Learning the Elasticity of a Series-Elastic Actuator for Accurate Torque Control

Bingbin Yu, José de Gea Fernández, Yohannes Kassahun, and Vinzenz Bargsten

DFKI, Robotics Innovation Center,
Bremen, 28359, Germany
{bingbin.yu,jose.de_gea_fernandez,
yohannes.kassahun,vinzenz.bargsten}@dfki.de

1 Abstract

Series elastic actuators (SEAs) have been frequently used in torque control mode by using the elastic element as torque measuring device. In order to precisely control the torque, an ideal torque source is critical for higher level control strategies. The elastic elements are traditionally metal springs which are normally considered as linear elements in the control scheme. However, many elastic elements are not perfectly linear, especially for an elastic element built out of multiple springs or using special materials [1][2] and thus their nonlinearities are very noticeable. This paper presents two data-driven methods for learning the spring model of a series-elastic actuator: (1) a Dynamic Gaussian Mixture Model (DGMM) [3] is used to capture the relationship between actuator torque, velocity, spring deflection and its history. Once the DGMM is trained, the spring deflection can be estimated by using the conditional probability function which later is used for torque control. For comparison, (2) a deep-learning approach is also evaluated which uses the same variables as training data for learning the spring model. Results show that the data-driven methods improve the accuracy of the torque control as compared to traditional linear models.

Keywords: series-elastic actuators, nonlinear springs, DGMM, deep learning, torque control

2 Introduction

In recent years, robots are increasingly developed to assist humans on direct physical interaction, not only in the field of assistance and rehabilitation robotics [4], but also start to be used in industrial scenarios [5]. For these robots that work close to humans in shared workspaces, safety is of outmost concern (especially for industrial robots which normally are fast and powerful). To achieve a safe human-robot interaction, one possible solution is to use a compliant actuator that is able to immediately sense the torque and accommodate for external force disturbances. For a rigid actuator, the torque can be measured by torque

sensors, e.g. a load cell, a strain gauge or a current sensor, and then be controlled by using a feedback loop [6]. Different from a rigid actuator, a serial elastic actuator (SEA) can estimate the torque from the deflection of its elastic element. Due to the passive compliance between the actuator and its link, SEAs provide additional benefits including lower reflected inertia, greater shock tolerance and more accurate force control.

A large number of SEA designs has already been developed, for instance as surveyed in [7],[4]. A typical design is a linear SEA, in which the spring system either uses a single spring [8] or a set of serial-connected springs [9], which connect to the motor through a ball screw. For a rotary series elastic actuator (RSEA) the design of the elastic coupling that restricts the size and reduces the weight of the device is usually challenging. For example, Kong and Jeon developed a compact RSEA with a coil spring and worm gears for knee joint assistance [10]; Stienen et al. developed a rotational hydroelastic actuator with a symmetric torsion spring for a powered exoskeleton [11]; or the elastic element of the CAPIO actuator [12] includes a set of small disc springs placed at both sides of a lever which connects to the link. In recent years, new elastic materials are also utilized: scientists at the Carnegie Mellon University used nonlinear rubber as the elastic element of the actuators for their snake robots [1]; Sudano et al. integrated a magnetic nonlinear torsion spring in a rotary elastic actuator for biorobotic applications [13]. However, due to mechanical effects caused by the construction itself, by the structure of the spring system (e.g. different initial pre-compression of coil springs), or the properties of the materials, many of these elastic couplings show very poor linearity, which is usually neglected. In this work, we propose two data-driven methods for modeling the torque profile of an SEA, which consider the nonlinearity of the elastic couplings for realizing better torque control approaches. The data-driven modeling methods are validated and compared using a newly designed RSEA.

Various torque control approaches have already been proposed for SEAs, e.g. velocity-source control [14], a cascade control by using velocity or current in inner loop and torque in outer loop; or feedforward force control with disturbance observer [15]. The performances of these higher level control strategies are influenced by the torque sources, if the nonlinearity of the spring and resistive frictions are too large, a precise model of the elastic element and the frictions is needed. Therefore, Ford et al. [16] proposed an online calibration method to compensate the nonlinear effects of the spring and accurately estimate the modules output torque by using motor current and spring deflection together. Lu [17] modelled the nonlinearity of the spring of a SEA by using a BP neural network and realized a stable velocity control. The paper is organized as follows. In Section 3, the hardware design of the RSEA and analysis of the spring coupling are presented. In Section 4, the two modeling methods of the elastic element are discussed. The experimental validation of the two models is performed in Section 5. In Section 6, based on the learned models, a torque control task is demonstrated. Finally, conclusions are given in Section 7.

3 SEA design and spring analysis

The assembled serial elastic actuator (see Fig. 1 left) is designed within the project FourByThree [18]. The actuator is powered by a Robodrive brushless DC motor and provides a maximum 50 Nm torque and 15 rpm speed in the link side by using a 1:120 Harmonic Drive gear. An FPGA (Spartan6)-based control stack incorporates all the required sensors (three absolute encoders, two motor current sensors, temperature sensors, etc.) and perform the required actuator control with an in-house developed communication protocol. A new elastic element based on coil springs has been developed for the actuator (Figure 1 right) which consists two springs in each spring segment: a lower-stiff spring firstly compresses singly until its deflection reaches approx. 5 degree, then a smaller higher-stiff spring which is placed inside the other starts to work. By using this design, the elastic spring is relatively 'soft' in the lower torque range, so that it provides a higher torque to deflection resolution in this torque range. Since the elastic spring is 'stiff' for a higher torque input, it brings a larger working range and avoid that the spring completely compresses at the maximum torque.

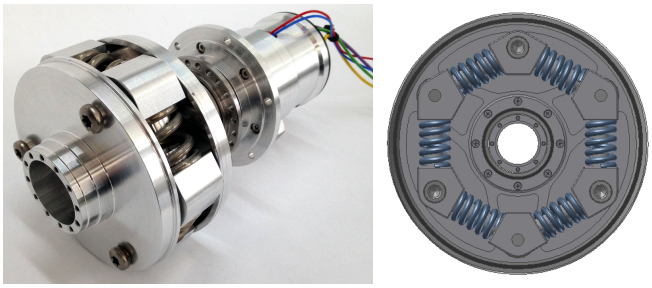

Fig. 1. *left:* FourByThree 50Nm-Actuator. *right:* Elastic element based on coil springs.

As shown in Figure 1, coil springs are used and each single spring is a linear element. However due to the internal friction and different pre-compression during assembly, the torque-deflection curve of the overall spring module is nonlinear. Figure 2 shows the result of an experiment used to demonstrate the nonlinearity of the spring coupling. In this experiment, the elastic actuator is controlled to a fixed rotation angle in position control. An external force/torque sensor (Lorenz-DF30) is employed to provide a torque ground truth in a range of -50Nm to +50Nm with an accuracy class of 0.05%. The motor is fixed on a test bed, the external torque is externally applied to the spring through the link lever in both directions. As the plot shows, the torque-deflection model of the spring presents a hysteresis characteristic, where a simple linear regression line is a poor choice of representation.

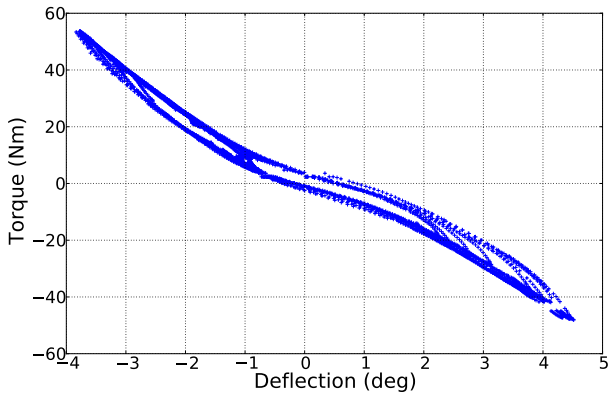

Fig. 2. Torque to deflection curve of the spring coupling. The output torque is measured with an external force/torque sensor and the deflection is measured by computing the difference of two absolute encoders at both sides of the spring.

4 Spring modeling methods

Since a spring is used as a torque sensor in a series-elastic actuator, an accurate spring model is the basis of a successful and accurate torque control. As discussed in Section 3, the torque-deflection curve of the coil spring component exhibits a nonlinear property. In order to account for the nonlinearity, two data-driven modeling approaches are used: a dynamic Gaussian mixture model (DGMM) and (2) a neural network (NN). Both methods will be described briefly in the following sections.

4.1 Spring modeling by DGMM

As a probabilistic modeling method, Gaussian mixture models (GMM) [3] are widely used in modeling complex and multi-variable data. To model the spring with more variables besides deflection, a dynamic Gaussian mixture model (DGMM) is studied, which represents a probability density function $P(x)$ as a variable-sized set of weighted Gaussian pairs (Eq. 1).

$$P(x) = \Sigma_{i=1}^{m} \hat{\omega_i} g_i(x), \tag{1}$$

where $g_i(x)$ is a multivariate Gaussian distribution

$$g_i(x) = p_i(x) \sim \mathcal{N}_i(\mu_i, \Sigma_i), \tag{2}$$

and $\hat{\omega_i}$ is the weight of the Gaussian $g_i(x)$

$$\hat{\omega_i} = \omega_i / \Sigma_{k=1}^{m}. \tag{3}$$

The quantity x is the observation vector. As Figure 1 shows, the torque to deflection curve of the spring coupling is nonlinear, therefore more variables are required: the rotation velocity v and the history of the spring deflection δ'. Consequently, the observation vector is

$$x = [\tau, \delta, \delta', v], \tag{4}$$

and the model of the system can be represented by

$$P[\tau, \delta, \delta', v]. \tag{5}$$

As a result, once a DGMM model $P[\tau, \delta, \delta', v]$ is learned with a training data set, the output torque can be estimated analytically as

$$\mathbb{E}[\tau|\delta, \delta', v]. \tag{6}$$

For control purposes one can also similarly estimate the deflection using

$$\mathbb{E}[\delta|\tau, \delta', v]. \tag{7}$$

4.2 Spring modeling by using neural network

An artificial neural network consists of an interconnected assembly of simple processing units [19]. In most of the cases, each processing unit calculates its output by taking a weighted sum of its inputs and transforming the sum by an activation function. In addition to the connection weights, the function represented by an artificial neural network is determined by the architecture of the neural network. Because of the possibility of optimizing a large number of parameters due to the advancement in computing, neural networks are used in various application areas such as computer vision, reinforcement learning, speech recognition and natural language processing resulting in progress beyond the state-of-the-art in terms of performance in most of the cases.

In this paper, five layered neural networks are used for modeling the spring component, which are realized using an open-source Deep Learning tool Keras [20]. Unlike DGMM, two separate networks are trained independently: once for $\mathbb{E}[\tau|\delta, \delta', v]$ and once for $\mathbb{E}[\delta|\tau, \delta', v]$. In both networks, three hidden layers are created with 50, 20 and 10 neurons with relu activation function in each layer. A relu is defined as $f(a) = \max(0, a)$, where a is a weighted sum of the inputs to a unit. To optimize the networks, "Adam" [21] is used which is an algorithm for the first-order gradient-based optimization of stochastic objective functions. After training, 771 parameters are learned for the networks of inverse model $\mathbb{E}[\delta|\tau, \delta', v]$.

5 Spring models validation

In order to validate the two modeling approaches, an experiment setup has been constructed as shown Figure 3. The FourByThree 50 Nm SEA (for more details,

see Section 3) is used and fixed on an adjustable base, so that the inclination of the actuator can be changed and the effects of gravity can be accounted for. An external force/torque sensor is mounted between the spring coupling and the link lever, which measures the output torque with a 125 Hz sampling frequency in order to validate the results.

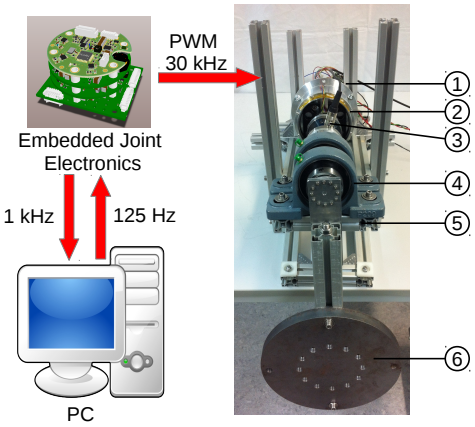

Fig. 3. Experimental setup used for spring modeling and torque control: 1) actuator; 2) spring coupling; 3) external force/torque sensor; 4) brake; 5) adjustable base; 6) load.

The models need to be trained first and then be used in the validation phase by estimating the output torque given measured variables. In the training procedure, the inclination angle of the actuator base is set to 0 degree and the training data is gathered by controlling the load to rotate in the range of approx. ± 170 degrees with position control smoothly in a very low speed. The position of the load at the link lever is changed in each train test and the actuator torque τ, deflection δ, first derivative of deflection δ' and velocity v are measured as the training inputs. The spring deflection is extracted from two absolute encoders, the first derivative of deflection δ' is calculated from the change of the deflection and the time used in a control cycle and the velocity v is acquired from the position sensor.

In the validation phase, the same load is fixed at random selected position on the lever arm and the inclination angle of the base is set to 37.8 degree. Based on the trained DGMM and neural network models, the actuator torque τ can be estimated by both methods respectively. Figure 4 shows the comparison results between the trained DGMM model $P[\tau, \delta, \delta', v]$, the neural network model, and the linear regression model.

As can be seen from the upper left and upper right plots, both DGMM and neural network models are able to predict the output torque given the measured variables with high accuracy. In contrast, the fitted linear regression function

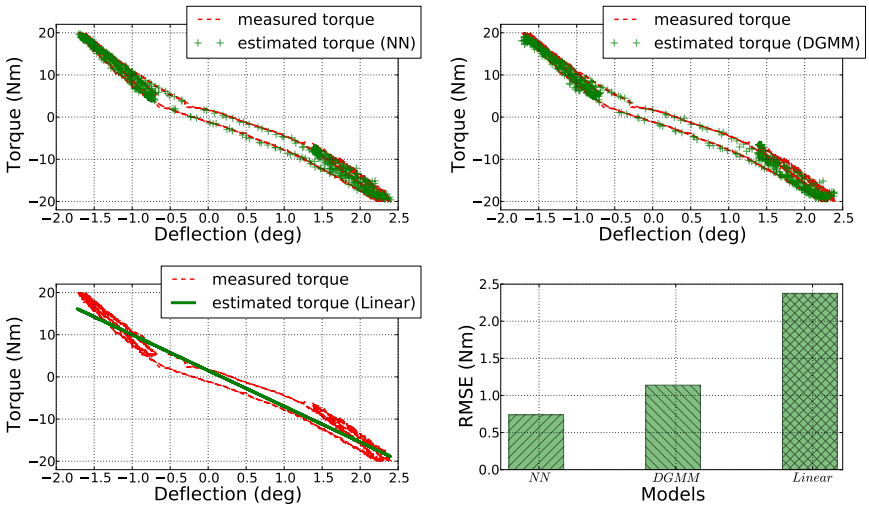

Fig. 4. *Upper left:* measured torque-deflection curve (dash line) and estimated torque with neural network model (crosses), *Upper right:* measured torque-deflection curves (dash line) and estimated torque with DGMM model (crosses), *lower left:* measured torque-deflection curve (dash line) and corresponding fitted linear line (line), *lower right:* root mean square errors (bars) of the three models in torque estimation.

has problems to represent the torque-deflection curve (see lower left plot). The RMSE between the estimated torque and measured torque is calculated for each model, as the lower right plot shows; the DGMM and neural network models present a comparable performances and have a large advantage compared to the linear model.

6 Torque control

Based on the learned spring models, a torque controller is proposed to control the spring torque to track the desired torque as precisely as possible. The complete torque control scheme of the SEA consists of three cascaded control loops for motor current, spring deflection and torque (see Fig. 5). Two absolute encoders are installed at both sides of the spring coupling for measuring the rotation angle and calculate the spring deflection. A deflection PID controller is implemented into the FPGA which closes the loop with the spring deflection and then cascades with an inner motor current controller. The first derivative of the deflection and the velocity are calculated to be used as the inputs of the spring model, together with the given desired torque, and the corresponding deflection value is estimated. This deflection will be then controlled by the inner deflection and current controllers. Due to an intrinsic property of the DGMM model, once the model $P[\tau, \delta, \delta', v]$ is learned from the training experiments, both forward

$E[\tau|\delta, \delta', v]$ and inverse models $E[\delta|\tau, \delta', v]$ can be estimated, whereas the inverse model is then used in the torque control. In contrast, the forward and inverse models need to be trained individually when using a neural network model.

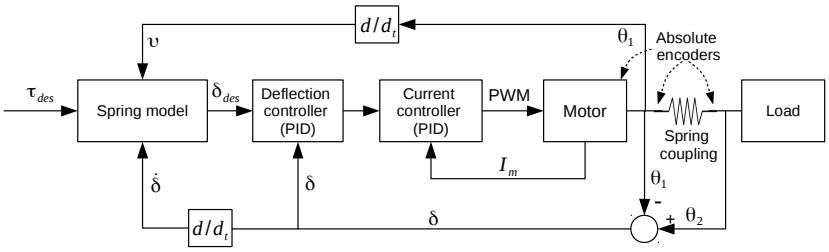

Fig. 5. Complete actuator torque control scheme.

The proposed torque control is verified in two torque tracking experiments. A chirp signal and a random-walk are given as the desired torques in these two experiments respectively and the different controllers which are based on different spring models are evaluated in measured output torques (see plot a,b of Fig. 6). As can be seen from the comparison in tracking errors (see plot c,d), the torque control by using DGMM model and neural networks present better results than by using linear model in both experiments. Since the viscous friction of the system can not be compensated by the spring models completely, the performance of random-walk tracking is better than chirp tracking, and the error of the chirp tracking is also raised slightly when the reference frequency is increased.

7 Conclusion

In this paper, two data-driven modeling methods are proposed to account for the nonlinearities of the spring coupling of a rotary elastic actuator. The models are learned from the training experiments of the actuator and verified by estimating the output torque with given measured variables in the test experiments. The experiment result presents a comparable performances of the DGMM and the deep learning/NN methods, which show a significant advantage compared to a linear regression model. As compared to the DGMM model, the deep learning model shows a slight improvement in torque estimation. On the other side, the DGMM model captures multiple relationship among the observed variables, which is more flexible to be utilized once learned in multiple ways by choosing which variables are used as inputs and which ones as outputs of the model. The learned nonlinear spring model is then used as an torque estimation module for a torque controller, which cascades with an inner motor current and deflection control loops. The proposed torque controller is verified by a set of experiments which demonstrated a precise torque control.

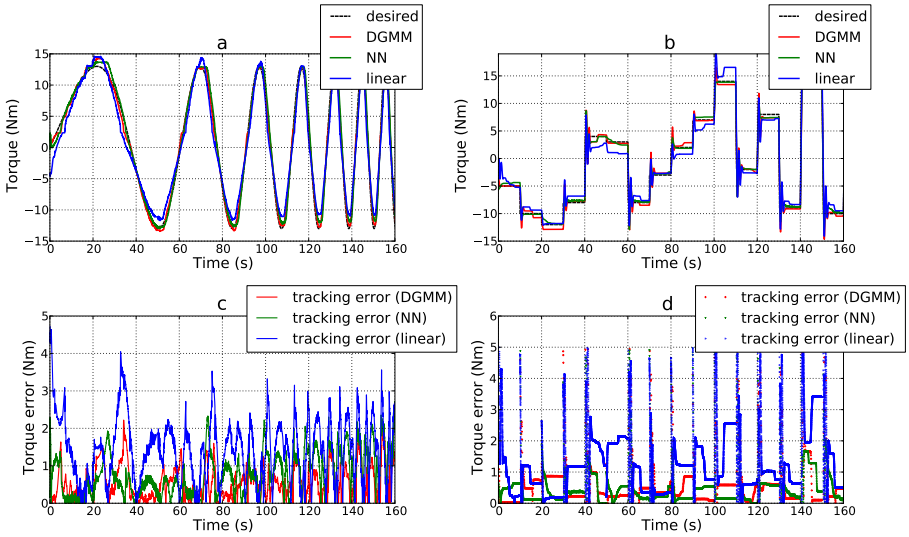

Fig. 6. *a:* Result of the torque tracking with a chirp reference signal, *b:* Result of the torque tracking with a random walk reference signal, *c:* Torque tracking error with given chirp reference, the root mean square errors are $RMSE_{DGMM} = 0.93$, $RMSE_{NN} = 1.07$ and $RMSE_{linear} = 1.77$, *d:* Torque tracking error with given random walk reference, the root mean square errors are $RMSE_{DGMM} = 0.90$, $RMSE_{NN} = 0.80$ and $RMSE_{linear} = 1.61$ respectively.

8 Acknowledgment

The FourByThree project has received funding from the European Union's Horizon 2020 research and innovation programme, under Grant Agreement No. 637095.

References

1. Rollinson, D., Ford, S., Brown, B., Choset, H.: DESIGN AND MODELING OF A SERIES ELASTIC ELEMENT FOR SNAKE ROBOTS. In: Proceedings of the ASME 2013 Dynamic Systems and Control Conference, (2013)
2. Paskarbeit, J., Annunziata, S., Basa, D., Schneider, A.: A self-contained, elastic joint drive for robotics applications based on a sensorized elastomer couplingDesign and identification, In: Sensors and Actuators A: Physical, vol. 199, pp. 56–66. Springer, Heidelberg (2013)
3. Edgington, M., Kassahun, Y., Kirchner, F.: Dynamic motion modelling for legged robots. In: 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4688–4694 (2009)
4. Yu, H., Huang, S., Thakor, N.V., Chen, G., Toh, S.L.: A novel compact compliant actuator design for rehabilitation robots. In: 2013 IEEE international conference on rehabilitation robotics,(2013)

5. Rethink Robotics, www.rethinkrobotics.com
6. Bargsten, V., de Gea Fernández, J.: COMPI: Development of a 6-DOF compliant robot arm for human-robot cooperation In: In Proceedings of the 8th International Workshop on Human-Friendly Robotics, (2015)
7. Paine, N., Oh, S., Sentis, L.: Design and Control Considerations for High-Performance Series Elastic Actuators. In: IEEE/ASME TRANSACTIONS ON MECHATRONICS, vol.19, no.3, (2014)
8. Pratt, G.A., Williamson, M.M.: Series Elastic Actuators. In: IEEE International Conference on Intelligent Robots and Systems, vol.1, pp. 399–406, (1995)
9. Arumugom, S., Muthuraman, S., Ponselvan, V.: Modeling and application of series elastic actuators for force control multi legged robots. In: Journal of computing, vol.1,(2009)
10. Kong, K., Bae, J., Tomizuka, M.: A Compact Rotary Series Elastic Actuator for Human Assistive Systems In: IEEE/ASME Transations of Mechatronics,vol.17,no.2,(2012)
11. Stienen, A.H.A., Hekman, E.E.G., Braak, H., Aalsma, A.M.M., Van der Helm, F.C.T., van der Kooij, H.: Design of a Rotational Hydroelastic Actuator for a Powered Exoskeleton for Upper Limb Rehabilitation. In: IEEE transactions on biomedical engineering, vol.57, no.3,(2010)
12. Mallwitz, M., Will, N., Teiwes, J., Kirchner, E.A.: The CAPIO Active Upper Body Exoskeleton and its Application for Teleoperation. In: Proceedings of the 13th Symposium on Advanced Space Technologies in Robotics and Automation. ESA/Estec Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA),(2015)
13. Sudano, A., Tagliamonte, N.L, Accoto, D., Guglielmelli, E.: A Resonant Parallel Elastic Actuator for Biorobotic Applications. In: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), (2014)
14. Wyeth, G.: Demonstrating the Safety and Performance of a Velocity Sourced Series Elastic Actuator In: 2008 IEEE International Conference on Robotics and Automation,(2008)
15. Li, Y., Feng, H.: Force control of series elastic acutator In: 2015 Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control, (2015)
16. Ford, S., Rollinson, D., Willig, A., Choset, H.: Online Calibration of a Compact Series Elastic Actuator In: 2014 American Control Conference (2014)
17. Lu, C., Mao,Y., Zhu, Q., Xiong, R.: Novel series elastic actuator design and velocity control In: Electric Machines and Control, vol.19, pp.83–88, (2015)
18. de Gea Fernández, J., Sprengel, H., Mallwitz, M., Zipper, M., Yu, B., Bargsten, V.: Designing Modular Series-Elastic Actuators for Safe Human-Robot Collaboration in Industrial Settings. In: Proceedings of the Climbing and Walking Robots and Support Technologies for Mobile Machines (CLAWAR),(2016)
19. Bishop, C. M. Neural networks for pattern recognition. Oxford university press. (1995)
20. Keras: Deep Learning library for Theano and TensorFlow, https://keras.io/
21. Kingma, D. P., Ba, J. L.:Adam: a Method for Stochastic Optimization, In: International Conference on Learning Representations (ICLR),(2015)