba, and C. Laschi, "Bioinspired locomotion and grasping in water: the soft eight-arm OCTOPUS robot," *Bioinspiration & Biomimetics*, vol. 10, no. 3, p. 035003, may 2015. [Online]. Available: https://doi.org/10.1088/1748-3190/10/3/035003

[7] J. Z. Ge, A. A. Calderón, L. Chang, and N. O. Pérez-Arancibia, "An earthworm-inspired friction-controlled soft robot capable of bidirectional locomotion," *Bioinspiration and Biomimetics*, vol. 14, no. 3, p. 036004, feb 2019. [Online]. Available: https://doi.org/10.1088/1748-3190/aae7bb

[8] J. Zou, Y. Lin, C. Ji, and H. Yang, "A reconfigurable omnidirectional soft robot based on caterpillar locomotion," *Soft Robotics*, vol. 5, no. 2, pp. 164–174, 2018, pMID: 29297768. [Online]. Available: https://doi.org/10.1089/soro.2017.0008

[9] 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, pp. 1–24, 02 2021. [Online]. Available: https://doi.org/10.1371/journal.pone.0246102

[10] 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.

[11] M. T. Gillespie, C. M. Best, E. C. Townsend, D. Wingate, and M. D. Killpack, "Learning nonlinear dynamic models of soft robots for model predictive control with neural networks," in *IEEE International Conference on Soft Robotics (RoboSoft)*, 2018, pp. 39–45.

[12] A. Centurelli, L. Arleo, A. Rizzo, S. Tolu, C. Laschi, and E. Falotico, "Closed-loop dynamic control of a soft manipulator using deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4741–4748, 2022.

[13] A. Tariverdi, V. Kalpathy Venkiteswaran, M. Richter, O. Elle, J. Tørresen, K. Mathiassen, S. Misra, and Ø. Martinsen, "A recurrent neural-network-based real-time dynamic model for soft continuum manipulators," *Frontiers in Robotics and AI*, vol. 8, p. 631303, 03 2021.

[14] T. Lesort, V. Lomonaco, A. Stoian, D. Maltoni, D. Filliat, and N. Díaz-Rodríguez, "Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges," *Information Fusion*, vol. 58, pp. 52 – 68, 2020.

[15] F. Piqué, H. T. Kalidindi, L. Fruzzetti, C. Laschi, A. Menciassi, and E. Falotico, "Controlling soft robotic arms using continual learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5469–5476, 2022.

[16] S. Aoi, P. Manoonpong, Y. Ambe, F. Matsuno, and F. Wörgötter, "Adaptive control strategies for interlimb coordination in legged robots: A review," *Frontiers in Neurorobotics*, vol. 11, p. 39, 2017.

[17] R. Szadkowski and J. Faigl, "Neurodynamic sensory-motor phase binding for multi-legged walking robots," in *International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1–8.

[18] P. Oikonomou, M. Khamassi, and C. S. Tzafestas, "Periodic movement learning in a soft-robotic arm," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 4586–4592.

[19] M. I. Jordan and D. E. Rumelhart, "Forward models: Supervised learning with a distal teacher," *Cognitive Science*, vol. 16, no. 3, pp. 307–354, 1992.

[20] J. M. Bern, Y. Schnider, P. Banzet, N. Kumar, and S. Coros, "Soft robot control with a learned differentiable model," in *IEEE International Conference on Soft Robotics (RoboSoft)*, 2020, pp. 417–423.

[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, pp. 124–134, 2019.

[22] H. Imamizu, T. Kuroda, T. Yoshioka, and M. Kawato, "Functional magnetic resonance imaging examination of two modular architectures for switching multiple internal models," *Journal of Neuroscience*, vol. 24, no. 5, pp. 1173–1181, 2004.

[23] D. Wolpert and M. Kawato, "Multiple paired forward and inverse models for motor control," *Neural Networks*, vol. 11, no. 7, pp. 1317–1329, 1998.

[24] K. Doya, K. Samejima, K.-i. Katagiri, and M. Kawato, "Multiple Model-Based Reinforcement Learning," *Neural Computation*, vol. 14, no. 6, pp. 1347–1369, 06 2002.

[25] V. Vikas, E. Cohen, R. Grassi, C. Sözer, and B. Trimmer, "Design and locomotion control of a soft robot using friction manipulation and motor–tendon actuation," *IEEE Transactions on Robotics*, vol. 32, no. 4, pp. 949–959, 2016.

[26] N. Zlatintsi, A. Dometios, N. Kardaris, I. Rodomagoulakis, P. Koutras, X. S. Papageorgiou, P. Maragos, C. Tzafestas, P. Vartholomeos, K. Hauer, C. Werner, R. Annicchiarico, M. Lombardi, F. Adriano, T. Asfour, A. Sabatini, C. Laschi, M. Cianchetti, A. Güler, and R. Lopez-Tarazon, "I-support: A robotic platform of an assistive bathing robot for the elderly population," *Robotics and Autonomous Systems*, vol. 126, p. 103451, 2020.