2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Application of Game-theoretic Learning to Gray-box Modeling of McKibben Pneumatic Artificial Muscle Systems

Kiminao Kogiso, Ryo Naito and Kenji Sugimoto

***Abstract*— We consider a gray-box modeling of a McKibben pneumatic artificial muscle (PAM) actuated by a proportional directional control valve. This paper presents a hybrid nonlinear model of the PAM system and then proposes a systematic parameter identification procedure that uses a game-theoretic learning algorithm to obtain the appropriate parameter values for the modeling. With a practical example, finally, we verify the proposed method by illustrating a process of searching for the parameter values together with figures of after-and-before learning. As a result, we see that the resulting parameters are better than ones obtained by our previously-proposed heuristic and trial-and-error-based algorithm.**

I. INTRODUCTION

The McKibben pneumatic artificial muscle has a high strength-to-weight ratio and good flexibility due to its mechanical structure. It consists of an internal rubber tube surrounded by a cylindrical mesh that is braided by inextensible threads. Both ends of the two-layered tube are sealed by caps to retain the cylindrical form, and one cap has a connecter to supply compressed air. By filling the inner tube with compressed air, the diameter of the rubber tube increases and the long axis shortens due to the inextensible threads. This is how the PAM generates a contraction force. On the other hand, releasing the compressed air from the PAM allows the elasticity of the rubber tube inside the PAM to return it to its original shape.

Modeling and control of the PAM are known to be one of the challenging issues in a robotics field. There are several studies and applications about modeling and control, such as nonlinear modeling [1]–[3], advanced control [2], [4]–[10] and model-based control of robots with PAM [11], [12]. Especially, most of the difficulties in the issues are caused by the nonlinear characteristics such as a hysteresis and compressed air flow, and so catching them has some merits, such as realization of fine controls, evaluation of how much the true dynamics is approximated, and etc. Making it easy to handle in terms of control, Vo-Minh *et al.* [13] treat with a Maxwell-slip model as a lumped-parametric quasi-static model in order to capture the force/length hysteresis. Motivated to acquire an accurate nonlinear hybrid model, our initial work [14] employs a friction model based on implicit Euler integration [15] and a fluid model of the proportional directional control valve, whose parameters can be identified by a heuristic but rational procedure proposed in [16]. Moreover, in [16], the authors have considered a situation where the mass of the weight continuously changes within an allowable weight range, taking into account future applications to the antagonistic PAM pairs [17], [18] and further development. The proposed algorithm, however, is a heuristically trial-and-error-based procedure for identifying the parameters so that it takes much computation times and experiences to obtain the appropriate values. Therefore, automation of the identification procedure must be useful for people those who want to get the PAM models, like the work [19] that proposes a gray-box identification with a locally-linearlized model for an industrial robot.

This paper aims at a gray-box modeling of the PAM system actuated by the proportional directional control valve. The gray-box modeling requires a mathematical model of the PAM system and its parameter identification process. In this paper, first, we present a hybrid nonlinear model of the PAM system that is based on our previous result in modeling [16] with modification of generalization in terms of the load. Analysis of the model reveals that it has nine key parameters characterizing transient and steady-state behaviors and also that some of the model parameters dependent on the load are ones affecting on only steady-state behaviors, where the steady state is considered when a step signal is input to the system as a control command to the valve. Next, this paper proposes a systematic method of automatically identifying the parameters by applying a game-theoretical learning technique, which is a main contribution. The learning algorithm searches for values of the parameters so that simulation data, generated by the model, is as close as possible to experimental data, sampled in advance. This can be achieved by measuring an error between the two data in terms of a utility. This paper describes how to materialize the utility in a procedure form since the materialization is a key to the achievement in our game-theoretic manner. Finally, to verify the gray-box modeling including the automatic identification, we show a process of searching for the parameter values together with figures of after-and-before learning, and the resulting parameters, which are better than ones obtained by the existing perfectly trial-and-error-based algorithm [16].

An essential issue of this work is that the determination of the parameters is to solve a complex problem of a simulation-based nonlinear optimization subject to a hybrid nonlinear constraint condition. The contribution of our work, thereby, can be also explained by that an adequate solver for the problem is developed by the game-theoretic learning algorithm. What the solver does is in a game fashion as follows. The parameter (that is a player) tries to maximize the utility by observing the opponent parameters' updates. Repeating this game play, the parameters can reach a so-called Nash

The authors is with Graduate School of Information Science, Nara Institute of Science and Technology, 8916-5 Takayamacho, Nara, Ikoma 6300192, Japan kogiso@is.naist.jp or kogiso@ieee.org

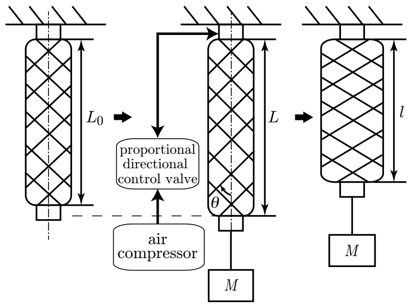


Fig. 1. Illustration of considered PAM system.

equilibrium where neither players want to deviate from the respective utility values. The resulting parameters are a local (or might be a global) optimum to the optimization problem and then the model of the PAM system can be automatically identified.

II. Physical Modeling of the PAM system

This paper considers a PAM system with a vertically suspended weight and a proportional directional control valve, as illustrated in Fig. 1. A model of the considered PAM system is in a switched system with 32 nonlinear subsystems $f_\sigma : \Re^3 \times \Re \rightarrow \Re^3$,

$$\dot{x}(t) = f_\sigma(x(t), u(t)) \text{ if } x(t) \in \mathcal{X}_\sigma, \tag{1a}$$

$$y(t) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} x(t), \tag{1b}$$

where the state variable $x \in \Re^3$ and output variable (measured by sensors) $y \in \Re^2$ are respectively defined as $x := [\epsilon \ \dot{\epsilon} \ P]^T$ and $y := [\epsilon \ P]^T$, $u \in \mathcal{U} \subset \Re$ is a control iput, $\sigma \in \Sigma := \{1, 2, \cdots, 32\}$ is a subsystem's index, $\mathcal{X}_\sigma := \{x \in \Re^3 \mid \Psi_\sigma(x) \leq 0\}$, $\Psi_\sigma : \Re^3 \times \Re \rightarrow \Re$ is a function derived from a conditional statement in the if-then rule. When inputting a constant command signal to the valve, $\bar{u} \in \mathcal{U}$, there exists a unique index, $\sigma \in \Sigma$, such that $f_\sigma([\bar{\epsilon} \ 0 \ \bar{P}]^T, \bar{u}) = 0$ is satisfied, where $\bar{\epsilon}$ and $\bar{P}$ are a contraction ratio and an inner pressure of the PAM in steady state, respectively.

The model (1) is written by summarizing several dynamic equations and is based on our results in modeling [14], [16], where some of the equations are partially modified to have dominant parameters take a positive value. The dynamic equations are related to a contraction force, some frictions, a PAM's volume, an inner pressure, a weight, and a flow rate of a control valve, all of which are introduced in Appendix. When considering the load that varies in time, we have to reveal which parameters depend on the load or not. In [16], fortunately, we have specified such parameters via doing several experiments for parameter identification, and then this paper characterizes relation of the parameters with the load in the following proposition.

Proposition 1: For the model (1), there exists an interval $[a, b]$ $(0 \leq a \leq b)$ such that parameters, K, θ, C_{q1}, C_{q2}, and c_c, are dependent on a load, $M \in [a, b]$, and other parameters, A_0, k_1, k_2, and c_v, are constant over all M.

Naturally, the load-dependent parameters can be considered functions of the load whose domain is $[a, b]$, i.e., $K(M)$, $\theta(M)$, $C_{q1}(M)$, $C_{q2}(M)$, and $c_c(M)$ defined over $M \in [a, b]$. Note, the PAM model shown in the appendix is already generalized in terms of M. The feature of *Proposition 1* can lead to the following corollary by the similar technique used to derive the main *Theorem 1* in [16], where its proof is omitted here so please see that literature.

TABLE I
Variables and parameters of PAM systems

D_0	: natural diameter of PAM [m]
L_0	: natural length of PAM [m]
L	: initial length of PAM after a weight loaded [m]
l	: length of PAM [m]
$\epsilon = \frac{L-l}{L}$	: contraction ratio [-]
V	: volume of PAM $[\mathrm{m}^3]$
D_1, D_2, D_3	: coefficients of polynomial $[\mathrm{m}^3]$
M	: mass of the weight [Kg]
g	: gravitational acceleration $[\mathrm{m/s^2}]$
P	: absolute internal gas pressure [Pa]
P_{tank}	: source absolute pressure [Pa]
P_{out}	: atmospheric pressure [Pa]
C_v	: specific heat at constant volume [J/Kg K]
k	: specific heat ratio for air [-]
R	: ideal gas constant [J/Kg K]
T	: absolute temperature [K]
K	: coefficient of elasticity $[\mathrm{N/m^3}]$
θ	: initial angle between a braided thread and cylinder long axis [deg]
C_{q_1}, C_{q_2}	: correction coefficients [-], [1/Pa]
c_c	: Coulomb friction [N]
A_0	: orifice area of control valve $[\mathrm{m}^2]$
k_1, k_2	: polytropic indexes $[-]$
c_v	: viscous friction coefficient [Ns/m]

Corollaly 1: (*Separability of Parameters*) The parameters of the model (1) can be divided into two groups: the parameters, $K(M), \theta(M), C_{q1}(M), C_{q2}(M), c_c(M)$ for any $M \in [a, b]$, characterize steady-state behaviors, and the other parameters, A_0, k_1, k_2, c_v, characterize transient behaviors.

The separability of the parameters is very interesting because although it looks like the formulated model (1) is too complex to investigate and catch what impact the parameters' change respectively gives on the transient and steady-state behaviors, the corollary says not true. This result gives an indication of making an effective procedure of identifying the nine parameters, which is our aim of this work.

III. Parameter Identification using Game-Theoretic Learning

What we want to do in this section is to solve the following optimization problem for the parameter identification: given experimental data $\mathbb{D}_{exp}$,

$$\begin{aligned} &\min_{p \in \Re^9} d(\mathbb{D}_{sim}(p), \mathbb{D}_{exp}) \\ &\quad \text{s.t. PAM model (1) generates } \mathbb{D}_{sim}(p), \\ &\quad\quad p_i > 0 \ \ \forall i \in \{1, 2, \cdots, 9\}, \end{aligned}$$

where $p = (K, \theta, C_{q1}, C_{q2}, c_c, A_0, k_1, k_2, c_v) \in \Re^9$ is a decision variable and $d(\cdot, \cdot)$ is a distance between two data sets. This optimization has a nonlinear objective and a hybrid nonlinear constraint condition, which would look too difficult to solve by existing algorithms, so that we try to develop a game-theoretic learning algorithm suitable for obtaining the solution (parameter values). In general, finding the global solution to such a nonlinear and complex optimization problem is so hard, while if an accurate measurement of the distance is possible to realize, it must be helpful to evaluate the solutions. For example, the accurate measurement enables to check how far or close the obtained solution is to the optimal or the other local. Therefore, our approach also takes into account materialization of how to accurately measure the distance between two data sets in terms of an area error on a corresponding plane, which is a feature of this approach.

Remark 1: The game-theoretic learning process of searching for the values is to update p with a time-invariant step size Δp, i.e., $p \leftarrow p + \Delta p$. The simulation data $\mathbb{D}_{sim}(p)$ is recalculated with the updated p. Then, we write $\mathbb{D}_{sim}(\Delta p)$ with an initial parameter value $p(0)$ in the rest of this paper, where Δp corresponds to a that is defined in the next section.

A. Game Setting

Let us define the game $\mathcal{G}$ with players, actions, and a utility: A set of rational N players is $\mathcal{P} = \{1, 2, \cdots, N\}$, an action of player i is $a_i \in \mathcal{A}_i := \{0, \Delta p_i\}$, and a utility is $u(a) = -d(\mathbb{D}_{sim}(a), \mathbb{D}_{exp})$, where an action profile is $a = (a_1, a_2, \cdots, a_N)$ and $a \in \mathcal{A} := \Pi_{i \in \mathcal{P}} \mathcal{A}_i$, then the strategic game is denoted as $\mathcal{G}(\mathcal{P}, \mathcal{A}, u)$. The player becomes a parameter that we want to identify. The utility is common for all players, is a non-positive function, $u : \mathcal{A} \to \Re_-$, and represents a scalar value related to an error between two data sets, whose construction and detail explanations will be shown in section III-B. For the i-th player's action $a_i \in \mathcal{A}_i$, its opponent's action profile is denoted as a_{-i}, i.e., $a = (a_i, a_{-i})$. The important concepts are described below, which are helpful for analyzing the game.

Definition 1: A best response of player i to an opponent action profile a_{-i} is denoted as $BR_i(a_{-i})$:

$$BR_i(a_{-i}) = \{a_i \in \mathcal{A}_i \mid u(a_i, a_{-i}) \geq u(a'_i, a_{-i}) \ \forall a'_i \in \mathcal{A}_i\}. \quad (2)$$

Definition 2: An action profile a^* is a Nash equilibrium if, for every $i \in \mathcal{P}$, $a^*_i \in BR_i(a^*_{-i})$ holds.

Our process of searching for the values of the parameters by repeating the game, follows a Cournot adjustment model. Supposed that parameter c_v is ith player, for example, the parameter at step t, denoted as $c_v(t)$, is updated by $c_v(t+1) = c_v(t) + 0$ if $a_i(t) = 0$, or $c_v(t+1) = c_v(t) + \Delta p_i$ otherwise, where Δp_i is a time-invariant step size to the player c_v. Then, the process makes a choice of more reasonable one of the two actions that yields a better utility value, i.e.,

$$\begin{aligned} a_i(t) &= \arg\max_{a_i \in \mathcal{A}_i} (u(a_i, a_{-i})) \\ &= \arg\max(u(0, a_{-i}), u(\Delta p_i, a_{-i})), \end{aligned}$$

where it should be noted that u takes a non-positive value.

This kind of the game setting can be viewed as a game deterministically played by rational players who wants to maximize the utility (the area error multiplied by minus one). That is to say that each strategic player, upon observing the opponent action a_{-i}, would select a reasonable action a_i that would minimize a cost of the error. In an equilibrium, every player has an action a^*_i that maximizes the utility, given the opponent action a^*_{-i}. Therefore, no rational players would find it profitable to unilaterally change its action. This equilibrium is known as to be a *Nash equilibrium* or *Cournot-Nash equilibrium*. Here, the equilibrium is when each player's action is included in a best response to the opponent action, then neither wants to deviate from the respective utility values. In this game-theoretic approach, the player (parameter) learns in a rational way to reach a certain equilibrium. This is why the identification process is called *learning*.

B. Materialization of the Utility

The utility is basically to measure an error between simulation and experimental data in both of steady-state and transient responses. Toward explaining how to measure, first, we introduce some notations for the data. $\mathbb{D}_{sim}(a)$ denotes a set of simulation data that is obtained by running a code of the hybrid nonlinear system with parameters updated by a and with M_i fixed. The simulation data set consists of steady-state data $\mathbb{D}_{sim,s}$ and transient data $\mathbb{D}_{sim,t}$, and furthermore, $\mathbb{D}_{sim,s}$ includes dilatation process data $\mathbb{D}^{dila}_{sim,s}$ and contraction process data $\mathbb{D}^{cont}_{sim,s}$, where the dilatation and the contraction are separated to catch a hysteresis loop appearing in steady state. That is,

$$\begin{aligned} \mathbb{D}_{sim} &= \{\mathbb{D}_{sim,s}, \mathbb{D}_{sim,t}\} = \{\mathbb{D}^{dila}_{sim,s}, \mathbb{D}^{cont}_{sim,s}, \mathbb{D}_{sim,t}\}, \\ &= \Big\{ \{(\bar{P}^d_{si}, \bar{\epsilon}^d_{si})\}^{N^d_s}_{i=1}, \{(\bar{P}^c_{si}, \bar{\epsilon}^c_{si})\}^{N^c_s}_{i=1}, \\ &\qquad \{(t_{si}, P_{si}(t_i), \epsilon_{si}(t_i))\}^{N^t_s}_{i=1} \Big\}, \end{aligned}$$

where N^d_s, N^c_s and N^t_s are the number of the corresponding data. Similarly, the experimental data $\mathbb{D}_{exp,s}$ consisting of $\mathbb{D}_{exp,s}$ and $\mathbb{D}_{exp,t}$ are sampled in advance using an practical equipment, and $\mathbb{D}_{exp,s}$ includes dilatation process $\mathbb{D}^{dila}_{exp,s}$ and contraction process $\mathbb{D}^{cont}_{exp,s}$. That is,

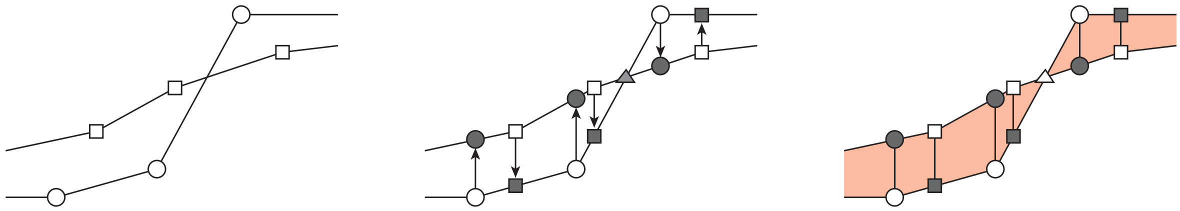
$$\begin{aligned} \mathbb{D}_{exp} &= \{\mathbb{D}_{exp,s}, \mathbb{D}_{exp,t}\} = \{\mathbb{D}^{dila}_{exp,s}, \mathbb{D}^{cont}_{exp,s}, \mathbb{D}_{exp,t}\}, \\ &= \Big\{ \{(\bar{P}^d_{ei}, \bar{\epsilon}^d_{ei})\}^{N^d_s}_{i=1}, \{(\bar{P}^c_{ei}, \bar{\epsilon}^c_{ei})\}^{N^c_e}_{i=1}, \\ &\qquad \{(t_{ei}, P_{ei}(t_i), \epsilon_{ei}(t_i))\}^{N^t_e}_{i=1} \Big\}, \end{aligned}$$

where N^d_e, N^c_e and N^t_e are the number of the corresponding data. Note that the both data sets are separately acquired so that the pressure sequences and the time sequences included are not always common to each other. We have to process the original data sets in an appropriate way to achieve accurate measurement of the error between them.

The utility function:

$$u(a) = -d(\mathbb{D}_{sim,\cdot}(a), \mathbb{D}_{exp,\cdot}) \leq 0,$$

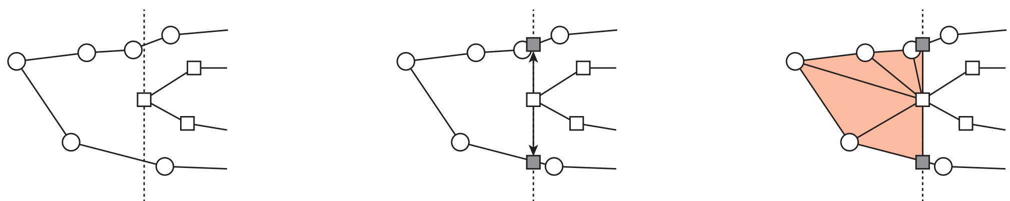
then, can be materialized in a procedure that follows below, where the subscript '$\cdot$' is 's' or 't'. Our idea of it, as illustrated in Fig. 2, is to partition the error area into triangles and trapezoids, and since they are convex sets, we can



(a) Original data of simulation ◯ and experiment □ over the middle range $[\alpha_l,\ \alpha_h]$.

(b) Projections of ◯ and □ onto the other segment, respectively. They are $\Psi(\mathbb{D}_{sim,s})$ and $\Psi(\mathbb{D}_{exp,s})$. If exist, a cross point is calculated, denoted as △, and a set of the cross points is denoted as $\mathbb{D}_{cp}$.

(c) The error over the middle range is a sum of trapezoidal and triangular areas in red, by using the processed data $\{\mathbb{D}_{sim,s}, \Psi(\mathbb{D}_{exp,s}), \mathbb{D}_{cp}\}$ and $\{\mathbb{D}_{exp,s}, \Psi(\mathbb{D}_{sim,s}), \mathbb{D}_{cp}\}$.



(d) Original data of simulation ◯ and experiment □ over the side range $[0,\ \alpha_l]$, where it is a low side range in this figure.

(e) Two projections of the edge point □, whose pressure is α_l, onto the simulation data segments. The points are ready to be denoted as $\mathbb{D}_{\alpha_l}$.

(f) The error over the range $[0, \alpha_l]$ is a sum of triangular areas in red, by using the data $\mathbb{D}_{\alpha_l}$.

Fig. 2. Procedure of calculating the utility $u(a)$, which provides an accurate area error between $\mathcal{D}_{sim}(a)$, and $\mathcal{D}_{exp}$, denoted as ◯ and □ respectively. (a)-(c) are about processing the data over the middle range and (d)-(f) are about processing over the (low) side range in the steady state. In the case of the transient responses, the similar technique of (a)-(c) is used to calculate error areas.

use a standard function such as `polyarea` on MATLAB. Although this idea is fundamental in computer science, some technical and unique situations must be treated adequately. For example, the data $\mathbb{D}_{sim}$ and $\mathbb{D}_{exp}$ include the hysteresis loop coming from the dilatation and contrition processes, so that we need take into account the processes in calculating the error, and the treatment must change depending on a pressure range. Additionally, the procedure is constructed to output a non-positive value and so $u(a)$ taking zero means that it is the happiest situation to each player who makes his/her decision of choosing a_i.

Step 1. To get ready for measuring the error in steady state, the pressure range are separated into three: $[0, \alpha_l]$, $[\alpha_l, \alpha_h]$, and $[\alpha_h, \infty]$, where α_l and α_h are given by

$$\begin{aligned}\alpha_l = \max\Big(&\min_i\left(\{\bar{P}^d_{si}\}_{i=1}^{N_s^d}, \{\bar{P}^c_{si}\}_{i=1}^{N_s^c}\right),\\ &\min_i\left(\{\bar{P}^d_{ei}\}_{i=1}^{N_e^d}, \{\bar{P}^c_{ei}\}_{i=1}^{N_e^c}\right)\Big), \qquad (3)\\ \alpha_h = \min\Big(&\max_i\left(\{\bar{P}^d_{si}\}_{i=1}^{N_s^d}, \{\bar{P}^c_{si}\}_{i=1}^{N_s^c}\right),\\ &\max_i\left(\{\bar{P}^d_{ei}\}_{i=1}^{N_e^d}, \{\bar{P}^c_{ei}\}_{i=1}^{N_e^c}\right)\Big), \qquad (4)\end{aligned}$$

where the middle range and the low side range are illustrated in Fig. 2(a) and Fig. 2(d).

Step 2. Process data over the middle range $[\alpha_l, \alpha_h]$ for measuring. For some $i \in N_s^d$, there exist appropriate indexes $j, j-1 \in N_e^d$ subject to $\bar{P}^d_{ej-1} \le \bar{P}^d_{si} \le \bar{P}^d_{ej}$. Then, let us consider projection of the simulation data $(\bar{P}^d_{si}, \bar{\epsilon}^d_{si})$ onto a segment of the experimental data as a map Ψ,

$$\Psi : (\bar{P}^d_{si}, \bar{\epsilon}^d_{si}) \mapsto \left(\bar{P}^d_{si}, \frac{\bar{\epsilon}^d_{ej} - \bar{\epsilon}^d_{ej-1}}{\bar{P}^d_{ej} - \bar{P}^d_{ej-1}}(\bar{P}^d_{si} - \bar{P}^d_{ej}) + \bar{\epsilon}^d_{ej}\right),$$

where all of the projected points is denoted as $\Psi(\mathbb{D}_{sim,s})$. Similarly, the experimental data is projected onto a segment of the simulation data,

$$\Psi : (\bar{P}^d_{ei}, \bar{\epsilon}^d_{ei}) \mapsto \left(\bar{P}^d_{ei}, \frac{\bar{\epsilon}^d_{sj} - \bar{\epsilon}^d_{sj-1}}{\bar{P}^d_{sj} - \bar{P}^d_{sj-1}}(\bar{P}^d_{ei} - \bar{P}^d_{sj}) + \bar{\epsilon}^d_{sj}\right),$$

where all of the projected points is denoted as $\Psi(\mathbb{D}_{exp,s})$. Furthermore, if there exist cross points of the simulation and experimental data, the cross points are calculated, denoted as $\mathbb{D}_{cp}$. These process are illustrated in Fig. 2(b) and Fig. 2(e), where circles and boxes in black are the projected points. Additionally, the above projections have to be done in the contraction process as well, but the details are omitted here.

Step 3. Process data over the side ranges, $[0, \alpha_l]$ and $[\alpha_h, \infty]$ for measuring. Consider a point giving α_l via (3) and connect the point and all points over the range $[0, \alpha_l]$. A set of those points is denoted as $\mathbb{D}_{\alpha_l}$. Similarly, consider a point giving α_h via (4) and connect the point and all points over the range $[\alpha_h, \infty]$. A set of those points is denoted as $\mathbb{D}_{\alpha_h}$.

Step 4. To get ready for measuring the error in the pressure and the contraction ratio over time, the corresponding data, $\mathbb{D}_{sim,t}$ and $\mathbb{D}_{exp,t}$, is processed by using the same technique as the projection used in **Step 2.**, and then $\{\mathbb{D}_{sim,t}, \Psi(\mathbb{D}_{exp,t})\}$ and $\{\mathbb{D}_{exp,t}, \Psi(\mathbb{D}_{sim,t})\}$ are obtained.

Step 5. Sort the processed data $\{\mathbb{D}_{sim,s}, \Psi(\mathbb{D}_{exp,s}), \mathbb{D}_{cp}, \}$,

$\{\mathbb{D}_{exp,s}, \Psi(\mathbb{D}_{sim,s}), \mathbb{D}_{cp}\}$, $\mathbb{D}_{\alpha_l}$ and $\mathbb{D}_{\alpha_h}$, which are related to the steady state, and the other data $\{\mathbb{D}_{sim,t}, \Psi(\mathbb{D}_{exp,t})\}$ and $\{\mathbb{D}_{exp,t}, \Psi(\mathbb{D}_{sim,t})\}$, which are related to the time responses. They can yield the smallest triangles and trapezoids by a certain geometric technique such as tessellation, and their areas are summed to get a total, as illustrated in Fig. 2(c) and Fig. 2(f). Finally, this procedure can output the respective errors multiplied by minus one, i.e., $u(a) = -d(\mathbb{D}_{sim,s}(a), \mathbb{D}_{exp,s})$ in the steady-state case, or $u(a) = -d(\mathbb{D}_{sim,t}(a), \mathbb{D}_{exp,t})$ in the time response case.

Note that the above procedure is a polynomial time.

C. Parameter Identification by Game-Theoretic Learning

A game-theoretic learning algorithm for automatically finding the parameters of our interest is proposed in **Algorithm 1**. In the algorithm, two games are played: One is $(\mathcal{P}^s, \mathcal{A}^s, u)$ with five players of $p^s = (K, \theta, C_{q1}, -\frac{1}{C_{q2}}, c_c)$ and their actions $\Delta p^s = (\Delta K, \Delta\theta, \Delta C_{q1}, \Delta C_{q2}, \Delta c_c)$ and zero, which is played from line 2 to 8, and the other is $(\mathcal{P}^t, \mathcal{A}^t, u)$ with four players of $p^t = (A_0, k_1, \frac{1}{k_2}, \frac{1}{c_v})$ and their actions $\Delta p^t = (\Delta A_0, \Delta k_1, \Delta k_2, \Delta c_v)$ and zero, which is played from line 11 to 17. A reason why the two games can be played comes from the separability of the *Corollary 1* that helpfully reduces interaction among the players. Here, the *for loop* from line 1 to 9 expresses that the parameters are determined for each loads of M_1 to M_{N_M}. From this, N_M pairs of the parameters related to the steady state are obtained and then the interpolation over the load range can be done at line 18 to get five M-dependent continuous functions about p^s.

Algorithm 1 Parameter Identification Algorithm

Require: $\mathbb{D}^M_{exp}$, $k \leftarrow 0$, $p^s(0) \neq p^s(-1)$, $p^t(0) \neq p^t(-1)$
Ensure: functions $p^s(M)$ and values p^t.
1: **for** $j = 1$ to N_M **do**
2:   **while** for all i, $p^s_{ij}(k)$ is $p^s_{ij}(k-1)$ **do**
3:     **for** $i = 1$ to 5 **do**
4:       $a^*_i \leftarrow \arg\max_{a_i \in \mathcal{A}^s_i} -d(\mathbb{D}^{M_j}_{sim,s}(a_i, a_{-i}), \mathbb{D}^{M_j}_{exp,s})$
5:       $p^s_{ij}(k) \leftarrow p^s_{ij}(k-1) + a^*_i$
6:     **end for**
7:     $k \leftarrow k + 1$
8:   **end while**
9: **end for**
10: $k \leftarrow 0$
11: **while** for all i, $p^t_i(k)$ is $p^t_i(k-1)$ **do**
12:   **for** $i = 1$ to 4 **do**
13:     $a^*_i \leftarrow \arg\max_{a_i \in \mathcal{A}^t_i} -d(\mathbb{D}_{sim,t}(a_i, a_{-i}), \mathbb{D}_{exp,t})$
14:     $p^t_i(k) \leftarrow p^t_i(k-1) + a^*_i$
15:   **end for**
16:   $k \leftarrow k + 1$
17: **end while**
18: Parameters $p^s_{i\cdot}$ related to steady-state behaviors are interpolated over $[M_1, M_{N_M}]$ by a least-square approximation.

Theoretical analysis of the proposed algorithm can lead to the following property.

Theorem 1: Suppose that initial positive values of parameter $p^s(0)$ and $p^t(0)$, close to zeros, give well-possedness to the switched nonlinear system of the PAM (1). All of the actions a^*, generating p^s and the M-dependent continuous functions, at line 4 and a^*, generating p^t, at line 13 are Nash equilibria, respectively.

Proof: There are several facts to support the theorem that all action sets (non-negative step sizes) are finite, that they are fixed to each parameters, that a time sequence of $u(a)$ is a non-decreasing function, and that the conditions at lines 2 and 11 describes that if they are met, the corresponding game play is terminated, that is, at that time there are no rational actions maximizing the respective utility value $u(a)$ to each players. Then, the actions becomes a best response each other. Moreover, since it is clear the algorithm is a polynomial time, this algorithm can terminates in finite time. ■

We can find that the resulting values of the parameters related to the transient responses as well as the resulting functions of the parameters related to the steady state are generated by the Nash equilibrium. The algorithm is contributed to make the parameter identification automatic, compared to our previous work [16]. The Nash equilibrium, however, does not always provide the best performance in the sense of the parameter identification. Then, we actually need to verify what results can be obtained by applying the PAM model (1) and the proposed algorithm to the gray-box modeling of an actual PAM system, which will be discussed in the next section.

IV. Application and Verification using Practical PAM System

This section shows how the PAM model (1) and the proposed game-theoretic algorithm works when it is applied to the gray-box modeling of the practical PAM system. The PAM system includes a proportional directional control valve of Fig. 3 and is the same as one used in our previous work [16]. For the mathematical model of the PAM system, we try to identify the parameters of the model. Since the identification procedure requires experimental data set $\mathbb{D}_{exp}$, we prepared data sets for each loads: $M_1 = 1$, $M_2 = 2$,..., and $M_9 = 9$ [Kg], where $N_M = 9$. Following the game setting, define the parameters (players) as $p^s = (K, \theta, C_{q1}, -\frac{1}{C_{q2}}, c_c)$ and $p^t = (A_0, k_1, \frac{1}{k_2}, \frac{1}{c_v})$, set their initial values to $p^s(0) = (2.175 \times 10^4, 0.5147, 0.5430, 2.161 \times 10^{-5}, 1.414)$ and $p^t(0) = (0.014, 1.4, 0.01, 19.9)$, and set their step sizes (actions) to $\Delta p^s = (0.75 \times 10^4, 0.0129, 0.0187, 0.0424 \times 10^{-5}, 0.0488)$ and $\Delta p^t = (1.45 \times 10^{-9}, 0.035, 0.025, 0.25)$. Those initial parameter values give the well-posedness to the PAM model (1), which means that simulation can be successfully done without any numerical errors, such as state variables taking an imaginary number and division by zeros. Additionally, we are also interested in specifying a region of parameters giving the well-posedness, but it requires reachability analysis of the hybrid nonlinear dynamic system, which will be discussed another time.

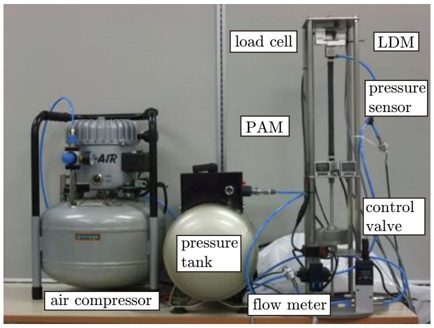


Fig. 3. Experimental equipment.

Resulting values and interpolated functions of the parameters obtained by applying the proposed identification, are listed in TABLE II. The steady-state and transient responses before applying the proposed game-learning algorithm are shown in Fig. 4(a) and Fig. 5(a), and the responses in case of the parameters learnt by the proposed algorithm are shown in Fig. 4(b) and Fig. 5(b), respectively, where the load is supposed to be M_1. From those figures, we see that the game learning properly functions to obtain the desirable parameter values and functions. In order to show a difference from our previous work [16], moreover, the steady-state and transient responses gotten by its heuristically trial-and-error-based algorithm are shown in Fig. 4(c) and Fig. 5(c). When looking at them, certainly the previous work seems to attain good parameters as well, but actually their utility values -1709 and -4.999×10^5 are respectively smaller than -1409 and -2.704×10^5 in case of the game-learning algorithm. This comparison means that the proposed algorithm is more effective because it automatically searches for the better (or the best) values in a sense of the utility. Here, processes of the parameters learning in a game play and of the utility values increasing toward zeros are shown in Fig. 6. The games in the steady state and transient responses were respectively played 62 and 43 times to each parameters, and the utility values go up to be close to the zeros.

It is clear that parameters giving $u(a^*) = -d(\cdot,\cdot) = 0$ are the global solution to the interesting optimization, which is an ideal situation in which there are no modeling errors, no sensing noises and the model has an appropriate structure to be able to catch the PAM dynamics. Such a situation is not realistic. Now that all of the obtained utility values are not zeros, the solutions should be local, but we can see how good the model with the obtained parameters is by checking the the utility value. In fact, we could obtain the better parameters than ones obtained by our previously-proposed algorithm [16].

Consequently, we practically show that the gray-box modeling consisting of the presented PAM model taking into account the load changes and the proposed game-theoretic learning algorithm for systematically identifying the physical parameters, works effectively[1].

[1]Model validation deviates a little from this paper's aim and there is a limitation of spaces so please see results of our previous works [14], [16].

TABLE II
IDENTIFICATION RESULTS OF THE PARAMETERS.

D_0 = 0.01 [m]	directly measurable parameters
L_0 = 0.250 [m]	
$D_1 = -347.8 \times 10^{-6}$[m^3]	
D_2 = 232.2 $\times 10^{-6}$[m^3]	
D_3 = 27.69 $\times 10^{-6}$[m^3]	
P_{tank} = 0.7013 $\times 10^6$ [Pa]	
P_{out} = 0.1013 $\times 10^6$ [Pa]	
k = 1.4 [-]	
R = 287 [J/Kg K]	
T = 293 [K]	
$K = (1.005M^2 - 0.9175M + 0.6435) \times 10^6$ [Kg]	for steady state
$\theta = (31.22\exp(-0.4332\mathrm{M}) + 31.44)\pi/180$ [rad]	
$C_{q1} = 0.1735 \log(10.00M) + 0.4183$ [$-$]	
$C_{q2} = (0.1133M - 1.451) \times 10^{-5}$ [1/Pa]	
$c_c = 1.8121M + 0.2318$ [N]	
A_0 = 0.051 $\times 10^{-6}$[m^2]	for transient
k_1 = 1.42, k_2 = 0.28 (extenstion) [-]	
k_1 = 1.42, k_2 = 0.36 (contraction) [-]	
c_v = 10 [N/(m/s)]	

V. CONCLUSIONS

This paper has proposed a systematic method for identifying the parameters of the PAM model, which based on the game-theoretic learning approach. It has been shown that the learning enables to automation how to determine the parameters using an error area between experimental and simulation data. From verification with the practical PAM system, we have could illustrate that the proposed game-theoretic algorithm is effective and enables to obtain the better parameters simulating the real behaviors of the PAM well.

As a future work, we have to mathematically analysis the parameters giving the well-posedness to the PAM system to specify the feasible parameter region, to incorporate the mixed strategy for improving the equilibrium, and to develop a gray-box modeling for antagonistic paris of PAMs to realize a position/force control system.

APPENDIX

This Appendix shows each components to give the hybrid model with the load changes (1), which is the generalization of our previous model [16] in terms of load M.

A. Contracting Force

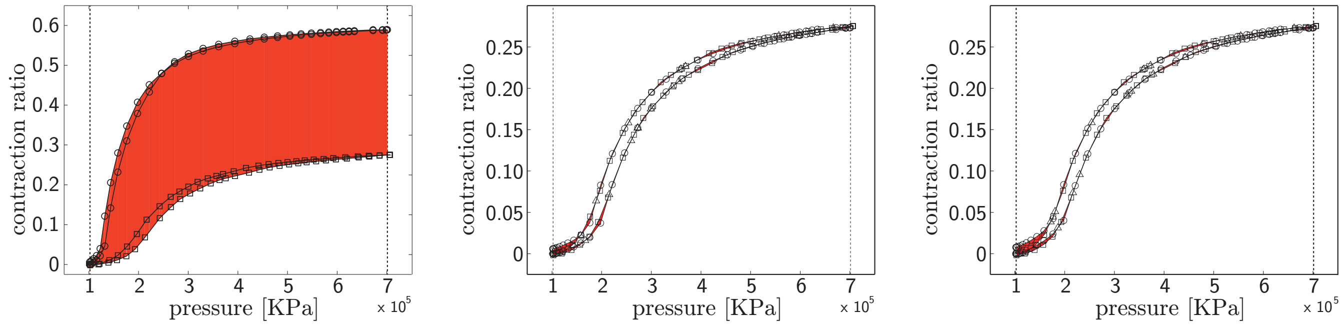
The contracting force is given by the following equation:

$$F(P,\epsilon,t) = \tfrac{\pi D_0^2}{4}(P(t) - P_{out})\left[\tfrac{3}{\tan^2\theta}\left\{1 - C_{q_1}(M)\times \left(1 + e^{C_{q_2}(M)(P(t)-P_{out})}\right)\epsilon(t)\right\}^2 - \tfrac{1}{\sin^2\theta}\right].$$

B. Pressure Change

The rate of change in inner pressure of the PAM is given by the following dynamic equation:

$$\dot{P}(t) = k_1 \frac{RT}{V(t)}\dot{m}(t) - k_2 \frac{\dot{V}(t)}{V(t)} P(t),$$

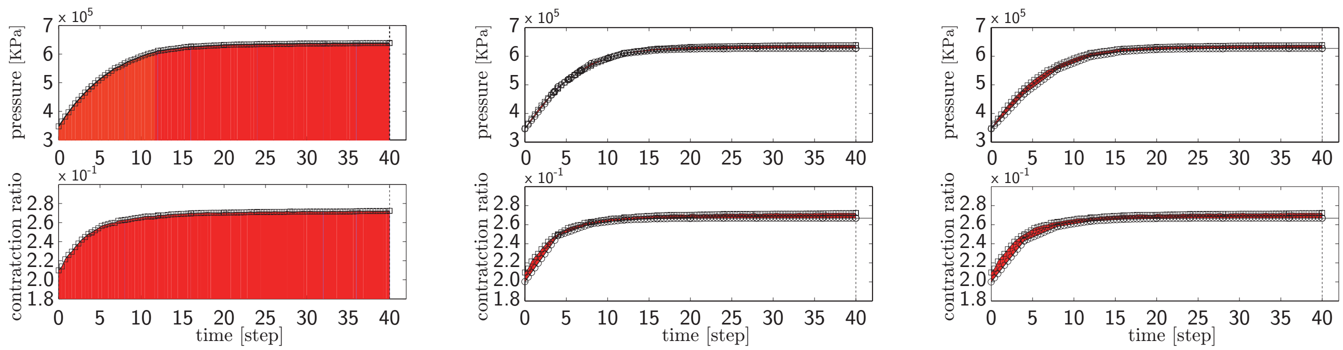


(a) Before the learning, then $u(a) = -362020$ with $p^s(0)$.

(b) After the learning, then $u(a^*) = -1409$ with $p^s(62)$.

(c) The heuristic algorithm in our work [16], then $-d(\mathbb{D}_{sim,s}, \mathbb{D}_{exp,s}) = -1709$.

Fig. 4. Comparison of steady-state responses of experimental data $\mathbb{D}_{exp,s}$ as '□' and simulation data $\mathbb{D}_{sim,s}$ as '○', before and after applying the proposed game-theoretic learning algorithm to the parameter identification, where '△' denotes a cross point. Areas in red expresses the error, $d(\mathbb{D}_{sim,s}, \mathbb{D}_{exp,s})$.

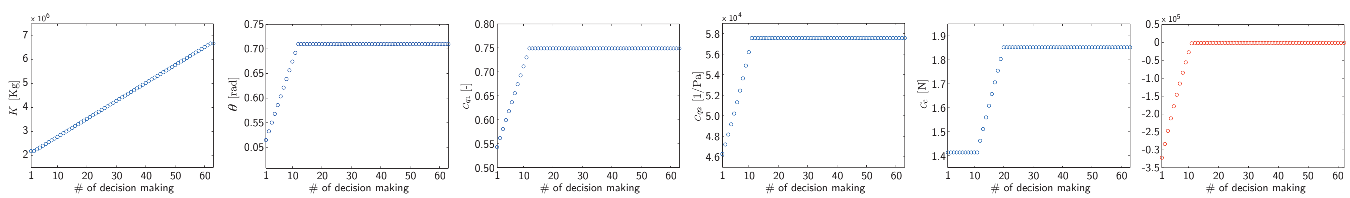


(a) Before the learning, then $u(a) = -2.086 \times 10^7$ with $p^t(0)$.

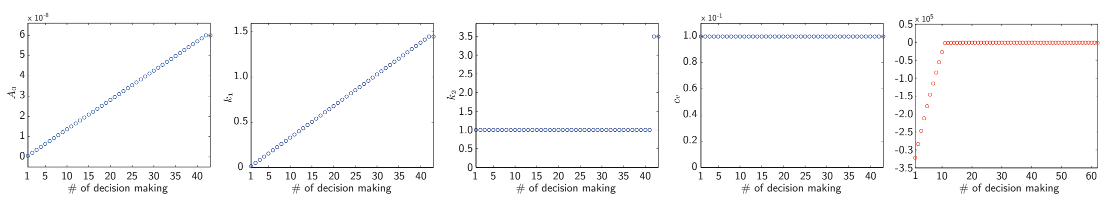
(b) After the learning, then $u(a^*) = -2.704 \times 10^5$ with $p^t(43)$.

(c) The heuristic algorithm in our work [16], then $-d(\mathbb{D}_{sim,s}, \mathbb{D}_{exp,s}) = -4.999 \times 10^5$.

Fig. 5. Comparison of time responses of experimental data $\mathbb{D}_{exp,t}$ as '□' and simulation data $\mathbb{D}_{sim,t}$ as '○', before and after applying the proposed game-theoretic learning algorithm to the parameter identification, where '△' denotes a cross point. Areas in red expresses the error, $d(\mathbb{D}_{sim,t}, \mathbb{D}_{exp,t})$.



(a) $p_1^s = K$. (b) $p_2^s = \theta$. (c) $p_3^s = C_{q1}$. (d) $p_4^s = -\frac{1}{C_{q2}}$. (e) $p_5^s = c_c$. (f) $u(a)$ in steady-state.



(g) $p_1^t = A_0$. (h) $p_2^t = k_1$. (i) $p_3^t = \frac{1}{k_2}$. (j) $p_4^t = \frac{1}{c_v}$. (k) $u(a)$ in transient.

Fig. 6. How the parameters and the utility values are updated during playing the games in the proposed algorithm.

where $\dot{V}$ is the volume change and k_1, $k_2 \in [1, 1.4]$ are polytropic indexes determined by the corresponding process.

C. Volume

The volume of the PAM is empirically defined as:

$$V(t) = D_1\epsilon(t)^2 + D_2\epsilon(t) + D_3,$$

where D_1, D_2, D_3 are determined from curve fitting with experimental volume data. Therefore,

$$\dot{V}(t) = 2D_1\dot{\epsilon}(t)(\epsilon(t) + D_2).$$

D. Mass Flow Rate in the Valve

The mass flow rate can be characterized by the following equation with a certain ratio $\alpha \in [0, 1]$ [14],

$$m(t) = \alpha(t)m_i(t) - (1 - \alpha(t))m_o(t),$$

where m_i is the mass flow rate from the intake port, which is specified below with $P_1 = P_{tank}$ and $P_2 = P$,

$$m_i(t) = \begin{cases} A_0 \frac{P_{tank}}{\sqrt{T}} \sqrt{\frac{k}{R}\left(\frac{2}{k+1}\right)^{\frac{k+1}{k-1}}} \\ \qquad \text{if} \quad P(t) \le \left(\frac{2}{k+1}\right)^{\frac{k}{k-1}} P_{tank}, \\ A_0 \frac{P_{tank}}{\sqrt{T}} \sqrt{\frac{2k}{R(k-1)}} \left(\frac{P(t)}{P_{tank}}\right)^{\frac{1}{k}} \sqrt{1 - \left(\frac{P(t)}{P_{tank}}\right)^{\frac{k-1}{k}}} \\ \qquad \text{if} \quad P(t) > \left(\frac{2}{k+1}\right)^{\frac{k}{k-1}} P_{tank}, \end{cases}$$

and m_o is the mass flow rate of the outlet port, which is specified with $P_1 = P$ and $P_2 = P_{out}$,

$$m_o(t) = \begin{cases} A_0 \frac{P(t)}{\sqrt{T}} \sqrt{\frac{k}{R}\left(\frac{2}{k+1}\right)^{\frac{k+1}{k-1}}} \\ \qquad \text{if} \quad P(t)\left(\frac{2}{k+1}\right)^{\frac{k}{k-1}} \ge P_{out}, \\ A_0 \frac{P(t)}{\sqrt{T}} \sqrt{\frac{2k}{R(k-1)}} \left(\frac{P_{out}}{P(t)}\right)^{\frac{1}{k}} \sqrt{1 - \left(\frac{P_{out}}{P(t)}\right)^{\frac{k-1}{k}}} \\ \qquad \text{if} \quad P(t)\left(\frac{2}{k+1}\right)^{\frac{k}{k-1}} < P_{out}. \end{cases}$$

As for α, it depends on where the spool is located by a command voltage u. Then, α is a function of u, i.e., $\alpha = \kappa(u)$, where κ is a monotonically increasing function with respect to $u \in \mathcal{U} := [\kappa^{-1}(0),\ \kappa^{-1}(1)]$.

E. Dynamic Equation Related to Weight

The dynamic equation of the load can be formulated as

$$ML\ddot{\epsilon}(t) = \begin{cases} F(P,\epsilon,t) - Mg - F_f(t) - \\ \qquad K(M)(L_0 - L(1-\epsilon(t)))^3, \quad \text{if} \quad \epsilon(t) \le \frac{L-L_0}{L}, \\ F(P,\epsilon,t) - Mg - F_f(t), \qquad \text{otherwise}, \end{cases}$$

where K is a coefficient of elasticity (a ratio of the restoring force to the difference between the measured length of the PAM and its natural length in the absence of the load), and F_f is a friction term following the switched dynamic below. If $\epsilon(t) \le (L - L_0)/L$, then $F_o = F - Mg - K(L_0 - L(1 - \epsilon))^3$; else $F_o = F - Mg$. The friction term is described below [15]:

$$F_f(t) = \begin{cases} c_v L\dot{\epsilon}(t) + c_c(M)\mathrm{sgn}(\dot{\epsilon}(t)), \quad \text{if} \ \dot{\epsilon}(t) \ne 0, \\ c_c(M), \quad \text{if} \ \dot{\epsilon}(t) = 0 \ \text{and} \ F_o(t) > c_c(M), \\ F_o(t), \quad \text{if} \ \dot{\epsilon}(t) = 0 \ \text{and} \\ \qquad\qquad F_o(t) \in [-c_c(M), c_c(M)], \\ -c_c(M), \quad \text{if} \ \dot{\epsilon}(t) = 0 \ \text{and} \ F_o(t) < -c_c(M), \end{cases}$$

where F_o is the net force acting on the PAM system, c_c is the Coulomb friction force and c_v is the viscous friction coefficient.

Note that the weight is hung from the PAM even if the gauge pressure is zero, i.e., $P = P_{out}$. In the zero case, there is no contraction force generated in F, and there is no force to balance the weight. To modify the force to balance, the PAM is considered an elastic body.

References

[1] C. P. Chou and B. Hannaford, "Measurement and modeling of McKibben pneumatic artificial muscles," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 1, pp. 90–102, 1996.

[2] B. Tondu and P. Lopez, "Modeling and control of McKibben artificial muscle robot actuators," *IEEE Control Systems Magazine*, vol. 20, no. 2, pp. 15–38, 2000.

[3] B. S. Kang, C. S. Kothera, B. K. S. Woods, and N. M. Wereley, "Dynamic modeling of pneumatic artificial muscles for antagonistic actuation," in *IEEE International Conference on Robotics and Automation*, 2009, pp. 182–187.

[4] G. A. Medrano-Cerda, C. J. Bowler, and D. G. Caldwell, "Adaptive position control of antagonistic pneumatic muscle actuators," in *IEEE/RSJ International Conference on Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots'*, vol. 1, aug. 1995, pp. 378–383.

[5] T. Kimura, S. Hara, T. Fujita, and T. Kagawa, "Feedback linearization for pneumatic actuator systems with static friction," *Control Engineering Practice*, vol. 5, no. 10, pp. 1385–1394, 1997.

[6] A. P-Arrese, A. Mendizabal, J. Arenas, R. Prestamero, and J. Landaluze, "Modelling in modelica and position control of a 1-DoF set-up powered by pneumatic muscles," *Mechatronics*, vol. 20, no. 5, pp. 535–552, 2010.

[7] J. H. Lilly and Y. Liang, "Sliding mode tracking for pneumatic muscle actuators in opposing pair configuration," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 4, pp. 550–558, 2005.

[8] D. W. Repperger, C. A. Phillips, and M. Krier, "Controller design involving gain scheduling for a large scale pneumatic muscle actuator," in *IEEE International Conference on Control Applications*, vol. 1, 1999, pp. 285–290.

[9] T. Vo-Minh, T. Tjahjowidodo, H. Ramon, and H. V. Brussel, "Cascade position control of a single pneumatic artificial muscle-mass system with hysteresis compensation," *Mechatronics*, vol. 20, no. 3, pp. 402–414, 2010.

[10] S. Ganguly, A. Garg, A. Pasricha, and S. Dwivedy, "Control of pneumatic artificial muscle system through experimental modeling," *Mechatronics*, vol. 22, no. 8, pp. 1135–1147, 2012.

[11] K. Hosoda, Y. Sakaguchi, H. Takayama, and T. Takuma, "Pneumatic-driven jumping robot with anthropomorphic muscular skeleton structure," *Autonomous Robots*, vol. 28, pp. 307–316, 2010.

[12] R. Niiyama, "Model-based trajectory control of robots with pneumatic actuator dynamics," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 634–639.

[13] T. Vo-Minh, T. Tjahjowidodo, H. Ramon, and H. V. Brussel, "A new approach to molding hysteresis in a pneumatic artificial muscle using the maxwell-slip model," *IEEE/ASME Transactions on Mechatronics*, vol. 16, no. 1, pp. 177–186, 2011.

[14] T. Itto and K. Kogiso, "Hybrid modeling of McKibben pneumatic artificial muscle systems," in *Joint IEEE International Conference on Industrial Technology & 43rd Southeastern symposium on System Theory*, 2011, pp. 57–62.

[15] R. Kikuuwe, N. Takesue, A. Sano, H. Mochiyama, and H. Fujimoto, "Admittance and impedance representations of friction based on implicit euler integration," *IEEE Transactions on Robotics*, vol. 22, no. 6, pp. 1176–1188, 2006.

[16] K. Kogiso, K. Sawano, T. Itto, and K. Sugimoto, "Identification procedure for McKibben pneumatic artificial muscle systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 3714–3721.

[17] Y. Honda, F. Miyazaki, and A. Nishikawa, "Angle control of pneumatically-driven musculoskeletal model using antagonistic muscle ratio and antagonistic muscle activity," in *IEEE International Conference on Robotics and Biomimetics*, 2010, pp. 1722–1727.

[18] K. Narioka and K. Hosoda, "Motor development of an pneumatic musculoskeletal infant robot," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 963–968.

[19] E. Nernholt and S. Moberg, "Nonlinear gray-box identification using local models applied to industrial robots," *Automatica*, vol. 47, no. 4, pp. 650–660, 2011.