Feedforward control for a manipulator with flexure joints using a Lagrangian Neural Network

Eline Heerze, Ronald Aarts, Bojana Rosic

University of Twemte, Applied Mechanics and Data Analysis, Enschede, The Netherlands
E-mail: R.G.K.M.Aarts@utwente.nl, B.Rosic@utwente.nl

Abstract: *Feedforward control of a manipulator can be generated with a sufficiently accurate stable inverse model of the manipulator. A Feedforward Neural Network (FNN) can be trained with experimental data to generate feedforward control without knowledge about the system at hand. However, the FNN output can show unphysical behaviour especially in operational regimes where the training data is sparse. We consider including a Lagrangian Neural Network (LNN) that is expected to predict the (inverse) multibody system behaviour more robustly.*

Introduction

Feedforward control can greatly improve the accuracy of a manipulator. In a typical implementation an inverse dynamic model of the multibody system at hand is used to compute the control input from the reference trajectory. The achievable performance gain from this feedforward depends heavily on the correctness and completeness of the model. In a *model-based* approach a *white-box* model with the equations of motion of the multibody system is derived and its parameters are estimated. Assuming these parameters have a clear physical meaning, it is expected that the model can be used for a wide variety of trajectories. However, the "richness" of the model is obviously restricted to the features included in the model structure. Alternatively, in a *data-driven* approach a *black-box* model is identified purely from data e.g. using machine learning techniques. A Feedforward Neural Network (FNN) [1] doesn't require any knowledge about the system dynamics and its parameters. However, care has to be taken to avoid overfit and incorrect model outputs are likely for operating conditions that were not sufficiently included in the training data. In this paper we augment the black-box FNN model with a Lagrangian Neural Network (LNN) [3] to improve the robustness of feedforward control.

Method

We consider the manipulator with two degrees of freedom (DOF) shown in figure 1 [2]. Two actuators drive the rotation of two arms resulting in a translational end-effector motion in a horizontal plane. It is equipped with flexure joints allowing rather smooth operation with low

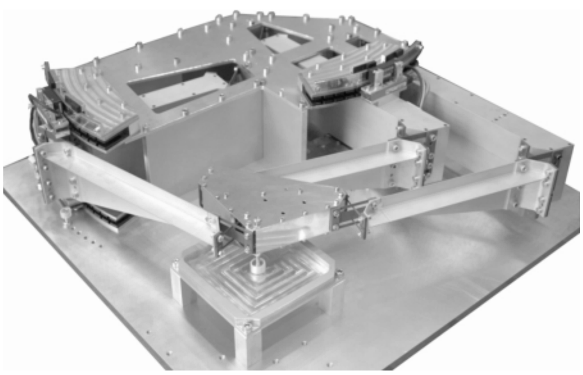

Figure 1: The 2 DOF manipulator with flexure joints [2] (photo by Ger Folkersma).

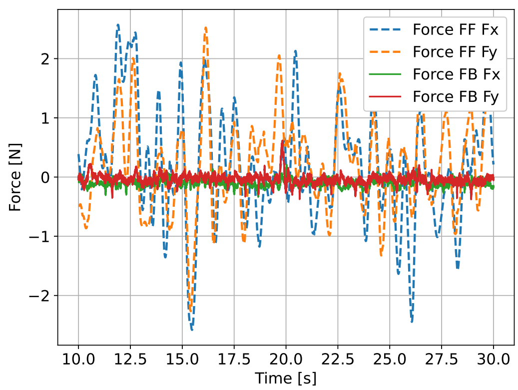

Figure 2: Feedback (FB) and feedforward (FF) controller inputs.

friction and hysteresis. Consequently, contributions from the link mass and joint stiffness dominate in the non-linear equation of motion written in independent generalised coordinates $\boldsymbol{q}$ as

$$\boldsymbol{F} = \boldsymbol{F}_{\mathrm{c}} + \boldsymbol{F}_{\mathrm{nc}} = \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \frac{1}{2}\dot{\boldsymbol{q}}^T \frac{\partial \boldsymbol{M}(\boldsymbol{q})}{\partial \boldsymbol{q}}\dot{\boldsymbol{q}} + \frac{\partial V(\boldsymbol{q})}{\partial \boldsymbol{q}} + \boldsymbol{F}_{\mathrm{nc}}, \tag{1}$$

where $\boldsymbol{F}_{\mathrm{c}}$ represents the conservative part in the total actuator force vector $\boldsymbol{F}$. It is expressed in the symmetric and positive definite mass matrix $\boldsymbol{M}$ and potential energy V, the latter due to the stiffness in all joints. The non-conservative forces $\boldsymbol{F}_{\mathrm{nc}}$ describe all remaining, possibly non-linear effects like cogging and damping due to the wiring of the actuators and sensors.
With an FNN it would be possible to learn the total forces $\boldsymbol{F}$ from trajectory data $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$, but then the physical structure of $\boldsymbol{F}_{\mathrm{c}}$ isn't taken into account. Hence a LNN [3] is used for this part of the feedforward. Its outputs are estimates of the non-negative potential energy $\hat{V}$ and a lower triangular matrix $\hat{\boldsymbol{L}}$ that is split into off-diagonal $\hat{\boldsymbol{l}}_o$ and positive diagonal $\hat{\boldsymbol{l}}_d$ terms. All estimates are functions of the positions $\boldsymbol{q}$ only and are substituted in Eq. (1) to compute $\boldsymbol{F}_{\mathrm{c}}$ with mass matrix $\hat{\boldsymbol{M}} = \hat{\boldsymbol{L}}\hat{\boldsymbol{L}}^T$. The remaining forces $\boldsymbol{F}_{\mathrm{nc}}$ are estimated with a general FNN, where it is assumed they only depend on positions $\boldsymbol{q}$ and velocities $\dot{\boldsymbol{q}}$.
Various options are considered in literature to train the combined networks, trying to avoid that the FNN also learns part of the physical behaviour that should be in the LNN. We concluded from simulations that training the LNN first on the full $\boldsymbol{F}$ results in a sufficiently accurate estimate of $\boldsymbol{F}_{\mathrm{c}}$. Hence the LNN is trained first and next the FNN is trained on the residual.

Results and conclusion

The LNN and FNN are trained with 9 datasets in which a reference path for the end-effector is defined with multisine signals. The frequency content of these signals differs such that some datasets excite in particular the stiffness properties, i.e. the potential energy $V(\boldsymbol{q})$, whereas other datasets reveal more mass dominated dynamics. Forces $\boldsymbol{F}$ and positions $\boldsymbol{q}$ are measured from which velocities $\dot{\boldsymbol{q}}$ and accelerations $\ddot{\boldsymbol{q}}$ are derived with numerical differentiation and low-pass filtering. The LNN has 8 hidden layers with 32 neurons each and can be trained with a relatively small dataset of about 20,000 samples (including validation data). For the FNN with 2 hidden layers with 8 neurons each, 10 times more samples are used. Both in simulations and experiments the forces $\boldsymbol{F}$ can be estimated with a mean absolute error (MAE) below 2%. The forces estimated by the combined LNN and FNN are added as feedforward control in a real-time closed-loop experiment. Figure 2 shows an example of these feedforward signals (FF) in comparison with the remaining feedback signals (FB). Clearly, the estimated feedforward accounts for by far the largest part of the total forces required for the specified motion. Consequently, also the tracking accuracy is improved considerably.
These results demonstrate that the feedforward control as generated by the successively trained LNN and FNN can applied successfully for the considered manipulator. The robustness and more general applicability of this approach will be investigated in the future.

References

[1] Abdul-hadi, O.: *Machine learning applications to robot control.* PhD thesis, University of California, Berkeley, 2018.

[2] Brouwer, D.M., Folkersma, K.G.P., Boer, S.E. and Aarts, R.G.K.M.: *Exact constraint design of a two-degree of freedom flexure-based mechanism.* Journal of mechanisms and robotics, Vol. 5(4), 041011 (10 pages), 2013.

[3] Lutter, M., Ritter, C. and Peters, J.: *Deep Lagrangian Networks: Using physics as model prior for Deep Learning.* 7th International Conference on Learning Representations (ICLR), 2019.