A Learning Approach for Feed-Forward Friction Compensation

Viktor Johansson * Stig Moberg ** Erik Hedberg ***
Mikael Norrlöf*** Svante Gunnarsson***

* AstaZero, Sandhult, Sweden. (e-mail: viktor.johansson@astazero.com).
** Robotics and Motion Division, ABB AB, Västerås, Sweden
*** Department of Electrical Engineering, Linköping University, 58183 Linköping, Sweden (e-mail: first.last@liu.se)

Abstract: An experimental comparison of two feed-forward based friction compensation methods is presented. The first method is based on the LuGre friction model, using identified friction model parameters, and the second method is based on B-spline network, where the network weights are learned from experiments. The methods are evaluated and compared via experiments using a six axis industrial robot carrying out circular movements of different radii. The experiments show that the learning-based friction compensation gives an error reduction of the same magnitude as for the LuGre-based friction compensation.

© 2018, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.

Keywords: Friction, feed-forward compensation, splines, learning control, industrial robots

1. INTRODUCTION

To improve the path accuracy of industrial robots, a range of different methods can be applied. One such method is iterative learning control (ILC), which has proven feasible in practice. Iterative learning control is a method of learning a control scheme for a particular motion by iterating the motion and improving control based on the tracking error. The control is most often applied as an extension to an already existing feedback control. As pointed out in Yin et al. (2015) the method is mainly considered when an identical task is repeated for the same trajectory. However, when the reference is changed, the learned control input cannot be reused. This means that the learning does not generalize to new conditions.

An alternative method is to apply feed-forward control to the robot. This type of control aims to compensate for the internal dynamics in the system by compensating for its effect prior to the effect can be seen in the output. In feed-forward control the inverse of the plant model is often used to determine what the control input should be, such that the desired reference signal is achieved. This way the known dynamics and disturbances in the plant can be accounted for, which in turn results in higher path-following accuracy for different trajectories and tool load configurations.

Utilizing feed-forward control however requires accurate models of the plant. Deriving a model of the plant can be done through different methods, such as physical modeling and linear/nonlinear system identification methods, which are covered in Ljung (1999). Often an input-output model is necessary such that the control signal can be calculated and added to already existing control. A general model has the ability to include a range of different kinematic and dynamic behaviour which accurately describes the plant in a wide range of conditions. In the case of multi-axis industrial robots, actuator models and flexibility models as described in (Siciliano and Khatib, 2016, p. 113-138) can further improve precision and accuracy of the system model.

However, deriving a complete and accurate model of the system is in general complicated and time consuming. Some parameters in the model are also varying depending on temperature and wear. As an alternative it is therefore often preferable to model these effects locally such that particular non-linearities, internal disturbances and model errors can be compensated for with high accuracy in a region rather than globally.

In the context of industrial robots, the model elements that could be modeled locally could be the effects of errors in the model of inertia, friction in the motor and flexibility in the links.

In order to model and identify parameters of multi-axis industrial robots there exists a range of different data-driven approaches. One such example is presented in Camoriano et al. (2016), where the authors describe a methodology to learn the inverse dynamics of the robot structure through a semi-parametric approach. The method relies on a parametric model of the robot constructed through rigid body dynamics (RBD) and a non-parametric approach using incremental kernel methods. As a result the combination of both models provides a trait to enable robotic systems to adapt to changing conditions of the environment and the mechanical properties of the robot. This could for example be used to mitigate certain internal disturbances and increase the performance of the system.
Furthermore, in Ljung (1999) multiple data-driven nonlinear identification methods are presented. Among the presented methods are various types of function approximators for systems as well as neural network methods. Examples of the latter are Feed-Forward Neural Network (FFNN) and Recurrent Neural Network (RNN) which can be used for identifying nonlinear systems without extensive physical modeling. Depending on the choice of general functions for the neural networks, they are associated with different names.

Other methods such a Gaussian Process Regression (GPR) have also been utilized to learn system dynamics, as can be seen in Meier and Schaal (2016). The authors present an online learning approach based on drifting GPR which are shown to be on par with other state of the art methods for learning inverse dynamics. Related to this, in Wang et al. (2015) the authors use GPR for constructing a feedforward control of a robot performing laser cutting and showing that the path tracking accuracy is improved.

In addition to previously described approaches, learning feed-forward control (LFFC) in Velthuis (2000), is another method of applying feed-forward control as an addition to a feedback controlled system. The method is similar to performing system identification, however instead of learning based on the output of the system, LFFC learns by minimizing the error of the control signals. In Velthuis (2000) two different types of neural networks are used as function approximators, Multi Layer Perceptron (MLP) network and B-spline neural networks (BSN). In Cuong and Minh (2015) BSN is applied as feed-forward control to a two-link rigid robot arm, which increases accuracy of evaluated using random motions performed by the robot.

This paper is based on the work presented in Johansson (2017) with emphasis on the experiments carried out using an experimental six degrees-of-freedom experimental robot. Due to space limitations only selected parts of the results presented in Johansson (2017) are included here. The paper is organised as follows. Section 2 gives a brief presentation of the type of robot considered in the paper and the control system in operation. In Section 3 the friction models to be studied are presented, and next Section 4 discusses how these models can be used for friction compensation. Section 5, which is the main contribution of the paper, presents some selected results from the experiments. Finally Section 6 provides some conclusions.

2. ROBOT MODELING AND CONTROL

The paper considers control of industrial robots of serial type as illustrated in Figure 1. Neglecting mechanical flexibilities the motion of the robot is described by the equations

\[ M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + \tau_f(\dot{q}) = \tau \]

where \( q \) is the column vector of the \( n \) joint angles. Furthermore \( M(q) \) is the matrix of inertia while \( C(q, \dot{q})\dot{q} \) represents the vector of Coriolis and centrifugal terms, \( g(q) \) is a vector of gravity torques, and \( \tau \) is the vector of joint torques. In addition \( \tau_f(\dot{q}) \) denotes the friction torque vector, where every element is a function of the respective angular velocity.

Fig. 1. Six degrees-of freedom industrial robot manipulator.

It is assumed that the robot is controlled using a combination of feed-back and feed-forward control according to Figure 2.

Fig. 2. Control system based on feed-forward (FF) and feed-back (F) control. \( G \) denotes the system to be controlled.

3. FRICTION MODELLING

Friction is a complex phenomenon and it has been studied in numerous publications. See for example Armstrong-Hélouvry (1991), Al-Bender and Swevers (2008), Harnoy et al. (2008), Stotsky (2007), and Bittencourt and Gunnarsson (2012). The phenomenon comes from the interaction of surfaces on a microscopic level, and depending on properties of the surfaces, such as the material and the level of surface finishing in the machinery method, friction effects will vary. In general the effects of friction cause a tangential force to the motion, which acts as a type of resistance to the system in question. The behaviour of friction depends on several physical properties, such as temperature, velocity, lubrication, and load, which will affect the resistance in different ways. In robot manipulator applications where high precision is required, friction is known to cause problems, especially for low-velocity motions. Friction can appear distributed in the system but a major part can be related to the motors and gearboxes.

3.1 The LuGre friction model

One model which has been extensively used in robot applications is the LuGre model, Åstrom and de Wit
function. For constant velocities the LuGre model also has

\[ f(v) = f_c + (f_s - f_c)e^{-(v/v_s)^2} \]  

(3)

where \( v \) is the relative velocity of the contact surfaces and \( z \) is the internal state which contributes to the dynamic behaviour of the friction. The function \( f(v) \) is the viscous friction which is most commonly given as a linear function. \( f(v) = f_v v \). \( g(v) \) captures the Coulomb friction and Stribeck effect where the parameters \( f_c \) is the Coloumb friction force, \( f_s \) the stiction force and \( v_s \) the Stribeck velocity.

Furthermore, as stated in Åstrom and de Wit (2008) for small displacements, the LuGre model produces a spring-like behavior through the parameters \( \sigma_0 \) and \( \sigma_1 \), which correspond to the spring and dampening constant respectively. \( F_d \) is then the friction force output of the function. For constant velocities the LuGre model also has a static representation, namely

\[ F_s(v) = g(v) \text{sign}(v) + f(v) \]  

(5)

where \( g(v) \) is given from (3). Friction depends on temperature but this is not included in the LuGre model.

### 3.2 Black-box friction model

While the LuGre model is based on physical insight an alternative approach is to formulate a friction model of black-box type and learn the parameters from data. The approach in this paper is based on B-spline networks (BSN). A B-spline network (BSN) is a network based on B-splines and weights.

The weights of the network are updated through the minimization of a cost function, which in the off-line case is given by

\[ J = \frac{1}{2} \sum_k (y_k^m - y(x_k))^2 \]  

(7)

where \( y_k^m \) and \( x_k \) represents measured data from the system to learn and \( y(x_k) \) is the output of the function given in (6).

The weights in the network, as presented in Figure 3 are updated through back-propagation. This is performed by taking the gradient with respect to the weights and adapting the weights with the corresponding value. In the offline case this gives the weight update

\[ \Delta w_i = \gamma \sum_k (y_k^m - y(x_k)) \mu_i^{(j)}(x_k) \]  

(8)

where \( \gamma \) is the learning rate, which is chosen in the interval \([0, 1]\) and \( x \) is the input to the system.

For the offline case Velthuis (2000) also proposes to normalize the weight update in order to prevent large weight adaptations. This is done in the following manner

\[ \Delta w_i = \gamma \frac{\sum_k (y_k^m - y(x_k)) \mu_i^{(j)}(x_k)}{\sum_k \mu_i^{(j)}(x_k)} \]  

(9)

where the offline adaptions are divided by the sum of all outputs of the given B-spline. By performing a normalization the effect of infrequent large errors become less influential, which helps the learning of the general behaviour.

### 4. FRICTION COMPENSATION

The friction compensation methods that will be evaluated and compared in this paper are based on feed-forward, which means that the friction estimate will be generated using the reference angular velocities, \( \dot{\theta}^{ref} \), for each of the joints. The result will hence depend on how well the overall robot control system works and how close the actual angular velocities are to the reference values.

#### 4.1 Friction compensation using the LuGre model

For the friction compensation based on the LuGre model, in (2), (3), and (4) the control structure is described in Figure 2, where the friction compensation is part the feed-forward block. In order to use the LuGre model for friction compensation it is necessary to have estimates of the different parameters in the model. In the experiments presented
in Section 5 these parameters are provided from previous identification experiments. To identify the friction parameters is in general a difficult and time consuming problem, and a brief summary of methods and references is given in Johansson (2017). The success of applying the LuGre based feed-forward friction compensation requires that the model itself gives a reasonable description of the friction phenomenon, that the parameter estimates are sufficiently accurate, and the joint angular velocities are close to the reference values.

4.2 Friction compensation using BSN

For clarity the BSN based feed-forward friction compensation will be represented by a separate block, and the overall structure of the control system is given in Figure 4. The variables $z$ and $v$ represent signals from the robot and the regulator used in the BSN and the learning procedure.

![Control system diagram]

Fig. 4. Control system based on feed-forward (FF) and feed-back (F) control extended with learning-based feed-forward.

In the application here the general equation for the update of the weights in (9), will use

$$\Delta w_{i,j} = \gamma \sum_{k=1}^{N_s} u_{i,p}(k) \mu((q^r_{i})(k)) - \sum_{k=1}^{N_s} \mu((q^r_{i})(k))$$

where $i$ denotes the joint and $j$ denotes the specific B-spline in the function. $N_s$ is the number of samples that were recorded during the execution of the system, $q^r_{i}$ is the velocity reference signal and $u_{i,p}$ the recorded training data on axis $i$. The variable $u_{p}$ is defined as $u_{p} = u_{c} - u_{f}$, where $u_{c}$ is the controller output, see Figure 4, and $u_{f}$ is the integral part of the controller output. The true friction force $F_F$ acting on the robot cannot be measured, and it is therefore replaced by an estimate. The argument is that if the feed-forward is based on an sufficiently accurate model of the robot, the task of the controller output is to counteract the friction force and to compensate for gravitational forces. By subtracting the integral part $u_{f}$ handling the gravitational forces, the remaining part of the input, i.e. $u_{p}$ represents an estimate of the friction force.

5. EXPERIMENTS

The aim of this section is to illustrate that the learning-based black-box approach gives an improvement of the accuracy which is equal, or close to, the improvement that is obtained by using the LuGre-model with carefully estimated parameters.

5.1 Experimental conditions

There are many parameters that can be varied in the experiments, and due to space limitations only a subset of the results presented in Johansson (2017) can be presented. The setup of the experiments was as follows:

- The experiments were carried out using a six degrees-of-freedom experimental robot.
- The tests were done in four operating points located in the corners of a square of the size 40 × 40 cm. See Figure 5.
- At each operating point the robot carried out circular motions with radii 1, 3 and 5 mm respectively.
- The movements were done with TCP velocities 10, 40 and 100 mm/s respectively. The presentation in the paper will be limited to operation point one and TCP velocity 40 mm/s.
- The learning process, i.e. updating of the BSN parameters was run for ten iterations, and the update weight was selected $\gamma = 0.95$ throughout the experiments.
- Using measurements from the real robot a suitable number and distribution of the knots in the BSN was selected. The distribution was not uniform, instead a more dense distribution was used for low velocities. More details are given in Johansson (2017).
- The learning procedure was carried out for each of the three selected TCP velocities.
- In addition to the BSN presented above an extended version was tested, where two different networks were trained and used depending on the sign of the angular acceleration. These are denoted extended/dynamic BSN in Section 5.

The accuracy is evaluated using the the root mean square (RMS) of the error,

$$RMS = \sqrt{\frac{1}{N_s} \sum_{k=1}^{N_s} e(k)^2}$$

where $e(k)$ is the error between the reference path and the recorded path, $N_s$ is the number of sampled points. The true TCP is measured using a laser tracker system.

5.2 Feed-forward friction compensation using the LuGre model

Tables 1 - 3 show the maximum deviation of the TCP from the reference circle in the four positions and for the three radii. Except for the smallest circle there is a substantial improvement using the feed-forward. As noted above there is no obvious improvement using the dynamic LuGre model.

Table 1. Max deviation of TCP without feed-forward.

| Circ. r1 | Pos. 1 | Pos. 2 | Pos. 3 | Pos. 4 |
|----------|--------|--------|--------|--------|
| 0.5      | 0.53   | 0.31   | 0.53   |
| 0.41     | 0.35   | 0.38   | 0.46   |
| 0.31     | 0.28   | 0.35   | 0.36   |
Fig. 5. The friction compensation methods were evaluated around four different operation points in the workspace. The operating points are shown in the figure. At each of the four operating points circular movements with radii of 1 mm, 3 mm, and 5 mm respectively were carried out. The results presented in the paper are from operating point one.

Table 2. Max deviation of TCP with static LuGre.

| Pos. | Circ. r1 | Circ. r3 | Circ. r5 |
|------|----------|----------|----------|
| 1    | 0.48     | 0.2      | 0.14     |
| 2    | 0.52     | 0.3      | 0.25     |
| 3    | 0.93     | 0.32     | 0.18     |
| 4    | 0.59     | 0.23     | 0.2      |

Table 3. Max deviation of TCP with dynamic LuGre.

| Pos. | Circ. r1 | Circ. r3 | Circ. r5 |
|------|----------|----------|----------|
| 1    | 0.55     | 0.17     | 0.15     |
| 2    | 0.58     | 0.24     | 0.2      |
| 3    | 0.89     | 0.31     | 0.17     |
| 4    | 0.65     | 0.24     | 0.21     |

Similar observations to the ones above can be made using Tables 4 - 6 showing the RMS values of the TCP error for the four positions and three circles. There is a substantial reduction by the friction compensation, but only a minor difference between the static and the dynamic LuGre model.

Table 4. RMS of TCP without feed-forward.

| Pos. | Circ. r1 | Circ. r3 | Circ. r5 |
|------|----------|----------|----------|
| 1    | 0.362    | 0.187    | 0.12     |
| 2    | 0.324    | 0.188    | 0.141    |
| 3    | 0.41     | 0.206    | 0.14     |
| 4    | 0.343    | 0.209    | 0.147    |

Table 5. RMS of TCP with static LuGre.

| Pos. | Circ. r1 | Circ. r3 | Circ. r5 |
|------|----------|----------|----------|
| 1    | 0.307    | 0.109    | 0.066    |
| 2    | 0.263    | 0.133    | 0.115    |
| 3    | 0.378    | 0.133    | 0.084    |
| 4    | 0.28     | 0.117    | 0.083    |

Table 6. RMS of TCP with dynamic LuGre.

| Pos. | Circ. r1 | Circ. r3 | Circ. r5 |
|------|----------|----------|----------|
| 1    | 0.309    | 0.092    | 0.063    |
| 2    | 0.281    | 0.134    | 0.105    |
| 3    | 0.374    | 0.141    | 0.075    |
| 4    | 0.289    | 0.111    | 0.077    |

5.3 Feed-forward friction compensation using BSN

Tables 7 - 9 show the maximum deviation of the TCP from the reference circle in all four positions and for all three radii. Except for the smallest circle there is a substantial improvement using the feed-forward. As noted above there is no obvious improvement using the extended BSN. Similar observations can be made in Tables 10 - 12 showing the RMS of the TCP error.

Table 7. Max deviation of TCP without feed-forward.

| Pos. | Circ. r1 | Circ. r3 | Circ. r5 |
|------|----------|----------|----------|
| 1    | 0.5      | 0.41     | 0.31     |
| 2    | 0.63     | 0.35     | 0.28     |
| 3    | 0.91     | 0.38     | 0.35     |
| 4    | 0.58     | 0.46     | 0.36     |

Table 8. Max deviation of TCP with static BSN.

| Pos. | Circ. r1 | Circ. r3 | Circ. r5 |
|------|----------|----------|----------|
| 1    | 0.47     | 0.17     | 0.16     |
| 2    | 0.61     | 0.3     | 0.25     |
| 3    | 0.86     | 0.2     | 0.21     |
| 4    | 0.53     | 0.4     | 0.18     |

Table 9. Max deviation of TCP with extended BSN.

| Pos. | Circ. r1 | Circ. r3 | Circ. r5 |
|------|----------|----------|----------|
| 1    | 0.52     | 0.19     | 0.16     |
| 2    | 0.6       | 0.28     | 0.25     |
| 3    | 0.86     | 0.4      | 0.21     |
| 4    | 0.57     | 0.37     | 0.18     |

Table 10. RMS of TCP without feed-forward.

| Pos. | Circ. r1 | Circ. r3 | Circ. r5 |
|------|----------|----------|----------|
| 1    | 0.362    | 0.187    | 0.12     |
| 2    | 0.324    | 0.188    | 0.141    |
| 3    | 0.41     | 0.206    | 0.14     |
| 4    | 0.343    | 0.209    | 0.147    |

Table 11. RMS of TCP with static BSN.

| Pos. | Circ. r1 | Circ. r3 | Circ. r5 |
|------|----------|----------|----------|
| 1    | 0.263    | 0.187    | 0.12     |
| 2    | 0.264    | 0.188    | 0.141    |
| 3    | 0.377    | 0.206    | 0.14     |
| 4    | 0.25     | 0.209    | 0.147    |

Table 12. RMS of TCP with extended BSN.

| Pos. | Circ. r1 | Circ. r3 | Circ. r5 |
|------|----------|----------|----------|
| 1    | 0.273    | 0.187    | 0.12     |
| 2    | 0.269    | 0.206    | 0.141    |
| 3    | 0.383    | 0.209    | 0.14     |
| 4    | 0.257    | 0.209    | 0.147    |

6. CONCLUSIONS

A comparison of two feed-forward based friction compensation methods has been presented. The first method is based on the LuGre friction model, using identified friction model parameters, and the second method is based on B-spline network, where the network weights are learned from experiments. The methods have been evaluated and...
compared via experiments using an experimental six axis industrial robot carrying out circular movements of different radii. The experiments have shown that the learning-based friction compensation gives an error reduction of the same magnitude as for the LuGre-based friction compensation.

The two approaches are based on different types of friction models, and it is difficult to draw general conclusions about the ease-of-use of the different approaches. The LuGre model is of grey-box type and it is a non-trivial and time consuming task to obtain good estimates of the various parameters in the model, and this sometimes requires specially designed experiments. The approach based on the BSN-model includes a number of design variables, like the number and type of splines, the location of the knots, the learning gain, etc, and it requires insight to choose these in a suitable way. More research and experiments are needed in order to get a clear picture of the trade-off between achievable error reduction and the necessary efforts by the user to tune the algorithms.

ACKNOWLEDGEMENTS

The work was partly supported by the VINNOVA Competence Center LINK-SIC (Linköping Center for Sensor Informatics and Control).

REFERENCES

Al-Bender, F. and Swevers, J. (2008). Characterization of friction force dynamics. Control Systems Magazine, 26(6).

Armstrong-Hélouvry, B. (1991). Control of Machines with Friction. Springer.

Bittecuor, A. and Gunnarsson, S. (2012). Static friction in a robot joint - modeling and identification of load and temperature effects. Journal of Dynamic Systems, Measurements, and Control, 135(4).

Camoriano, R., Traversaro, S., Rosasco, L., Metta, G., and Nori, F. (2016). Incremental semiparametric inverse dynamics learning. In 2016 IEEE International Conference on Robotics and Automation (ICRA), 544–550.

Cuong, N.D. and Minh, T.X. (2015). Learning feed-forward control for a two-link rigid robot arm. International Journal of Electronics and Electrical Engineering, 3(4), 279–284.

Harnoy, A., Friedland, B., and Cohn, S. (2008). Modeling and measuring friction effects. IEEE Control Systems Magazine, 28(6).

Johansson, V. (2017). Learning Local Models of Multi-axis Robot for Improved Feed-Forward Control. Technical report, Master’s Thesis 2017: EX029, Department of Signals and Systems, Chalmers University of Technology, Gothenburg, Sweden.

Ljung, L. (1999). System identification: theory for the user. Prentice Hall, Upper Saddle River, N.J, 2nd edition.

Meier, F. and Schaal, S. (2016). Drifting gaussian processes with varying neighborhood sizes for online model learning. In 2016 IEEE International Conference on Robotics and Automation (ICRA), 264–269.

Siciliano, B. and Khatib, O. (2016). Springer Handbook of Robotics. Springer International Publishing, Cham, 2nd edition.

Stotsky, A. (2007). Adaptive estimation of the engine friction torque. European Journal of Control, 13, 618–624.

Åstrom, K.J. and de Wit, C.C. (2008). Revisiting the lugre friction model. IEEE Control Systems, 28(6), 101–114. doi:10.1109/MCS.2008.929425.

Velthuis, W. (2000). Learning feed-forward control - theory, design and applications. Ph.D. thesis, University of Twente, Enschede, The Netherlands.

Wang, C., Zhao, Y., Chen, Y., and Tomizuka, M. (2015). Nonparametric statistical learning control of robot manipulators for trajectory or contour tracking. Robotics and Computer-Integrated Manufacturing, 35, 96 – 103.

Yin, S., Li, X., Gao, H., and Kaynak, O. (2015). Data-based techniques focused on modern industry: An overview. IEEE Transactions on Industrial Electronics, 62(1), 657–667.