A Probabilistic Motion Control Approach for Teleoperated Construction Machinery

Hyung Joo Lee[a] and Sigrid Brell-Cokcan[a]

[a]Chair of Individualized Production (IP), RWTH Aachen University, Schinkelstr. 1 52062 Aachen Germany
E-mail: lee@ip.rwth-aacehen.de, brell-cokcan@ip.rwth-aacehen.de

Abstract –

Automation of construction machinery has the potential to improve efficiency and safety during the construction process. However, most construction machinery is directly teleoperated limiting the control to a single paradigm. Moreover, the abundance of non-linearities due to the design of pumps, valves and the interaction of different actuators complicates planning precise movements. Complicated motion planning optimization can be applied to program the movement in the desired manner which often requires a model representing the system dynamics. Whereas this approach is promising, modeling the system dynamics is a formidable task. In this work, we present a framework that uses a probabilistic approach to learn movements from expert demonstrations. In this way, efficient movements can be learned without explicitly estimating system dynamics. Here, efficiency is defined by the experiences of the human operator which makes the programming able to benefit from existing knowledge of operators. The performance of the proposed scheme is evaluated with a real demolition machine BROKK 170.

Keywords –

On-site robotics; construction robotics; robot motion programming; automation in construction

1 Introduction

Currently, remote-controlled hydraulically driven construction machinery is of great importance and mainly used on construction sites. However, local accuracy is often limited in remote-controlled systems due to the interface (for example 2D camera feedback). Moreover, it is challenging for the operator to generate optimized motions with a remote controller since the operator can only see the appropriateness of the motion only when it is already executed. As a result, manual work is still mostly preferred on construction sites. Sequentially, automating construction machinery has already drawn much attentions over the last years [1–5]. The common goal in automating construction machinery is to generate accurate motions in an automated way under various conditions (outdoor conditions, dirty and dynamic environments) to fulfill given tasks.

In particular, the automation of machinery can be regarded as controlling the end-effector in the desired manner. In automated machines, a precise movement is a key part, however, there are several difficulties in automating hydraulic construction machinery arising from the abundance of non-linearities due to the design of pumps, valves and fluid flow [6]. A common approach is the design of dynamics-based control, which typically uses a model for the system to be controlled to predict system states and develop a controller minimizing the discrepancy between the predicted and measured system states. Although this model-based control shows promising results in many researches [7, 8], modeling system dynamics can be often challenging due to its nonlinear characteristics. Moreover, force sensors required for force control are not standard components of construction machinery, since the system is designed to be directly controlled by human operators [9]. Hutter et al. [10] achieved joint torque control based on feedback from pressure sensors integrated into servo valve. However, it is common that pressure sensors are not installed in each valves but only in the main valve due to economic reasons [11].

On the other hand, with the advent of research advances in robotics, robotic systems with a large number of degrees of freedom were developed [12, 13]. Subsequently, Programming by Demonstration (PbD) has drawn the attention of many researchers [14–15], since it allows to program a robot just by showing the desired manner of performing tasks. In contrast to traditional motion planning methodologies, this approach offers an intuitive and less time consuming alternative for non-experts to teach a robot skill [16-18]. Popular approaches for encoding the demonstrations in a way that can be used for motion planning include Dynamic Movement Primitives (DMPs) and Gaussian Mixture Regression (GMR).

Our work concentrates on mitigating the aforementioned issues in automating construction machinery by utilizing the recent methodology from robotics. More precisely, we aim at replacing the time-consuming model-based motion programming of hydraulic construction machinery by an automatic programming process: Programming by Demonstration. In this paper, we present a framework that learns a direct

joint motion mapping from expert demonstrations. This learned mapping so-called policy allows to reproduce joint space motions under different environmental conditions while non-linearities in the hydraulic system are handled by utilizing experiences of the human expert in operating the system.

2 Previous work and challenges

In our previous work [19], we presented a teleoperated demolition machine BROKK 170 that has been retrofitted with controllers, so that one can also program its motion with high-level commands. Although the experiments from our previous work showed the feasibility that proven construction machinery can be adapted and accept advances from robotic to extend its capability, we identified that a Cartesian controller solely based on encoders is not sufficient to enable accurate arm control. As depicted in Figure 1, the machine was not able to follow the given trajectory without any error. The main error source is the unstable system pressure. The hydraulic machine BROKK 170 used in our work consists of one main pump supplying volumetric flow for the whole system. This flow is used in each control valves to move actuators and returned into the tank. During this circulation, flow losses are raised by partially closed or opened valves, bent pipes, expansions or contractions. A generic joint position level controller such as an Inverse Kinematics (IK) solver rapidly reaches its limits, since it typically takes the desired (x, y, z) values as input and outputs the corresponding joint configuration without considering the non-linearities in a hydraulic system.

We aim at overcoming the aforementioned issue by demonstrating basic movements to the machine. Using prior experience and trainings, expert human operators can create more controlled motion with the hydraulic system than IK-based motion controller. By capturing the operator's behavior to overcome the non-linearities of the system, we aim at improving the accuracy in planning of the motion.

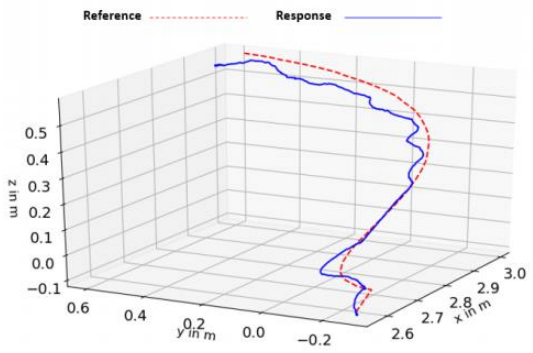

Figure 1. Tracking result using an IK solver based framework

3 Manipulator Control

The manipulator control can be formulated as finding proper joint space configurations $\boldsymbol{q} \in \mathbb{R}^N$ given an operational space description $\boldsymbol{x} \in \mathbb{R}^M$,where N is the number of degree-of-freedoms (DoF) of the machine and M is the operation dimension. As depicted in Figure 2, the operational space $\boldsymbol{x}$ and joint configuration $\boldsymbol{q}$ of the demolition machine can be described as:

$$\boldsymbol{x} = [x \quad y \quad z \quad \theta]^T \in \mathbb{R}^4 \tag{1}$$

$$\boldsymbol{q} = [q_1 \quad q_2 \quad q_3 \quad q_4 \quad q_5]^T \in \mathbb{R}^5 \tag{2}$$

The mapping problem can be formulated as:

$$\dot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{3}$$

where $\boldsymbol{J} \in \mathbb{R}^{M \times N}$is the Jacobian matrix. To obtain the $\dot{\boldsymbol{q}}$ out of the relation in (10), we use the closed loop inverse kinematic (CLIK) approach introduced in [20]. However, as described in the previous section 2, this analytical method does not consider non-linearities in a hydraulic system causing a discrepancy between the estimated and real motion. To address this problem, we employ a probabilistic approach based on Gaussian Process Regression (GPR) first to learn policies from expert demonstrations and program the manipulator by reproducing the learned policies under different environmental conditions.

3.1 Closed-loop inverse kinematic controller

For manipulators with N > M, the inverted Jacobian matrix can be obtained as follow:

$$\boldsymbol{J}^+ = \boldsymbol{J}^T(\boldsymbol{J}\boldsymbol{J}^T)^{-1} \tag{4}$$

The pseudo-inverse has the property to provide the best possible solution according to the equation $\boldsymbol{J}\dot{\boldsymbol{q}} = \dot{\boldsymbol{x}}$. If $\dot{\boldsymbol{x}}$ is not in the range of $\boldsymbol{J}$, an exact value of $\dot{\boldsymbol{q}}$ is not available. However, the provided $\dot{\boldsymbol{q}}$ still minimizes the magnitude difference of $\boldsymbol{J}\dot{\boldsymbol{q}} = \dot{\boldsymbol{x}}$ [21]. By using this property and closed-loop behavior in CLIK, the convergence to the desired $\dot{\boldsymbol{x}}$ can be ensured. The equation from (3) can be formulated as follows:

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^+(\dot{\boldsymbol{x}} + \boldsymbol{K}\boldsymbol{e}) \tag{5}$$

where $\boldsymbol{K} \in \mathbb{R}^{M \times M}$ is a positive definite gain matrix and $\boldsymbol{e} \in \mathbb{R}^{M \times 1}$ is the remained difference between the desired and actual motion. The CLIK controller is further extended with damped-least squares for achieving robustness in the vicinity of singularities [21] and with weighted least norms for joint limits [22].

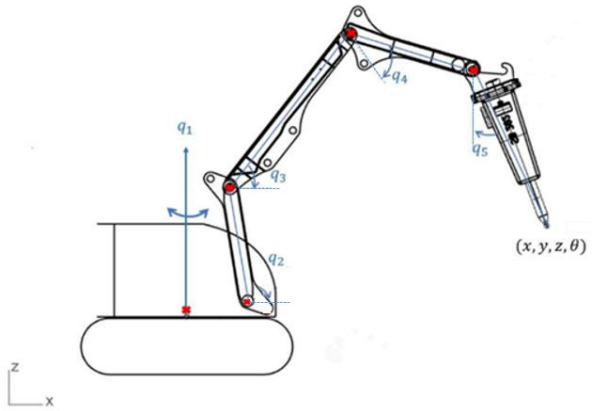

Figure 2. Joint geometric representation of the 5 degree of freedom BROKK 170. The joint space coordinates are noted in blue, whereas the operational space coordinates are in black. The joint positions are noted in red.

3.2 Gaussian Process Models

GPR is a non-parametric method for regression that models a joint distribution of the observed data without directly modeling a regression function[23]. By assuming the existence of m demonstration, each demonstration can be described as $D = \{\xi_i, y_i\}_{i=1}^{n}$. Here, ξ denotes the environmental information, such as the temporal value, y refers to the observations such as the trajectories in the joint space that ξ should be mapped to and n denotes the length of the demonstration.

In GPR, the distribution of observations can be modeled as follows:

$$\begin{bmatrix} y \\ y^* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X,X) & K(X,X^*) \\ K(X^*,X) & K(X^*,X^*) \end{bmatrix}\right) \quad (6)$$

where K is the kernel matrix defined as follows:

$$\begin{aligned} K(X,X) &\in \mathbb{R}^{M\times M}, & K_{i,j} &= k(\xi_i, \xi_j) \\ K(X^*,X) &\in \mathbb{R}^{1\times M}, & K_{1,j} &= k(\xi^*, \xi_j) \\ K(X,X^*) &\in \mathbb{R}^{M\times 1}, & K_{i,1} &= k(\xi_i, \xi^*) \\ K(X^*,X^*) &\in \mathbb{R}, & K &= k(\xi^*, \xi^*) \end{aligned} \quad (7)$$

A key parameter in GPR is the kernel function denoted as $k(\cdot,\cdot)$ in (2). It encodes the structure of functions in the space of distributions. In this work, the squared exponential kernel function is employed:

$$k(x_1, x_2) = \sigma_f^2 \, exp\left(-\frac{1}{2l^2}(x_1 - x_2)^T(x_1 - x_2)\right) \quad (8)$$

The goal of GPR is to predict the joint motion y^* conditioned on the new environmental information $\boldsymbol{\xi}^*$ and the observed information from the demonstrations. It is assumed that noises σ_n from process and measurement are identically distributed within demonstrations. In this case, the joint distribution of y^* can be described as:

$$y^*|y, D \sim \mathcal{N}\,(\mu^*, \Sigma^*) \quad (9)$$

where

$$\mu^* = K(X^*,X)(K(X,X) + \sigma_n^2 I)^{-1} y \quad (10)$$

$$\Sigma^* = K(X^*,X^*) + \sigma_n^2 I - K(X^*,X)(K(X,X) + \sigma_n^2 I)^{-1} K(X,X^*) \quad (11)$$

The performance of GPR depends on so-called hyper parameters [σ_n, σ_n, l] which can be defined by maximizing the marginal log likelihood [23]. We use GPflow to model GPR and also to estimate the parameters [24].

Using these equations, the proposed controller generates joint trajectories based on expert human demonstrations which indicate the operator maneuvers we would like to apply to the manipulator.

3.3 Programming by Demonstration Framework

During demonstrations of different lengths, the sensor information is collected and stored as training input-output pairs. The mapping from the input to the output is called a policy and is modeled from equations (6) - (11), so that the model can predict a new output based under different input conditions based on the learned policy. In our work, multiple demonstrations of different lengths are used to model the policy. Accordingly, the time alignments of the demonstrated trajectories are normalized using Dynamic Time Warping (DTW) method [25].

Suppose we have m demonstrations with BROKK 170, then the demonstrations are normalized to the length T. The data set D for the GP will be of length $n = mT$. Thus, the data set can be formulated as $D = \{\xi_i, y_i \} \mid i = 0, \ldots, mT$.

In our training phase, policies are created with respect to the initial joint states q and the temporal value. In our experiments, we use simple timestamp t as the temporal value. More precisely, joint angle values are collected and stored during an expert demonstration, where the expert controls each joint to move the end-effector tip straight forward. Policies are created with respect to these collected values so that new trajectories in the joint space can be generated for the new initial joint angle values and the timestamps.

In the reproduction phase, a new initial starting point is defined with joint angle values. Together with the timestamps, the initial joint angle values are used as input for the trained Gaussian process models.

In the evaluation phase, the predicted trajectories are directly tracked with a fixed maximum joint speed. The

resulting end-effector trajectory is obtained by computing the forward kinematics relationship and compared with the end-effector trajectory generated by the CLIK solver.

4 Experiment

We demonstrate our framework on a hydraulic demolition machine BROKK 170. Originally, the BROKK 170 demolition machine is designed to be controlled with a remote controller in a strict master-slave relationship limiting the control to a single paradigm. To control the machine with computation algorithms and integrate computational capabilities in the machine control, we follow the approach introduced in [19].

Our host PC interacts with the main control unit (MCU) of the machine through a bus controller MCP2515 and an embedded controller Mega2560. The communication takes place via the Controller Area Network (CAN) bus system. The bus protocol containing information about bus messages is obtained from the manufacturer. The script is developed in C++ on the host PC and then compiled and deployed to the embedded controller via Arduino IDE.

In this work, the kinematics of the machine and the shape of each axes are assumed to be known, so that the joint angles are enough to describe the full configuration of the machine, in particular the end-effector pose. Like most other construction machines, BROKK 170 used in this work is not equipped with any sensors. We use wire-type encoders (BCG05, SICK) to measure the joint angle values q_2, q_3, q_4, q_5 and a rotary encoder (A3M60, SICK) to measure q_1. The joint angle values are separately measured and sent to the host PC via Wi-Fi-Modules ESP8266 at a rate of 20 Hz. UDP packets are used for data exchange between the Arduino IDE and the proposed controller developed in Python.

The experiment consists on moving the manipulator straight forward from the initial point which requires structured joints movement. Moreover, the non-linearities of the hydraulic system complicate the task, so that trajectories generated by a conventional IK solver result in poor quality, as depicted in Figure 4. Throughout the experiments, the base q_1 is not used, since the trajectories for the given task lie within the xz-plane. The second axis q_2 is also not utilized in our experiment, since it cannot be simultaneously moved with other axes in the current setup.

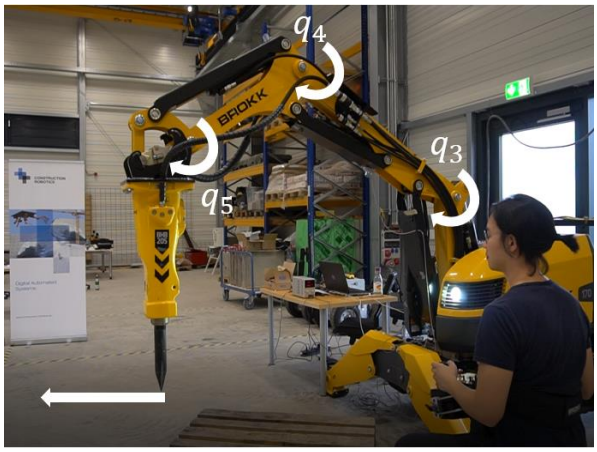

Figure 3. The task considered in this work: By moving the joints q_3, q_4 and q_5 with joysticks the operator tries to end-effector tip straight forward.

For the given task, the environment conditions are the initial joint positions and the timestamp, $\boldsymbol{\xi} = \{(q_{3\_o}, q_{4\_o}, q_{5\_o}, t_i) | i = 0, \dots, T\}$. The joint trajectories are used as observations, $y = \{(q_{3\_i}, q_{4\_i}, q_{5\_i}) | i = 0, \dots, T\}$. Three demonstrations are performed under various initial positions, as shown in Figure 4. Since the execution time for demonstrations slightly differs The demonstrations are of different lengths. We order these demonstrations according to their length:

$$D_{m-1} < D_m < D_{m+1} \tag{7}$$

Using the median length sequence $\bar{D} = D_{[2]}$ as a reference demonstration, the rest of the demonstrations are aligned in the time domain using DTW. The result of the DTW is illustrated in Figure 5. The thick trajectories represent the reference trajectories estimated as described above. It can be clearly seen that the temporal variations in the demonstrations are synchronized after the DTW process.

We perform trajectory planning with a new initial joint values $[q_{3\_o}, q_{4\_o}, q_{5\_o}] = [-0.547, 1.107, 1.016]$. The reproduction of the expert's demonstrations with respect to the new condition is visualized in Figure 6. It is worth noting that the step size for each joint trajectory is empirically estimated. The resulting end-effector trajectory is visualized in Figure 5. Since the effect from the non-linearities in the hydraulic system increases with the joint angle speed, we evaluate the developed system with two different joint angle speed: $\dot{q}_{max} = 0.11\ rad/s$ and $\dot{q}_{max} = 0.07\ rad/s$.

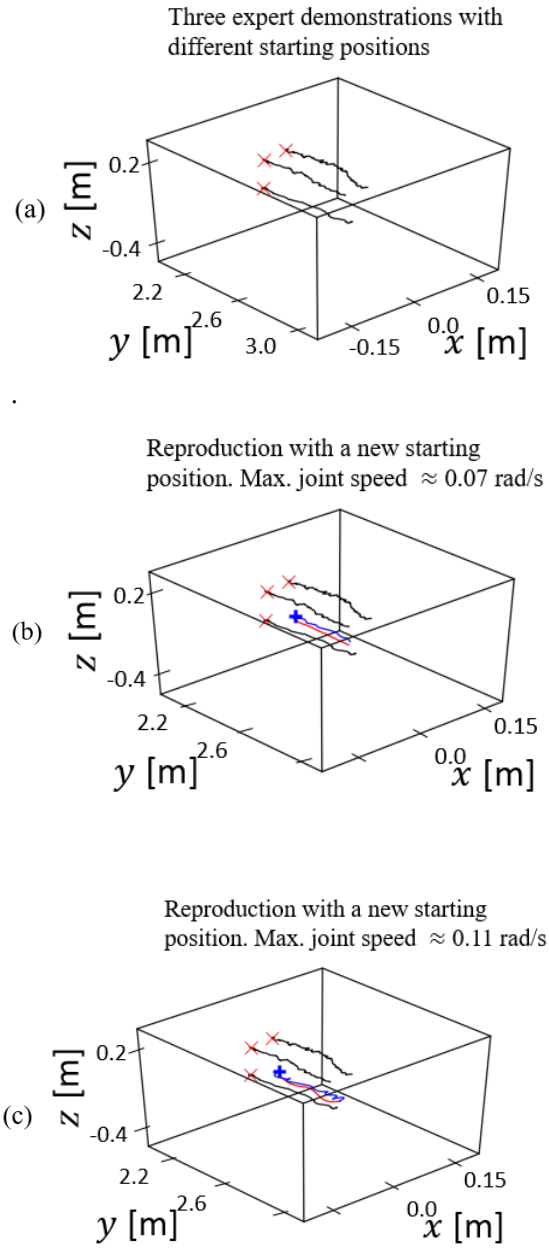

Figure 4. Three demonstrations for the given task in 3D Cartesian space (a). During the demonstrations different initial joints angles are considered. The resulting end-effector tip positions are visualized with 'x' signs. The demonstrations are reproduced for the new initial joints angles (blue '+' signs). The corresponding end-effector tip trajectories are visualized in (b) and (c) by computing the forward kinematic relationship (blue). The red trajectories are generated using the CLIK solver.

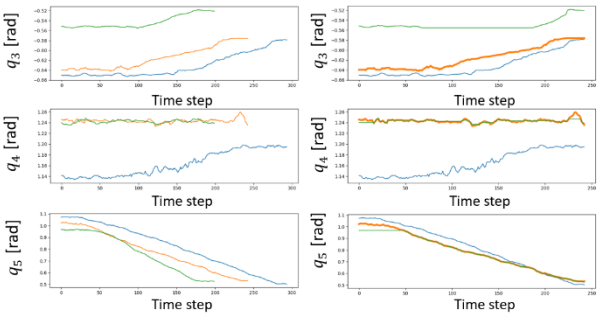

Figure 5. Results of collected trajectories in the joint space before (left) and after (right) the DTW process. The thick lines in the right plots represent the reference trajectories.

To quantify the correctness, first, the end-effector position is computed using the forward kinematic solution with the reproduced joint trajectories. Then, the desired end-effector trajectory (i.e. from the given initial position straight forward) is uniformly resampled with the total number of data points on the computed end-effector trajectory. Finally, the RMS error between the computed and planned end-effector trajectories in the operational space is computed.

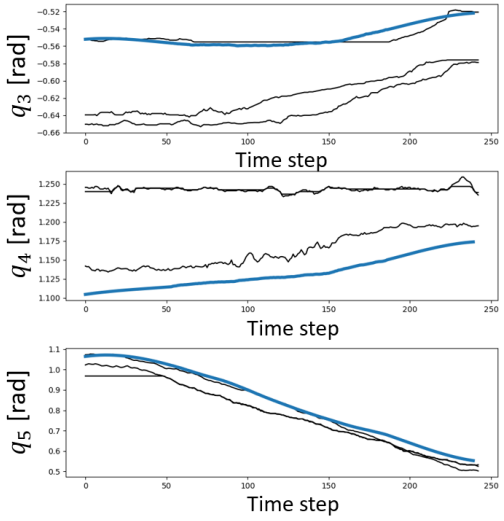

Figure 6. The black trajectories show the collected values from the demonstrations. The blue trajectories show the reproduction attempt by considering the new condition $[q_{3\_o}, q_{4\_o}, q_{5\_o}] = [-0.547, 1.107, 1.016]$. It can be clearly seen that the joint 5 q_5 is mainly used in the demonstration and this policy is preserved in the reproduced trajectories.

The same task is executed using the CLIK solver and the results are compared with the tracking results from the developed GPR based system. For the given task, the RMS errors of the CLIK solver are 0.027 m at the maximum joint angle speed of 0.07 rad/s and 0.039 m at

the maximum speed of 0.11 rad/s, respectively. Whereas the reproduced trajectories result in 0.016 m and 0.025 m, respectively.

Table 1. RMS error comparison in tracking task

	Max. Speed	RMSE
CLIK	0.07 rad/s	0.027 m
GPR	0.07 rad/s	0.016 m
CLIK	0.11 rad/s	0.039 m
GPR	0.11 rad/s	0.025 m

The end-effector tip trajectories generated by both the CLIK solver and the proposed system are visualized in Figure 7. We see that the trajectory from the CLIK solver deviates from the desired trajectory, whereas the deviation is smaller by the trajectory that is generated from expert demonstrations.

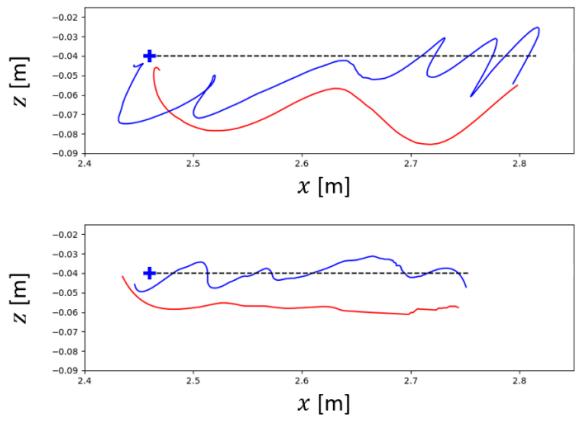

Figure 7. Results with both the reproduction from expert demonstration (blue) and from the CLIK solver (red). The desired trajectory is visualized with dashed line.

5 Discussion

In the experiments, the dimension of environmental conditions was set to four. Because of the low dimensionality, the policy of controlling a nonlinear hydraulic system could be modeled and reused to generate an appropriate end-effector trajectory using a relatively small number of expert demonstrations.

The proposed approach presents advantages by directly providing reference trajectories in the joint space. Compared to the inverse kinematic approach which typically generates joint motion without considering the non-linearities in a hydraulic system. As a result, the discrepancy in Figure 7 occurred which is increased according to the increased joint angle speed. During the demonstrations the expert corrects the joint motions according to the effect generated by the non-linearities in the hydraulic system. More precisely, the expert takes into account that he cannot perfectly move multiple axes at the same time and adapt the axes motion according to the resulting non-linear motion. By reproducing trajectories in the joint space considering this policy we slightly improve the tracking result. The captured policy that the joint 5 q_5 should be mainly utilized for the given task is valid for the faster joint angle speed, thus improving the tracking result as shown in Table 1.

However, as the GPR models the joint distribution of the observed data, the learned policy is only valid in the range of given demonstrations. Although the expert can demonstrate to cover the great area of workspace to reduce the uncertainty in the prediction caused by unforeseen environmental conditions. An efficient approach which can apply the existing demonstrations for the unforeseen conditions should be studied.

6 Conclusion

This study presented a framework for generating trajectories based on expert demonstrations. This work showed an alternative to programming a teleoperated construction machinery without a dynamical model. The experiment showed the preliminary performance of the proposed scheme. The expert demonstrations were normalized in the time domain using DTW and the mapping between the environmental conditions and the joint trajectories was modeled using GPR. By learning the mapping between the exact joint trajectories and the initial joint positions, the policy from a human expert could be learned so that a specific joint can be mainly utilized to perform the given task. The proposed scheme was implemented in a teleoperated deconstruction machinery, and the performance of the developed system was tested through preliminary experiments showing improved results compared to the inverse kinematic approach. Experimental results indicate the feasibility of construction machinery automation using generalizing the human demonstrations.

Future work will consider the use of Reinforcement Learning to equip the machine with self-improvement abilities. As shown in Figure 4, it was even for an experienced operator a challenging task to move the tip of the end-effector straight forward, since the operator can only see the appropriateness of its joystick manipulation after the machine has moved. We also aim at extending our framework to dynamic environmental conditions, since in this work only static conditions such as the initial joint angle values are considered.

Acknowledgements

This work is partially supported by the BROKK DA GmbH.

7 References

[1] D. Jud, G. Hottiger, P. Leemann, and M. Hutter, "Planning and Control for Autonomous Excavation," *IEEE Robot. Autom. Lett.*, vol. 2, no. 4, pp. 2151–2158, 2017.

[2] S. J. Keating, J. C. Leland, L. Cai, and N. Oxman, "Toward site-specific and self-sufficient robotic fabrication on architectural scales," *Sci. Robot.*, vol. 2, no. 5, eaam8986, 2017.

[3] A. Montazeri and J. Ekotuyo, "Development of dynamic model of a 7DOF hydraulically actuated tele-operated robot for decommissioning applications," in *2016 (ACC)*, Boston, MA, USA, pp. 1209–1214.

[4] S. Tafazoli, S. E. Salcudean, K. Hashtrudi-Zaad, and P. D. Lawrence, "Impedance control of a teleoperated excavator," *IEEE Trans. Contr. Syst. Technol.*, vol. 10, no. 3, pp. 355–367, 2002.

[5] D. Schmidt, M. Proetzsch, and K. Berns, "Simulation and control of an autonomous bucket excavator for landscaping tasks," in *2010 IEEE ICRA*, Anchorage, AK, pp. 5108–5113.

[6] S.-U. Lee and P. H. Chang, "Control of a heavy-duty robotic excavator using time delay control with switching action with integral sliding surface," in *Proceedings 200 IEEE ICRA*, Seoul, South Korea, pp. 3955–3960.

[7] M. M. Bech, T. O. Andersen, H. C. Pedersen, and L. Schmidt, "Experimental evaluation of control strategies for hydraulic servo robot," in *2013 IEEE International Conference on Mechatronics and Automation*, Kagawa, Japan, pp. 342–347.

[8] A. Bonchis, P. I. Corke, and D. C. Rye, "Experimental evaluation of position control methods for hydraulic systems," *IEEE Trans. Contr. Syst. Technol.*, vol. 10, no. 6, pp. 876–882, 2002.

[9] J. Mattila, J. Koivumaki, D. G. Caldwell, and C. Semini, "A Survey on Control of Hydraulic Robotic Manipulators With Projection to Future Trends," *IEEE/ASME Trans. Mechatron.*, vol. 22, no. 2, pp. 669–680, 2017.

[10] M. Hutter *et al.*, "Towards optimal force distribution for walking excavators," in *2015 (ICAR)*, Istanbul, Turkey,pp. 295–301.

[11] BROKK AB, "Brokk Troubleshooting Guide | Repair Maintain Troubleshoot Guide Brokk Demoltion Equipment,"

[12] M. Keppler, D. Lakatos, C. Ott, and A. Albu-Schaffer, "A passivity-based approach for trajectory tracking and link-side damping of compliantly actuated robots," in *2016 (ICRA)*, Stockholm, Sweden, pp. 1079–1086.

[13] B. Bauml *et al.*, "Catching flying balls and preparing coffee: Humanoid Rollin'Justin performs dynamic and sensitive tasks," in *2011 IEEE ICRA*, Shanghai, China, pp. 3443–3444.

[14] P. Abbeel, A. Coates, and A. Y. Ng, "Autonomous Helicopter Aerobatics through Apprenticeship Learning," *IJRR*, vol. 29, no. 13, pp. 1608–1639, 2010.

[15] T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, and J. Peters, "An Algorithmic Perspective on Imitation Learning," *FNT in Robotics*, vol. 7, 1-2, pp. 1–179, 2018.

[16] T. Osa, N. Sugita, and M. Mitsuishi, "Online Trajectory Planning in Dynamic Environments for Surgical Task Automation," in *RSS X*, Jul. 2014.

[17] S. Schaal, "Is imitation learning the route to humanoid robots?," *Trends in Cognitive Sciences*, vol. 3, no. 6, pp. 233–242, 1999.

[18] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: learning attractor models for motor behaviors," *Neural computation*, vol. 25, no. 2, pp. 328–373, 2013.

[19] H.J. Lee, S. Brell-Cokcan, K. Schmitz," A General Approach for Automating Teleoperated Construction Machines", *in 2019 Advances in Service and Industrial Robotics*, vol.980, pp.210-219.

[20] B. Dariush, M. Gienger, B. Jian, C. Goerick, and K. Fujimura, "Whole body humanoid control from human motion descriptors," in *2008 IEEE ICRA*, Pasadena, CA, USA, pp. 2677–2684.

[21] Y. Nakamura and H. Hanafusa, "Inverse Kinematic Solutions With Singularity Robustness for Robot Manipulator Control," *Journal of Dynamic Systems, Measurement, and Control*, vol. 108, no. 3, pp. 163–171, 1986.

[22] T. F. Chan and R. V. Dubey, "A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators," *IEEE Trans. Robot. Automat.*, vol. 11, no. 2, pp. 286–292, 1995.

[23] C. M. Bishop, *Pattern recognition and machine learning*, 8th ed. New York, NY: Springer, 2009.

[24] M. van der Wilk, V. Dutordoir, S. T. John, A. Artemev, V. Adam, and J. Hensman, "A Framework for Interdomain and Multioutput Gaussian Processes," Mar. 2020. [Online]. Available: http://arxiv.org/pdf/2003.01115v1

[25] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 26, no. 1, pp. 43–49, 1978.

[26] J. Lee, N. Mansard, and J. Park, "Intermediate Desired Value Approach for Task Transition of Robots in Kinematic Control," *IEEE Trans. Robot.*, vol. 28, no. 6, pp. 1260–1277, 2012.