Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

IOC Based Trajectory Generation to Increase Human Acceptance of Robot Motions in Collaborative Tasks

Eert Hoogerwerf * **Mukunda Bharatheesha** * **Debora Clever** **

* *Faculty of Mechanical, Maritime and Materials Engineering, Delft University of Technology, The Netherlands (affiliation, when working on the contribution)*
eert.hoogerwerf@gmail.com, mukunda1028@gmail.com
** *ABB AG, Research Center, Ladenburg, Germany*
debora.clever@de.abb.com

Abstract: Collaboration between humans and robots is an important aspect of Industry 4.0. It can be improved by incorporating human-like characteristics into robot motion planning. It is assumed that humans move optimal with respect to a certain objective or cost function. To find this function, also for a robot, we use an inverse optimal control approach identifying what linear weighted combination of physically interpretable cost functions best mimics human point-to-point motions. A bi-level optimization is used, where the upper level compares the optimal robot result of the lower level with human reference motions. Two depth cameras are combined in a setup to record these reference motions.
The resulting weighted cost functions are then used to generate new motions for a seven degrees of freedom robot arm. The resulting optimized motions are compared to standard robot motions based on linear interpolation in joint or task space. The comparison is performed by means of a small experiment where preliminary observations show that humans experience these motions as more anthropomorphic and feel at least equally comfortable and safe compared to existing motion planning strategies.

Keywords: Human-Robot-Collaboration, Path Planning, Optimal Trajectories, Inverse Optimal Control, Motion Analysis

1. INTRODUCTION

In an environment where humans and robots are working together a mutual understanding of each other is crucial. One way to improve this is by humanizing Human-Robot Interaction (HRI), which means that the robot not only tries to understand the human's actions and intentions, but also that it uses human-like features to communicate its own, see e.g. Sciutti et al. (2018). This makes sense because humans are highly trained in interacting with other humans.

Human and robot arm motion planning has to deal with a high level of redundancy and out of the infinite amount of possibilities it is believed that humans choose a path that is optimal with respect to a certain objective function, see e.g. Alexander (1996). This objective function is situation dependent and generally unknown. By inverse optimization using human measurements these objective functions can be determined. Applying optimal control with the found objective functions can enable human-like motion characteristics for robots.

1.1 Contribution

The contribution of this paper is on an inverse optimal control based motion generation approach based on physically interpretable objective functions for human-robot-collaboration and the comparison of the resulting motions to those from standard point to point motion generation approaches that are based on linear interpolation in joint or task space. The comparison is not done based on quantitative measures but by a small experiment, analyzing subjective acceptance of humans working together with a

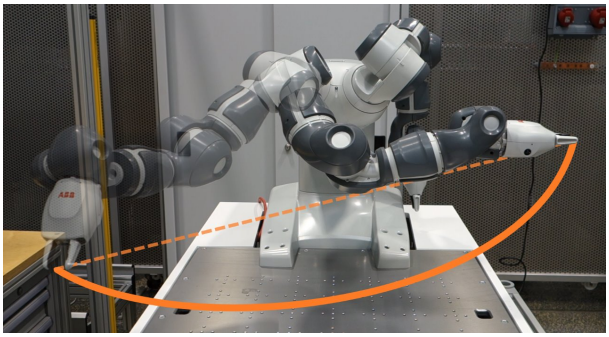

Fig. 1. The YuMi robot performing an optimal motion based on the cost function derived with inverse optimal control using human recorded motion. Point-to-point motions on this robot are currently performed by linearly interpolating a path in joint or in Cartesian space.

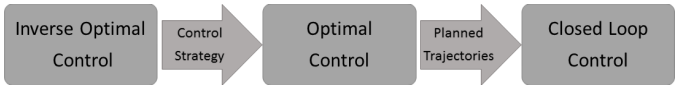

Fig. 2. Methodology. Human control objective is identified using inverse optimal control. Corresponding robot trajectories are generated using optimal control and executed on real robot.

robot. The robot we consider here is the ABB YuMi with seven degrees of freedom in each of the two arms.

The paper is organized as follows: In Chapter 2 the theory behind inverse optimal control using the bi-level approach is summarized, followed by our implementation in Chapter 3. The results are given in Chapter 4 in the form of a composite cost function, which is analyzed with an experiment to determine how humans experience the resulting motions.

1.2 Related work

Inverse optimal control has been used in several publications to either analyze human motion or to generate robot motion. A bi-level optimization approach was introduced in Mombaur et al. (2010) for finding the optimization criteria humans use for a locomotion tasks. A combination of five criteria, involving minimization of time, acceleration and orientation towards the target, gave the best results. In Clever et al. (2018) the approach was extended for the generation of gait trajectories in complex environments using an objective function identified from human captured walking motions. A similar bi-level approach was used for arm motions in Berret et al. (2011) for a two dimensional pointing task. The cost functions that had the biggest contributions where those that minimized energy consumption and angle acceleration. The same approach was later used for a more realistic three dimensional task by Sylla et al. (2014), where it was found that a geodesic and an energy criteria best mimicked human motion. Arm reaching motions in collaboration tasks were researched by Mainprice et al. (2015). Two participants performing a number of pick-and-place tasks while standing next to each other were recorded. For the inverse optimal control they use cost functions based on distances to the other participant and functions achieving smoothness. The trajectories planned with the recovered weights are found to better resemble human motion than with manually tuned weights. In Albrecht et al. (2011) a one-level optimization has been used and it is stated that a combination of torque change and joint jerk minimization generates humanoid robot reaching motions, which match human recorded motions closely. Also based on a one-level approach, task dependence of cost function was examined for a box moving task using time dependent weights in Englert et al. (2017).

Whereas in the cited literature, quality was rated by how well the generated motions imitate human motions, in our contribution, we rate the quality of generated motions by how humans working together with the robot perceive these motions in comparison to the standard motions. Different to Mainprice et al. (2015), here we study the suitability of various physically interpretable objective functions for human-motion-collaboration that go beyond smoothening and distance. In contrast to Clever et al. (2018), where the transferred optimization strategy includes a re-scaling of the identified weights, here we directly use the robot model within the identification step. Hence, we investigate the objective for the considered robot, when moving as close as possible to what has been captured in human recordings.

2. INVERSE OPTIMAL CONTROL

With optimal control a certain objective function is minimized to derive a local optimal solution. For inverse optimal control this objective function is to be determined using a known optimal solution. In the case of a robot arm this can also be a recorded motion of a human. The learned objective function can then be used to achieve human-like characteristics for reaching new targets. This generalization to new situations is an important feature of inverse optimal control.

Because real-time control is not an objective here and to have more flexibility in the choice of cost functions, a bi-level approach is chosen. The upper level optimizes parameters α_i based on the difference between the solution of the lower level and the recorded reference motion. The lower level solves a forward optimal control problem using the objective parameters supplied by the upper level.

2.1 Linear combination

One way of learning the objective function U can be by assuming that it is a linear combination of n cost functions U_i weighted by parameters α_i that do not change over time. The total cost is then

$$U = \int_0^T \Big[\sum_{i=1}^{n} \alpha_i U_i(\boldsymbol{z}(t), \boldsymbol{u}(t)) \Big] dt, \tag{1}$$

with $\boldsymbol{z}$ representing the states, $\boldsymbol{u}$ the controls and T the motion duration. The problem of finding the objective function is now reduced to finding the weights α_i. A weight can go to zero if its corresponding cost function does not contribute to the reference motion.

2.2 Upper level

The inverse optimal control problem can be split into two levels. The upper level optimization is used to determine the weight factors $\boldsymbol{\alpha} := [\alpha_1, \ldots, \alpha_n]$ using a least-squares minimization

$$\min_{\alpha} \sum_{j=1}^{m} ||v^*(\boldsymbol{\Phi}(\boldsymbol{\alpha}), t_j) - v_j^M||^2, \tag{2}$$

with the comparison criterion v_j^M of the physical measurements M_j, $j = 1, \ldots, m$ and the comparison criterion $v^*(\boldsymbol{\Phi}(\boldsymbol{\alpha}), t_j)$ of the optimization results $\boldsymbol{\Phi}(\boldsymbol{\alpha}) := (\boldsymbol{z}^{*,\boldsymbol{\alpha}}(t), \boldsymbol{u}^{*,\boldsymbol{\alpha}}(t), T^{*,\boldsymbol{\alpha}})$ for the current set of weights $\boldsymbol{\alpha}$, evaluated at the time point t_j matching the time of measurements. Here the motions are compared based on the position of the end-effector $f(\boldsymbol{z})$, the arm angle $\theta(\boldsymbol{z})$ and the final time T, hence $v^*(\boldsymbol{\Phi}(\boldsymbol{\alpha}), t) := (f(\boldsymbol{z}^{*,\boldsymbol{\alpha}}(t)), \theta(\boldsymbol{z}^{*,\boldsymbol{\alpha}}(t)), T^{*,\boldsymbol{\alpha}})$, see section 3.3.

Because the lower level evaluations are expensive and the results can be noisy, a derivative free optimization method

is used for the upper level. In this case the BOBYQA algorithm developed by Powell Powell (2009). The system is implemented in Matlab using the NLopt library from Johnson (2008) for the upper level optimization.

2.3 Lower level

The lower level is an optimal control problem formulated as

$$\min_{z,u,T} \int_0^T \Big[\sum_{i=1}^{n} \alpha_i U_i(\boldsymbol{z}(t), \boldsymbol{u}(t))\Big] dt \tag{3}$$

subject to: *System equation:*

$$\dot{\boldsymbol{z}} = \begin{bmatrix} \dot{\boldsymbol{q}} \\ \boldsymbol{u} \end{bmatrix}, \tag{4}$$

Forward kinematics bounday conditions:

$$f(\boldsymbol{q}(0)) = \boldsymbol{x}_0; \tag{5}$$

$$f(\boldsymbol{q}(T)) = \boldsymbol{x}_e \tag{6}$$

Joint angular speeds bounday conditions:

$$\dot{\boldsymbol{q}}(0) = 0; \tag{7}$$

$$\dot{\boldsymbol{q}}(T) = 0. \tag{8}$$

The joint angles and joint angular speeds are the states $\boldsymbol{z} = \begin{bmatrix} \boldsymbol{q} \\ \dot{\boldsymbol{q}} \end{bmatrix}$, the joint accelerations are the controls $\boldsymbol{u} = \ddot{\boldsymbol{q}}$, and f defines the forward kinematics, mapping joint angles to the end-effector pose in task space. System (3) - (8) is solved with a direct boundary value problem approach using multiple shooting as developed by Bock and Plitt (1984) together with an SQP method to solve the nonlinear programming problems. The controls are discretized by making them represent the nodes in a shape-preserving piece-wise cubic interpolation function, because this leads to a smooth function without the risk of overshoot. The time between the nodes (τ_i) is made variable as well, where a linear constraint ensures that the sum of all τ_i's is equal. Here, we use the `fmincon` function from Matlab, with the `ode45` function for numerical integration of the system equation.

The composite cost function of (1) can be combined with the system equation and be solved simultaneously by the `ode`-solver

$$\begin{bmatrix} \dot{\boldsymbol{z}} \\ \dot{U} \end{bmatrix} = \begin{bmatrix} \dot{\boldsymbol{q}} \\ \boldsymbol{u} \\ \sum_{i=1}^{n} \alpha_i U_i(\boldsymbol{z}, \boldsymbol{u}) \end{bmatrix}. \tag{9}$$

This way the accuracy of the solution of the cost is also incorporated in the error control of the `ode`-solver.

The SQP algorithm requires gradients of the objective and constraint functions to determine its search direction and to determine the first-order optimality of the current step. The forward kinematics boundary constraints (5) and (6) are based on the positional and rotational differences of the end-effector with its target. The analytical Jacobian $\boldsymbol{J}(\boldsymbol{q})$ of the arm can be used to efficiently and exactly calculate the gradient of this function towards $\boldsymbol{q}$

$$\dot{\boldsymbol{x}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \qquad \boldsymbol{J}(\boldsymbol{q}) = \Big(\frac{\partial x_i}{\partial q_j}\Big)_{i,j}. \tag{10}$$

The gradient of the objective function and the gradient of the continuity constraints are calculated numerically using a finite difference method.

3. HUMAN AND ROBOT MOTION

In this section we elaborate our choice of cost functions, describe the models and present our motion recording system.

3.1 Cost functions

The considered cost functions are inspired by previous research, not necessarily related to inverse optimal control. Criteria, minimizing hand and angular velocity, acceleration and jerk have been used to achieve smoothness of the motion in Flash and Hogan (1985). In Albrecht et al. (2011) minimization of hand jerk, joint jerk and torque change is considered.

Minimization of energy was shown to be an important criteria for humans in Berret et al. (2011). It can be calculated as

$$W = \int_0^T \dot{\boldsymbol{q}}^\mathsf{T} \boldsymbol{\tau} dt, \tag{11}$$

where $\boldsymbol{\tau}$ is the torque in the joints. However, this assumes that moving in the opposite direction of the torque generates energy, while in reality it costs energy. So instead the absolute work of torques is used, where we make the assumption that moving with and against the torque consumes the same amount of energy.

In Sylla et al. (2014) the geodesic criteria is used, it minimizes joint velocities squared times the inertia matrix. It prefers the shortest path in joint space and maximizes smoothness of the motion. Together with the energy criteria it was shown to lead to realistic human motions.

The cost functions used for the optimization are displayed in Table 1. $U_{\ddot{x}}$ was chosen because the squared function tends to increase the movement time. Other squared functions, like on joint acceleration $\ddot{\boldsymbol{q}}^\mathsf{T}\ddot{\boldsymbol{q}}$ or jerk $\dddot{\boldsymbol{q}}^\mathsf{T}\dddot{\boldsymbol{q}}$, had their weights go to zero indicating that they did not contribute to the upper level minimization.

Table 1. Cost functions used in optimization

Time	U_T	$T = \int_0^T 1 dt$
Hand acceleration	$U_{\ddot{x}}$	$\int_0^T \ddot{\boldsymbol{x}}^\mathsf{T}\ddot{\boldsymbol{x}} dt$
Energy	U_{energy}	$\int_0^T \sum \lvert\dot{q}_i \tau_i\rvert dt$
Geodesic	$U_{geodesic}$	$\int_0^T \dot{\boldsymbol{q}}^\mathsf{T}\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}} dt$

3.2 Models

The robot used in this research is the YuMi from ABB. It has two arms, both with seven degrees of freedom, and is intended for collaboration with humans. A model of the YuMi was made with the Robotic Toolbox from Corke (2017).

Figure 3 shows the kinematic structure of the human and robot models. Except for the orientation of the last joint, the main difference lies in the offsets between the joints. The human shoulder can be approximated as a ball-and-socket joint, see e.g. Soslowsky et al. (1992) and is modeled with three revolute joints with zero offset. Similarly, both the elbow and the wrists joints consist of two revolute joints with zero offsets. The similar structure

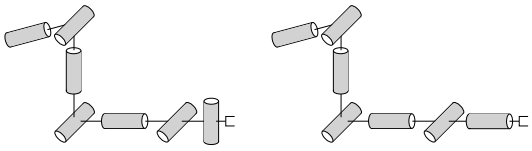

Fig. 3. The human arm (left) and robot arm (right) can be modeled similarly except for the orientation of the last joint. The differences in offsets between the joints are not shown here.

of the human and the robot arm allows to use the robot model directly within the inverse optimization, avoiding a re-scaling of weights, when using the identified objective for robot motion generation. As the energy and geodesic cost functions are based on the dynamics parameters of the model (e.g. link mass, center of mass and inertia parameters), these are included as well.

3.3 Motion recording

To capture reference data for the upper level of the inverse optimal control, we used a dual depth camera setup for recording human motion. A Microsoft Kinect v2 was chosen because of its ability to markerless track human body positions. Internally it uses a classifier to segment different parts of the body based on depth image data, Shotton et al. (2011). The SDK gives access to 25 body positions in three dimensional space. To improve position and orientation tracking of the lower arm and the hand a Leap Motion was added to the system, also a depth camera but with a closer range. The program for the motion recording was programmed in C++ on Windows. The two cameras are calibrated using a checkerboard pattern using functions provided by the OpenCV library. The body positions of the two cameras are transformed into the same coordinate frame, were the weighting is based on the confidence value δ_{conf} provided by the Leap Motion. This ensures that the hand is tracked even outside of the range of the Leap Motion without making sudden jumps.

The position and orientation of the hand with respect to the body frame are recorded, together with the arm angle and the duration of the motion. These are used to form the reference data υ_M for the upper level optimization. The arm angle parametrizes the motion in the nullspace of the arm. It is defined as the angle between a vertical reference plane and a plane though the shoulder, elbow and wrist. Together these seven values uniquely describe the arm configuration, see Kreutz-Delgado et al. (1990) and make comparison of motions between two manipulators with small kinematic differences possible.

4. RESULTS AND VALIDATION

This chapter shows the results of the bi-level optimization, expressed as the optimized weights itself and the contribution of each weighted cost to the total cost. To determine if our method for generating trajectories does indeed lead to improved collaboration we have performed an experiment where we compare it with two existing motion planning strategies. Human participants are asked to rate the shown motions on a number of scales indicating anthropomorphism and feeling of comfort and safety.

4.1 Results of the bi-level optimization

The bi-level optimization was run on four recorded trajectories, two from left to right and two from right to left. for all considered examples, the algorithm was able to derive a robot motion that closely matched the human reference, as is shown in Figure 4 for the arm angle.

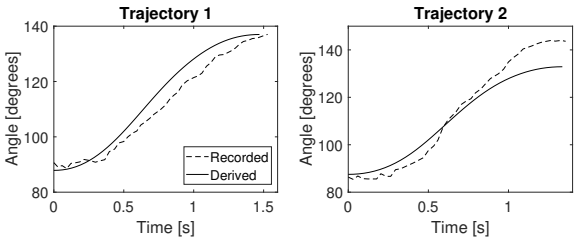

Fig. 4. Arm angle as recorded and derived with the bi-level optimization for two trajectories.

Table 2. Optimized weights for the four trajectories

	α_T	α_{energy}	$\alpha_{geodesic}$	$\alpha_{\dddot{q}}$	$\alpha_{\ddot{x}}$	RMSE
I	1	0.0110	1.0304	0	0.7001	7.08
II	1	0.0200	0.9691	0	0.5207	4.62
III	1	0	1.0957	0	0.7222	10.16
IV	1	0.0196	1.2866	0	0.7035	7.52

Table 2 shows the result of the optimization of the weights for the four trajectories. Averaging the found weights would not be feasible because they are relative to each other. So instead the set of weights that performs best on all four trajectories was chosen as final result:

$$U = T + 0.020\, U_{energy} + 0.97\, U_{geodesic} + 0.52\, U_{\ddot{x}} \quad (12)$$

Figure 5 shows the contribution of each weighted cost to the total cost. It is calculated as

$$c_i = \frac{\alpha_i U_i}{U(\boldsymbol{\alpha})} \qquad \text{with} \quad U(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i U_i. \quad (13)$$

The main difference with the result from Sylla et al. (2014) is the large influence of T and $U_{\ddot{x}}$. This can be explained by their use of a fixed end time. The geometric and temporal properties of human arm motion are decoupled, according to Biess et al. (2007). T and $U_{\ddot{x}}$ counteract each other, whereas geodesic and energy have little influence on the duration of the motion.

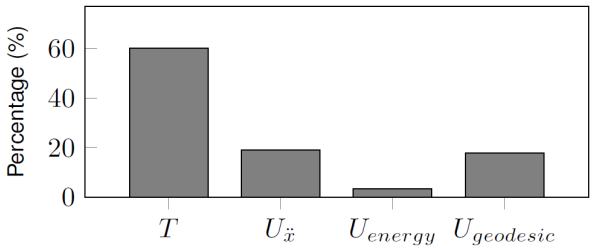

Fig. 5. Contribution of each cost function to the total cost.

4.2 Validation of results on the robot

The experiment was preformed using the YuMi robot, which at the moment has two implementations for planning point-to-point motions: MoveL, with paths linear

interpolated in Cartesian space and MoveJ, linear interpolated in joint space. These two are compared with our own implementation using the same forward optimal control scheme as was used in the lower level of section 2.3, but now with the identified cost function of (12). This motion will be referred to as 'Optimal'.

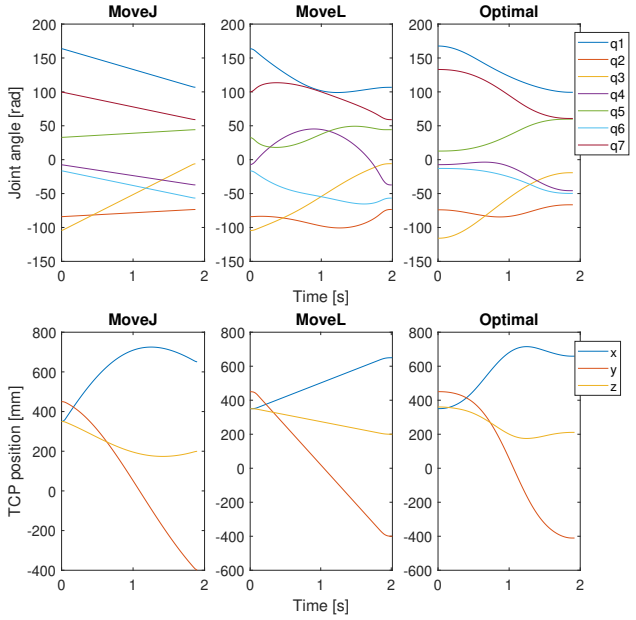

Fig. 6. Comparison of the three motion implementations used in the experiment (Joint angles / end-effector positions).

Figure 6 (top) shows the joint angle trajectories of the three motion planning implementations. As expected, MoveJ has linear interpolated paths in joint space and MoveL has curved paths. Interestingly, Optimal is somewhat in between. In the Cartesian trajectory of the end-effector MoveL has linear interpolated paths, see Figure 6 (bottom). MoveJ and Optimal both show curved paths, with Optimal following a less rounded curve.

The joint angle trajectories are generated offline and exported to motion commands in a RAPID module, which can run on the YuMi robot. The correct joint angular speeds are achieved by specifying the duration of each individual command.

Experiment A questionaire based on Bartneck et al. (2009) is used to determine the reactions of humans on the developed method. It is proposed to use a semantic differential scale to give rating to different criteria regarding anthropomorphism, animacy, likeability, perceived intelligence and perceived safety. We chose to use *Fake - Natural* and *Moving rigidly - Moving elegantly* to determine the anthropomorphism of the motions and *Anxious - Relaxed* and *Quiescent - Surprised* to see how comfortable or safe the participant feels. A group of 10 participants was tested, with an average age of 26 years. The participants were asked to rate their familiarity with robots on a scale from 1 (low) to 5 (high), on average this was 2.

Experimental procedure The participant was first given a description of the experiment and was asked to sign a consent form. The participant was sitting in front of the robot. First the three different motions were shown one after the other. Then the motions were shown again, but now with time in between to fill in the four scales. This procedure was repeated four times, with different start and end targets for the robot arm, so in total 12 motions were shown. The order of the motions was mixed for every new target, and the order of the scales was mixed between participants.

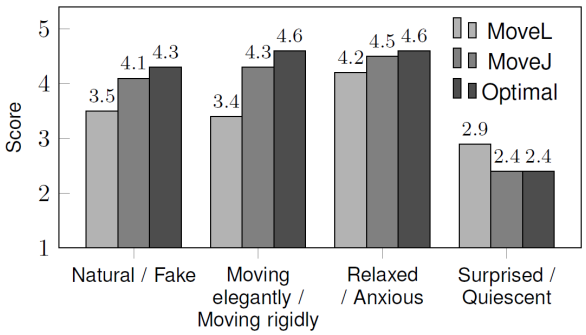

Fig. 7. Average scores given for the three motion planning strategies.

Table 3. Answers to the scales expressed in percentages

MoveL	1	2	3	4	5	
Fake	12.5	10.0	22.5	30.0	25.0	Natural
Moving rigidly	7.5	20.0	20.0	30.0	22.5	Moving elegantly
Anxious	0	10.0	12.5	30.0	47.5	Relaxed
Quiescent	22.5	15.0	22.5	30.0	10.0	Surprised
MoveJ	1	2	3	4	5	
Fake	0	7.5	10.0	45.0	37.5	Natural
Moving rigidly	0	5.0	15.0	27.5	52.5	Moving elegantly
Anxious	0	2.5	2.5	37.5	57.5	Relaxed
Quiescent	35.0	25.0	12.5	22.5	5.0	Surprised
Optimal	1	2	3	4	5	
Fake	0	2.5	2.5	55.0	40.0	Natural
Moving rigidly	0	0	5.0	27.5	67.5	Moving elegantly
Anxious	0	0	5.0	32.5	62.5	Relaxed
Quiescent	32.5	27.5	7.5	30.0	2.5	Surprised

5. DISCUSSION

From the average scores displayed in Fig. 7 it is clear that the optimal motions are perceived as more natural and more elegant than the other two implementations. The participants feel less relaxed and more surprised for the MoveL motions, but between the MoveJ and Optimal motions they feel similar. Table 3 shows the percentage of every choice on the scales. The Optimal motion scores higher for *natural*, *moving elegantly* and *relaxed*.

Using an alpha level of 0.05, a dependent-samples t test was conducted. The first two scales are measuring anthropomorphism. Combining them the test indicates that Optimal is perceived as significantly more anthropomorphic

than MoveL and MoveJ. When it comes to the last two scales the results can be combined to give a measure for perceived safety. The t test shows that there is a significant difference between the MoveL and Optimal motions but that the participants do not experience a significant difference in perceived safety between MoveJ and Optimal. However, both are a significant improvement to MoveL.

It is important to state that perceived safety is only a subjective feeling, while the actual safety of the robot needs to be ensured with other measures.

6. CONCLUSIONS

We applied inverse optimal control to point-to-point motions of a seven degrees of freedom robot arm. A bi-level optimization was introduced to identify a physically interpretable cost function suitable for the generation of robot motions that match the human recorded motions. The results of this study confirm our hypotheses that the identified cost function can be used to generate motions for new targets, that show positive effects on how the optimal motions are perceived. Our approach helps towards increasing human acceptance of robot motions and thereby towards the improvement of human-robot interaction.

In the future the approach of this work could be used for an online implementation where the forward optimal control plans its paths in a way that is comfortable for the human. The weights can also be varied based on the requirements of the current situation, for example a higher focus on energy and time minimization in the absence of a human coworker. Another interesting idea is to make the weights dependent on the specific human interacting with the robot, as preferences between humans can greatly differ. More research is needed on the tasks dependency of the cost functions, as here only point-to-point motions are considered. More complex tasks and situations might require different cost functions.

REFERENCES

Albrecht, S., Ramirez-Amaro, K., Ruiz-Ugalde, F., Weikersdorfer, D., Leibold, M., Ulbrich, M., and Beetz, M. (2011). Imitating human reaching motions using physically inspired optimization principles. In *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, 602–607. IEEE.

Alexander, R.M. (1996). *Optima for animals*. Princeton University Press.

Bartneck, C., Kulić, D., Croft, E., and Zoghbi, S. (2009). Measurement instruments for the anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety of robots. *International journal of social robotics*, 1(1), 71–81.

Berret, B., Chiovetto, E., Nori, F., and Pozzo, T. (2011). Evidence for composite cost functions in arm movement planning: an inverse optimal control approach. *PLoS computational biology*, 7(10), e1002183.

Biess, A., Liebermann, D.G., and Flash, T. (2007). A computational model for redundant human three-dimensional pointing movements: integration of independent spatial and temporal motor plans simplifies movement dynamics. *Journal of Neuroscience*, 27(48), 13045–13064.

Bock, H.G. and Plitt, K.J. (1984). A multiple shooting algorithm for direct solution of optimal control problems. *IFAC Proceedings Volumes*, 17(2), 1603–1608.

Clever, D., Hu, Y., and Mombaur, K. (2018). Humanoid gait generation in complex environments based on template models and optimality principles learned from human beings. *The International Journal of Robotics Research*.

Corke, P. (2017). *Robotics, Vision and Control: Fundamental Algorithms In MATLAB® Second, Completely Revised*, volume 118. Springer.

Englert, P., Vien, N.A., and Toussaint, M. (2017). Inverse KKT: Learning cost functions of manipulation tasks from demonstrations. *The International Journal of Robotics Research*, 36(13-14), 1474–1488.

Flash, T. and Hogan, N. (1985). The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of neuroscience*, 5(7), 1688–1703.

Johnson, S.G. (2008). The NLopt nonlinear-optimization package. URL `http://ab-initio.mit.edu/nlopt`.

Kreutz-Delgado, K., Long, M., and Seraji, H. (1990). Kinematic analysis of 7 DOF anthropomorphic arms. In *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, 824–830. IEEE.

Mainprice, J., Hayne, R., and Berenson, D. (2015). Predicting human reaching motion in collaborative tasks using inverse optimal control and iterative re-planning. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, 885–892. IEEE.

Mombaur, K., Truong, A., and Laumond, J.P. (2010). From human to humanoid locomotionan inverse optimal control approach. *Autonomous robots*, 28(3), 369–383.

Powell, M.J. (2009). The BOBYQA algorithm for bound constrained optimization without derivatives. *Cambridge NA Report NA2009/06, University of Cambridge, Cambridge*, 26–46.

Sciutti, A., Mara, M., Tagliasco, V., and Sandini, G. (2018). Humanizing human-robot interaction: On the importance of mutual understanding. *IEEE Technology and Society Magazine*, 37(1), 22–29.

Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A. (2011). Real-time human pose recognition in parts from single depth images. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 1297–1304. Ieee.

Soslowsky, L.J., Flatow, E.L., Bigliani, L.U., and Mow, V.C. (1992). Articular geometry of the glenohumeral joint. *Clinical orthopaedics and related research*, (285), 181–190.

Sylla, N., Bonnet, V., Venture, G., Armande, N., and Fraisse, P. (2014). Human arm optimal motion analysis in industrial screwing task. In *Biomedical Robotics and Biomechatronics (2014 5th IEEE RAS & EMBS International Conference on*, 964–969. IEEE.