

## CONTROL OF PNEUMATIC CYLINDERS USING ITERATIVE LINEAR QUADRATIC REGULATOR WITH DEEP LOCAL LINEAR DYNAMICS FOR EXPLOSIVE MOTIONS

YUKI NAKAMURA<sup>a</sup>, IZUMI KARINO<sup>b</sup>, SHOTARO MORI<sup>b</sup>,

KAZUTOSHI TANAKA<sup>b</sup>, SATOSHI NISHIKAWA<sup>b</sup>, RYUMA NIIYAMA<sup>b</sup>, and YASUO KUNIYOSHI<sup>b</sup>

<sup>a</sup>*Graduate School of Interdisciplinary Information Studies, The University of Tokyo, Eng. Bldg.2, 7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan.*

<sup>b</sup>*Graduate School of Information Science and Technology, The University of Tokyo, Eng. Bldg.2, 7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan.*

*E-mail: {y-nakamura, karino, s-mori, tanaka, nisikawa, niiyama, kuniyosh} @isi.imi.i.u-tokyo.ac.jp*

A pneumatic actuator is an attractive option for creating explosive robotic motions, such as jumping and running, because of its high power-to-weight ratio and compliance. However, control of pneumatic actuators suffers from the nonlinear dynamics of air and system identification problems. We propose a method combining deep-learning of local linear dynamics of the system and an iterative linear quadratic regulator (iLQR) for explosive motions by pneumatic robots. To verify the performance of the method, we performed a simple task to control the position and velocity of a one-degree-of-freedom slider driven by a push-pull pneumatic cylinder. The results show that the slider reached a target position with less than 1.4% error (3.5 mm against a 250 mm range of motion) with a task period of 0.5 s. The velocity was 1.09 m/s (starting from a negative position) and 0.551 m/s (starting from a positive position) against the target velocity of 1.00 m/s.

### 1. Introduction

Locomotion in animals is dynamic covering a broad variety of demands, particularly when the animal is either fleeing from predators or is chasing prey in the wildness to survive. For example, animals can jump onto target locations, jump over various gaps, and run on irregular terrain. Dynamic locomotion is also useful for mobile robots to expand their locomotive range. As dynamic locomotion by nature is explosive, we include motions such as throwing and hitting in the following discussion. For explosive motion, it is important to reach a specific state that includes position and velocity at a desired time. For example, the end-effector must be a specific state at the time when a ball is being hit.

A pneumatic actuator is suitable for giving robots the ability to perform dynamic motions because of its high power-to-weight ratio and compliance. Explosive motions, such as jumping,<sup>1,2</sup> jump-and-hit,<sup>3</sup> running,<sup>4,5</sup> and swinging a badminton racquet<sup>6</sup> have been achieved by pneumatic robots using heuristic feed-forward commands. These robots used a heuristic method because air actuation has a delayed response because of its compressibility; mathematical modeling is therefore difficult due to the non-linear and complex nature of the dynamics. However, designing commands manually for various motions is time-consuming and it produces a limited number of motion sets.

Ball throwing was achieved using an iterative linear quadratic regulator (iLQR) with a mathematical model.<sup>7</sup> The iLQR<sup>8</sup> provides optimal control with a small computational cost. Optimal control is effective for explosive motions of pneumatically actuated robots because it can use nonlinear dynamics or make transitions without feedback control. However, precise control depends on modeling accuracy making it difficult to improve mobility control in pneumatic systems because of the complexity and non-linearity. Moreover, when applied to a robot with many degrees of freedom, the calculation of complex forward dynamics might be a problem because explosive motions need fast calculations.

Learnt pneumatic dynamics is promising for solving those problems. The position control of a soft robot arm with a pair of antagonistic pneumatic actuators was achieved using model predictive control and dynamics learned with a deep neural network.<sup>9</sup> Position tracking by pneumatic actuators was achieved using non-linear model predictive control with an echo state Gaussian process,<sup>10</sup> which is a type of recurrent neural network useful for modeling dynamic systems. Dexterous manipulation with a pneumatically actuated 24-degree-of-freedom (DoF) tendon-driven hand was achieved using a method that optimizes trajectories using an iteratively refitted time-varying linear model.<sup>11</sup> These cited papers describe optimal control with learnt dynamics of a pneumatic system for non-explosive motion. However, to the best of our knowledge, there have been no studies applying it to explosive motions.

In this paper, we propose a method for a pneumatic control system using iLQR with deep local linear dynamics for explosive motion. After we explored the appropriate architecture of the dynamics network, we examined the performance of the method by a single pneumatic cylinder.

## 2. Method

### 2.1. Iterative Linear Quadratic Regulator (iLQR)

Nonlinear optimal control for discrete time steps is a process that calculates an input time-series  $u_{0:T-1}$  by minimizing the cost sum from time 0 to  $T$ , written as

$$J(x_0, u_{0:T-1}) = \sum_{t=0}^{T-1} l(x_t, u_t) + l^f(x_T), \quad (1)$$

when defining the running cost function  $l(x_t, u_t)$  at each time step  $t$  and final cost function  $l^f(x_T)$  at final time  $T$  in the system  $x_{t+1} = f(x_t, u_t)$ . Here,  $x_t$  is the state and  $u_t$  is the input. Using the final cost, this method can deal with the state at a specific time that is often important in explosive motions, such as hitting and throwing objects. The iLQR method solves this problem by linearly approximating the system locally at each time as  $x_{t+1} = Ax_t + Bu_t + o$ , optimizing it once, and iterating it several times. The calculation time of this method is shorter than other nonlinear optimal control methods, such as differential dynamic programming or the conjugate gradient method because of the linearization.

Previous studies<sup>7,8</sup> set  $A = f_x(x, u)$ ,  $B = f_u(x, u)$ , and  $o = O$  using the Jacobian determinant in a mathematical model, where  $O$  is the zero matrix. In this study, we applied iLQR to pneumatic actuators with complicated dynamics using deep local linear dynamics  $A, B, o$ , as described in Sec. 2.2. Our approach potentially contributes to an effective computation by replacing the complex calculation of the Jacobian determinant from a mathematical model with learnt dynamics.

### 2.2. Learning locally linear dynamics

Watter *et al.* used an approach that combines deep learning of locally linear dynamics and iLQR.<sup>12</sup> They constructed locally linear dynamics on the latent compressed state space to deal with high-dimensional image inputs. However, designing the cost function of iLQR in the latent space is not intuitive. Also, state compression is not necessary if the state dimension is small. Therefore, we estimated locally linear dynamics in the raw state space. We denote sensor inputs at step  $t$  as  $s_t$  and control inputs at step  $t$  as  $u_t$ . We also use  $v_{t-n:t} = (v_{t-n}^T, v_{t-n+1}^T, \dots, v_t^T)^T$  to denote a concatenated vector. We define the system as containing the state  $x_t = s_{t-d:t}$  and control inputs  $u_t$ . We estimate the locally linear dynamics of the system  $A(s_{t-i:t}, u_{t-i:t})$ ,  $B(s_{t-i:t}, u_{t-i:t})$ , and  $o(s_{t-i:t}, u_{t-i:t})$  from the previous  $i$  ( $\geq d$ ) step inputs  $s_{t-i:t}$  and  $u_{t-i:t}$ . We estimate the next states  $\hat{x}_{t+1}$  by applying these dynamics. We

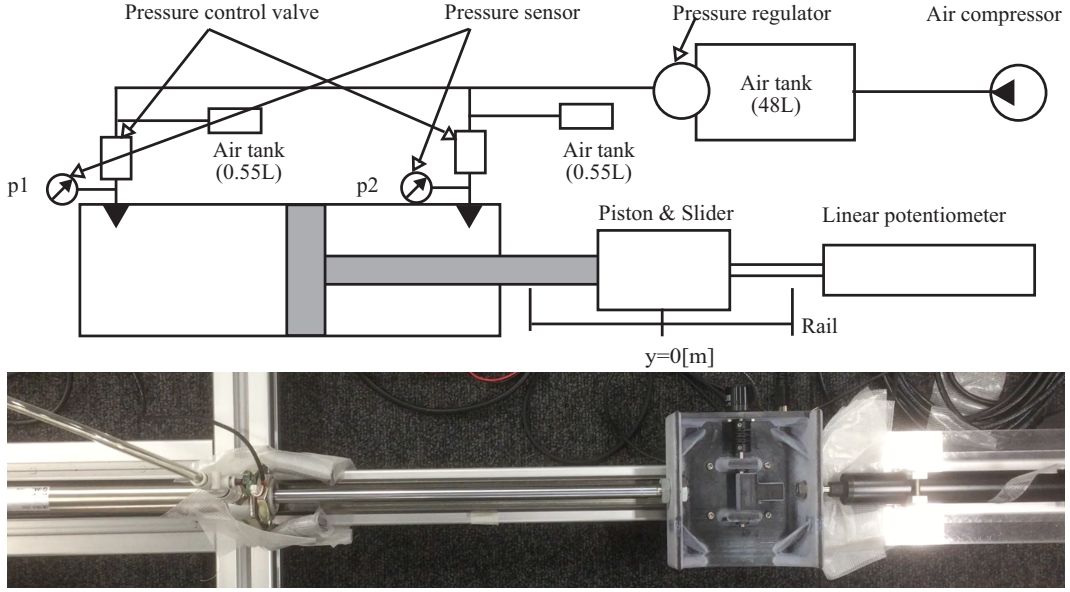


Fig. 1. Schematic diagram (top) and photo (bottom) of the experimental setup.

repeat the same prediction  $p$  times with the predicted state as the next input. Finally, we obtain the estimations  $\hat{x}_{t+1:t+1+p}$ . We optimize the locally linear dynamics estimators  $A$ ,  $B$ , and  $o$  through supervised learning by minimizing the prediction error for the  $p$  steps,  $L = \frac{1}{|\mathcal{D}|} \sum_{(s_{t-i:t}, u_{t-i:t}, s_{t+1:t+1+p}) \in \mathcal{D}} \|x_{t+1:t+1+p} - \hat{x}_{t+1:t+1+p}\|_2^2$ , where  $\mathcal{D}$  is a data set and  $|\mathcal{D}|$  denotes the size of the data set. The estimated dynamics  $\hat{x}_{t+1} = Ax_t + Bu_t + o$  is written as

$$\begin{bmatrix} s_{t-d+1:t} \\ \hat{s}_{t+1} \end{bmatrix} = \begin{bmatrix} O & I \\ A' & \end{bmatrix} s_{t-d:t} + \begin{bmatrix} O \\ B' \end{bmatrix} u_t + \begin{bmatrix} O \\ o' \end{bmatrix}, \quad (2)$$

where  $I$  is the identity matrix and  $O$  is the zero matrix. Therefore, we estimate the partial matrices  $A'(s_{t-i:t}, u_{t-i:t})$ ,  $B'(s_{t-i:t}, u_{t-i:t})$ , and  $o'(s_{t-i:t}, u_{t-i:t})$ . We use neural networks to estimate  $A'$ ,  $B'$ , and  $o'$ .

### 3. Experiments

#### 3.1. Setup

We developed a control system with a push-pull pneumatic cylinder and pressure control valves (Fig. 1). We suppose that the control system is a component of a robot with multiple degrees of freedom. The control target is the states of a slider attached to the piston rod of the pneumatic cylinder. The cylinder has a chamber diameter of 25 mm and a stroke of 250 mm (CM2XB25-250Z, SMC Corp.). We controlled the inner pressure of each chamber through pressure control valves (Tecno Basic, Hoerbiger Corp.) with pressure commands  $u = (u_1, u_2)$ .

The maximum supply pressure was set to 0.3 MPa. The inner pressures  $p_1$  and  $p_2$  of the chambers were measured with pressure sensors (XFGM-6001MPGSR, Fujikura, Ltd.). The position  $y$  of the slider was measured with a linear potentiometer (LP-250FJ-5K, Midori Precisions Co., Ltd.). We used the inner pressures and the position as sensor data  $s = (p_1, p_2, y)$ .

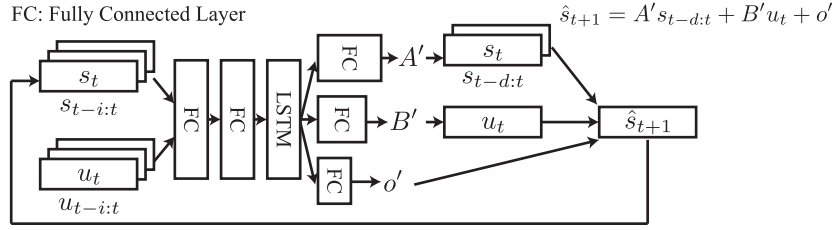


Fig. 2. Architecture of the prediction network. We denote sensor and control inputs at step  $t$  as  $s_t$  and  $u_t$ , respectively. We also use  $v_{t-n:t} = (v_{t-n}^T, v_{t-n+1}^T, \dots, v_t^T)^T$  to denote a concatenated vector. We define the system as containing the state  $x_t = s_{t-d:t}$  and control inputs  $u_t$ . We estimate the locally linear dynamics of the system  $A(s_{t-i:t}, u_{t-i:t})$ ,  $B(s_{t-i:t}, u_{t-i:t})$ , and  $o(s_{t-i:t}, u_{t-i:t})$  from the previous  $i$  ( $\geq d$ ) step inputs  $s_{t-i:t}$  and  $u_{t-i:t}$ . We only parametrized their partial matrices  $A'(s_{t-i:t}, u_{t-i:t})$ ,  $B'(s_{t-i:t}, u_{t-i:t})$ , and  $o'(s_{t-i:t}, u_{t-i:t})$  (see Eq. (2)).

### 3.2. Collecting data for learning locally linear dynamics

In learning the dynamics, we collected data from 1,000 trials. Each trial consisted of the following process: start at a random position, fit the slider to the starting position with proportional control, and collect data with random control inputs. Each trial included 20 control steps and a sampling period set at 50 ms. The maximum control input pressure was 0.3 MPa, and the supply pressure at the pressure regulator attached to the air tank (38 L) was 0.5 MPa.

### 3.3. Network architectures for estimating local linear dynamics

For the network architecture (Fig. 2), we applied fully connected layers to the inputs  $s_{t-i:t}, u_{t-i:t}$  and a succeeding long short-term memory (LSTM) layer as an option. The LSTM is a recurrent neural network. The sensor  $s_t$  and control  $u_t$  were normalized to  $[0, 1]$ . All of the layers used layer normalization.<sup>13</sup> A fully connected layer followed each of these shared hidden layers to calculate  $A'$ ,  $B'$ , and  $o'$ . We investigated several neural network architectures to estimate the system dynamics. We searched for the best neural network architecture by changing the depth of the fully connected layers, the number of units in each layer, the use of LSTM, time series length of inputs  $i$ , and the time series length of sensors  $d$  contained in the system's state. A comparison is shown in Fig. 3.

The number of prediction steps  $p$  was 20. We used 800 training datasets and 200 validation datasets. Each batch contained all the training data; the number of epochs was 50,000. We applied the Adam optimizer<sup>14</sup> with gradient clipping to restrict the norm of the gradient to at most 5.0. The learning rate decayed from  $10^{-2}$  to  $10^{-12}$  with a decay rate of 0.96.

The best performance had a 2.5% error ratio in the sensor range per element on average, where the network had a hidden layer of 32 units, a succeeding LSTM layer with 32 units, input length  $i = 1$ , and  $d = 1$  (Fig. 3). We found that the use of LSTM was the most effective to improve performance. Figure 4 compares the per-element prediction error between the best architecture with LSTM and the best architecture without LSTM. The network with LSTM had fewer errors than did the network without LSTM, except for the prediction error of the position  $y$  at the 20th step. Prediction errors tended to increase as the prediction steps increased.

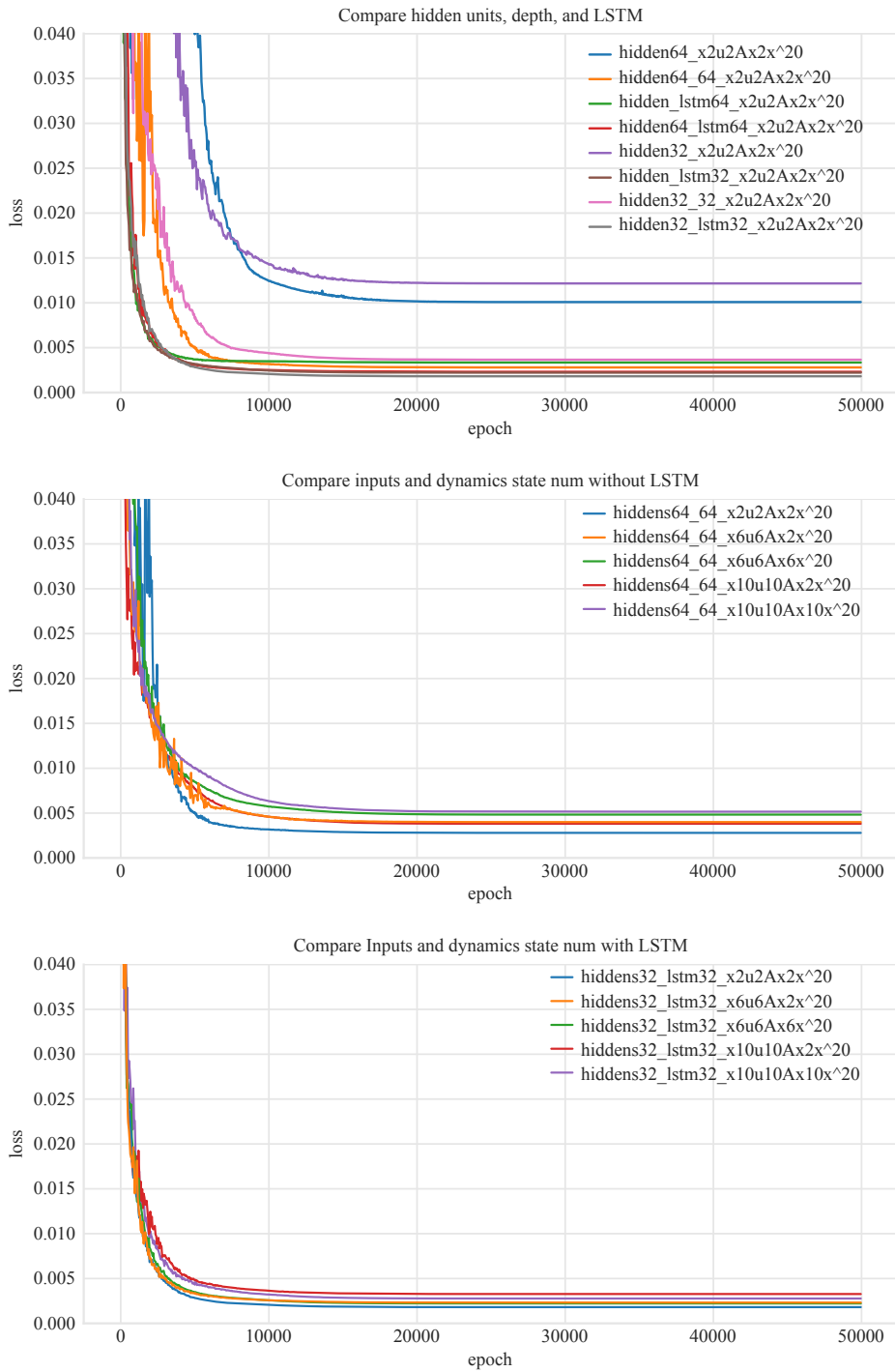


Fig. 3. Loss profile of the LSTM with different parameters. Each label hidden ( $h$ )-lstm( $l$ )-x( $i$ )u( $i$ )Ax( $d$ )x( $\hat{p}$ ) means an instance with hidden units  $h$ , the number of units of the succeeding LSTM layer  $l$ , the number of inputs to calculate locally linear dynamics  $i$ , the dimension of the system state  $d$ , and the prediction length  $p$ .

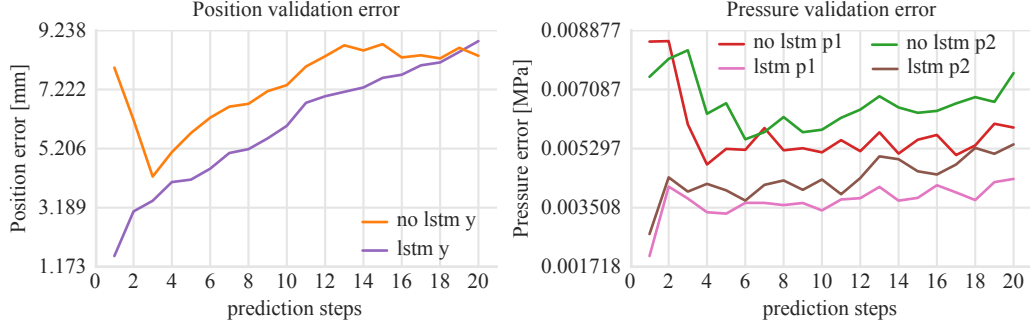


Fig. 4. Prediction errors of the learnt dynamics with different neural networks architectures.

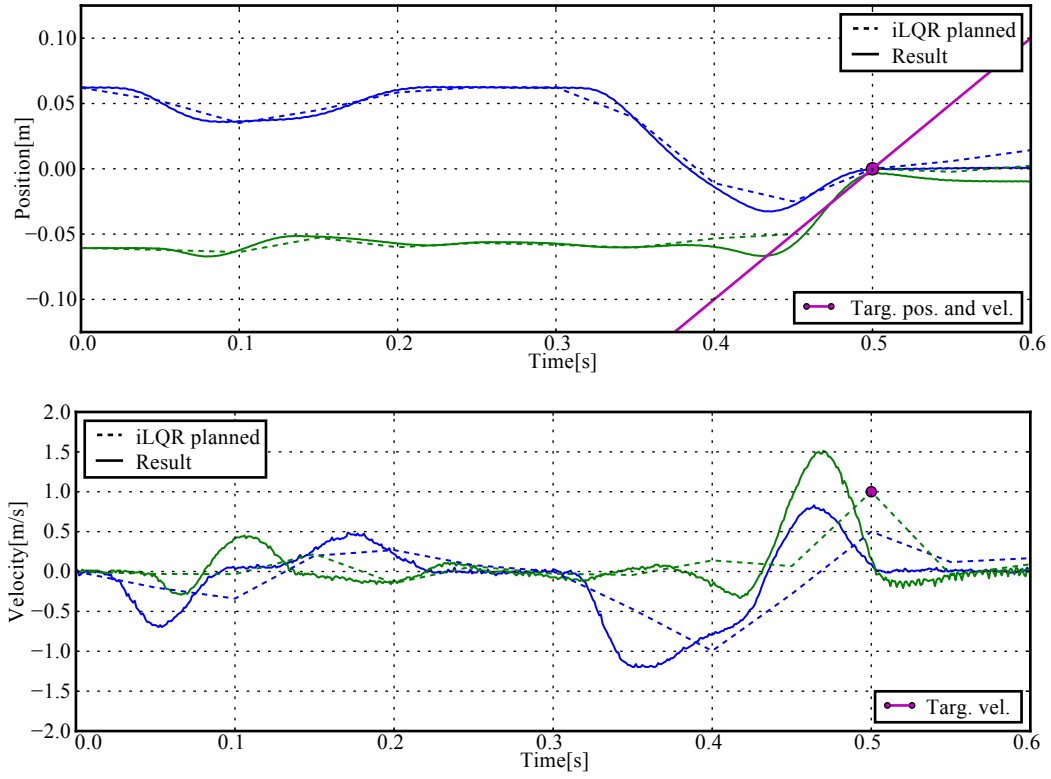


Fig. 5. Trajectories (top) and velocity trajectories (bottom) of motion generated in a pneumatic cylinder from different initial positions. The dot indicates the target position, and the slope of the line through the dot represents the target velocity (top). The dot indicates the target velocity (bottom). The blue curve starts from a positive position (0.075 m), the green curve starts from a negative position (-0.075 m).

### 3.4. Explosive motions using iLQR

We controlled the movements of a pneumatic cylinder by iLQR using an LSTM network with best prediction performance. We estimated the system described in Eq. (2) with  $i = 1$  and  $d = 1$ , that is,

$$x_{t+1} = \begin{bmatrix} O & I \\ A' & \end{bmatrix} x_t + \begin{bmatrix} O \\ B' \end{bmatrix} u_t + \begin{bmatrix} O \\ o' \end{bmatrix}. \quad (3)$$

The running cost  $l(x_t, u_t)$  and final cost  $l^f(x_T)$  were defined as

$$l(x_t, u_t) = \mu_u^T B(u_t, \bar{u}, \underline{u}) + \mu_x^T B(x_t, \bar{x}, \underline{x}) \quad (4)$$

$$l^f(x_T) = x_T^T R x_T, \quad (5)$$

where  $B(x, \bar{x}, \underline{x}) = -(\ln(\bar{x} - x) + \ln(x - \underline{x}))$  is the log barrier function,  $\mu_u$  and  $\mu_x$  are weight vectors, and  $R$  is a final weight matrix. We set  $R$  to include the weight of the position  $y_t$  and velocity  $(y_t - y_{t-1})/dt$ . As such, we aimed to reach the target position and velocity at the final time step  $T = 10$  (0.5 s) with the constraint

$$\underline{x} < x < \bar{x} \quad (6)$$

$$\underline{u} < u < \bar{u}. \quad (7)$$

We calculated iLQR several times while gradually decreasing the weights  $\mu_u$  and  $\mu_x$ . The optimization result then approached the solution of the constraint problems to minimize Eq. (5) with Eqs. (6) and (7) as the weights approached zero, where  $R = \text{diag}([0, 0, 1, 0, 0, 100])$ .

The experimental result is shown in Fig. 5. Planning with iLQR generated control inputs that successfully achieved the target state. As a result, the slider achieved a target position with less than 1.4% error (3.5 mm against a 250 mm range of motion). The target velocity, 1.00 m/s, was an average velocity calculated from the difference between the positions at 0.45 s and 0.50 s, because the control period was 0.05 s. The results for the average velocity between 0.45 s and 0.50 s were 1.09 m/s (green curve) and 0.551 m/s (blue curve). However, the results for the instantaneous velocity at 0.50 s were 0.220 m/s (green curve) and 0.0674 m/s (blue curve).

#### 4. Discussion

In this experiment, the slider reached the target position at the desired time. To achieve this task, we have to predict precisely the dynamics of the pneumatic system, which has complex dependency through time. The top of Fig. 3 shows that the neural network with LSTM can learn more quickly and predict more precisely than the neural network without LSTM. These results suggest that capturing the complex structure of the pneumatic system is more effective with LSTM.

However, there are two problems. It was more difficult to achieve both position and velocity than either property separately in the optimization process. We consider a start from a positive position (blue curve in Fig. 5) a more difficult condition than a start from a negative position (green curve) because of the need to move once in a direction opposite to that of the target velocity. The result of the optimization was affected by the initial control input series. Therefore, it is important to determine the initial control series. The other problem is the low time resolution for the instantaneous velocity, which creates a time gap for the velocity. It is more important for explosive motions to control the instantaneous velocity than the average velocity. To deal with the instantaneous velocity, there are two possible methods. One is to shorten the control period, although predicting the target state may be made more difficult as the steps in the prediction increase. The other is to include the instantaneous velocity in the predicted state, in contrast to calculating the velocity by the difference between the positions in this experiment.

#### 5. Conclusion and Future Work

To overcome the difficulty in controlling the pneumatic actuator, we proposed a method to combine deep learning of the local linear dynamics and iterative linear quadratic regulator

(iLQR) for control of the explosive motions for pneumatic robots. We controlled a pneumatic cylinder to reach a target position and velocity at the desired time. The position error is less than 1.4% error (3.5 mm against a 250 mm range of motion). The target velocity, the average velocity calculated for each control period (0.05 s), was 1.00 m/s. The results for the average velocity between 0.45 s and 0.50 s were 1.09 m/s (starting from a negative position, the case easy to optimize) and 0.551 m/s (starting from a positive position, the case difficult to optimize).

This is a first step in motion generation of a pneumatic robot in explosive motions for arbitrary targets, including dynamic locomotion. Future work will include an implementation of feedback control using real-time calculations and applications of this method to pneumatic robots with multiple degrees of freedom.

### Acknowledgments

This work was supported by the Japan Society for the Promotion of Science, KAKENHI Grants JP17H06575, JP18K18087, and JP18H05466, and the Kayamori Foundation of Informational Science Advancement.

### References

1. R. Niiyama, A. Nagakubo and Y. Kuniyoshi, Mowgli: A bipedal jumping and landing robot with an artificial musculoskeletal system, in *IEEE International Conference on Robotics and Automation (ICRA)*, 2007.
2. K. Hosoda, Y. Sakaguchi, H. Takayama and T. Takuma, *Autonomous Robots* **28**, 307 (2009).
3. K. Tanaka, S. Nishikawa, R. Niiyama and Y. Kuniyoshi, Humanoid robot performing jump-and-hit motions using structure-integrated pneumatic cable cylinders, in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2017.
4. R. Niiyama, S. Nishikawa and Y. Kuniyoshi, *Advanced Robotics* **26**, 383 (2012).
5. K. Narioka, A. Rosendo, A. Sproewitz and K. Hosoda, Development of a minimalistic pneumatic quadruped robot for fast locomotion, in *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2012.
6. S. Mori, K. Tanaka, S. Nishikawa, R. Niiyama and Y. Kuniyoshi, *IEEE Robotics and Automation Letters (RA-L)* **3**, 1727 (2018).
7. G. K. H. S. L. Das, B. Tondu, F. Forget, J. Manhes, O. Stasse and P. Soueres, Performing Explosive motion using a multi-joint arm actuated by pneumatic muscles with quasi-DDP optimal control, in *IEEE Conference on Control Applications (CCA)*, 2016.
8. W. Li and E. Todorov, Iterative linear quadratic regulator design for nonlinear biological movement systems, in *International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, 2004.
9. 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.
10. J. Huang, Y. Cao, C. Xiong and H. T. Zhang, *IEEE Transactions on Automation Science and Engineering*, 1 (2018).
11. V. Kumar, E. Todorov and S. Levine, Optimal control with learned local models: Application to dexterous manipulation, in *IEEE International Conference on Robotics and Automation*, 2016.
12. M. Watter, J. Springenberg, J. Boedecker and M. Riedmiller, Embed to control: A locally linear latent dynamics model for control from raw images, in *Neural Information Processing Systems conference (NeuralIPS)*, 2015.
13. J. L. Ba, J. R. Kiros and G. E. Hinton, *arXiv preprint arXiv:1607.06450* (2016).
14. D. P. Kingma and J. Ba, *arXiv preprint arXiv:1412.6980* (2014).