# Real-Time Predictive Control of an UR5 Robotic Arm Through Human Upper Limb Motion Tracking

Bukeikhan Omarali
School of Engineering, NURIS
Nazarbayev University
53 Kabanbay Batyr Avenue
Astana, Kazakhstan Z05H0P9
bukeikhan.omarali@nu.edu.kz

Tasbolat Taunyazov
School of Science and Technology
Nazarbayev University
53 Kabanbay Batyr Avenue
Astana, Kazakhstan Z05H0P9
tasbolat.taunyazov@nu.edu.kz

Askhat Bukeyev
National Lab Astana, NURIS
Nazarbayev University
53 Kabanbay Batyr Avenue
Astana, Kazakhstan Z05H0P9
askhat.bukeyev@nu.edu.kz

Almas Shintemirov
School of Science and Technology, Nazarbayev University
53 Kabanbay Batyr Avenue, Astana, Kazakhstan Z05H0P9
ashintemirov@nu.edu.kz

## ABSTRACT

This paper reports the authors' initial results on developing a real-time teleoperation system for an Universal Robots robotic arm through human motion capture with a visualization utility built on the Blender Game Engine open-source platform. A linear explicit model predictive robot controller (EMPC) is implemented for online generation of optimal robot trajectories matching operator's wrist position and orientation, whilst adhering to the robot's constraints. The EMPC proved to be superior to open-loop and naive PID controllers in terms of accuracy and safety.

## Keywords

Universal Robots UR5, real-time robot control, model predictive control, motion tracking

## 1. INTRODUCTION

The advent of virtual reality brings new possibilities for robot teleoperation as it allows to put the operator into the robot's environment. This calls for a shift of robotic arm control paradigms to developing more natural robot teleoperation systems capable of replicating limb motion of the human operator in real-time.

Mapping the human limb motion to the robotic arm is a rather complex problem due to existing technical limitations of commercial robotic arms, i.e. large inertia, limited joint acceleration and velocity. This task becomes more complex due to unpredictability of the human motion. In addition, slow robot feedback communication frequency and computational load of the robot control may prevent the system from running at the rates fast enough for smooth and accurate motions. Furthermore it needs to be taken into account that most commercial robots' APIs are not streamlined for real-time operations as they are usually designed to operate in scripted offline mode. In order to overcome limitations listed above one needs to develop a robot control system able to interpret the human operator motions, plan and execute the robot's trajectory in real-time whilst taking into account the robot's constrains.

This paper reports the authors' initial results on developing a real-time control system for an Universal Robot UR5 robotic arm through human motion capture with a visualization utility built on an open-source platform.



*HRI '17 Companion March 06-09, 2017, Vienna, Austria*



ACM ISBN 978-1-4503-4885-0/17/03.

DOI: http://dx.doi.org/10.1145/3029798.3038918

## 2. HUMAN LIMB MOTION TRACKER

The motion tracker system used in this project is based on work presented in [1] extended to cover both human arms with wrists. The motion tracker design scheme is given in Fig. 1a. Presently the robot only follows motions of the user's left arm. The right arm is intended for additional functionality outside of the paper's scope.

The motion tracker runs and is visualized in the Blender Game Engine on a modified Guy rig [2]. The robot CAD model and the external robot controller were also added to the Blender Game Engine for visualizing the robot operations as shown in Fig. 1b. The system is set up so that the robot's end effector position and orientation correspond to those of the operator's wrist (from now on referred to as "target"). The IK solver used is the Blender integrated iTaSC (instantaneous Task Specification using Constraints) SDLS (selective damped least squares) solver [3].

## 3. ROBOT COMMUNICATION

The robot used in this research is the Universal Robots UR5 robotic arm with a CB2 controller. There are number of drivers available for UR robots such as the ROS [4] or MATLAB [5] based drivers. However, authors intend to design a system independent from both platform and software, thus, a custom communication driver was developed.

The robot's API allows to either set the robot's joint angles (using a *servoj* command) or angular velocities (using a *speedj* command) that can be respectively used for open-

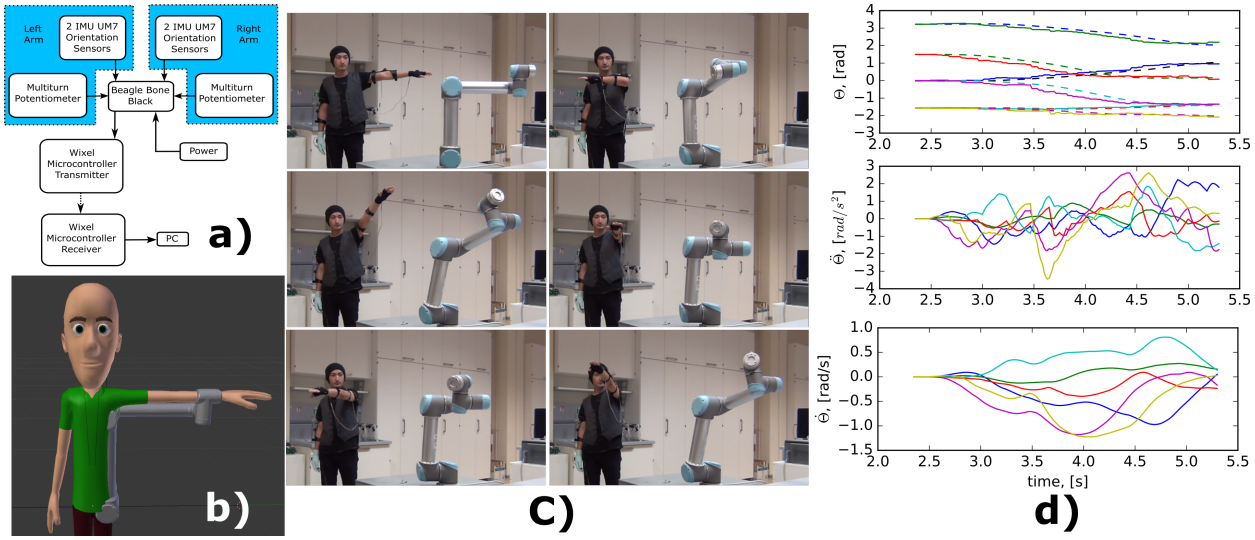


**Figure 1: The UR5 robot teleoperated in real-time through the human upper limb motion tracker system**

loop or closed-loop controls of the robot. The robot's IK solutions are streamed to the robot's controller from an external PC via a TCP/IP connection. The same connection is also used to obtain feedback on robot's states.

## 4. ROBOT PREDICTIVE CONTROL

Initial tests using open-loop robot teleoperation resulted to unacceptable jittery motions and significant inaccuracy. Further trials were conducted with implemented PID closed-loop position controllers. In these tests the robot often could not follow the target with a desired velocity due to internal torque limitations. In these situations the robot was moved with maximum allowable acceleration. However it would often catch up with the target at a velocity higher than the target's velocity. As a result, the robot would overshot and oscillate around the target position until it came to a stop.

It was then concluded that the robot needs to plan it's trajectory such that it would converge with the target's position and velocity whilst adhering to the robot's constraints. Assuming that the robot internal control loop implements inverse dynamics control, the external controller can be realized as a linear EMPC with a simple double integrator model. The optimization problem is solved for each joint separately according to (1).

$$\begin{aligned} \underset{u}{\text{minimize}} \quad & (r-y)^T Q_1 (r-y) + u^T Q_2 u + \Delta u^T Q_3 \Delta u \\ \text{subject to} \quad & \dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u; \quad y = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x; \\ & |\begin{bmatrix} 0 & 1 \end{bmatrix} * x| \leq v_{max}; \qquad |u| \leq u_{max}. \end{aligned} \tag{1}$$

The state vector consists of joint angle and angular velocity $x = [\theta \ \dot{\theta}]$ and $u$ is the control angular acceleration of the joint. The target vector $r$ is given by IK solver passed through a linear Kalman filter. The Kalman filter smooths the IK solution which in turn improves the performance of the EMPC. The maximum angular velocity and acceleration constraints $v_{max}$ and $u_{max}$ are set to 3.14 rad/s and 5 rad/s$^2$, respectively. Meanwhile, cost matrices $Q_1, Q_2, Q_3$ were defined by trials. As a rule of thumb, the best tracking occurs when the cost function is dominated by the position error penalty. The prediction horizon is set to 0.5 sec. The solver C code is generated using the Multi-Parametric Toolbox [6] and integrated into the Blender Python code using Cython.

EMPC tests resulted in smooth and accurate motions of the UR5 robotic arm, as illustrated in Fig. 1c. Figure 1d shows the EMPC performance sample: solid graphs in $\theta$ subfigure are the pre-Kalman IK solutions for each joint, dashed - the robot feedback. In $\dot{\theta}$ and $\ddot{\theta}$ subfigures solid graphs represent MPC generated angular velocities and accelerations used as inputs in *speedj*. It can be seen that the robot follows the target quite closely and does not exceed the robot's limits. The operation lag averages at 0.05 sec. when the operator moves under the robot's velocity and acceleration limits. If the operator moves too fast for the robot to keep up in real-time, the robot is able to catch up with the operator with little or no overshooting.

It is planned to integrate the system with virtual reality headset such that the robot could be operated distantly from a virtual environment.

## 5. REFERENCES


[1] T. Taunyazov, B. Omarali, and A. Shintemirov, "A novel low-cost 4-DOF wireless human arm motion tracker," in *2016 6th IEEE International Conference on Biomedical Robotics and BioMechatronics (BioRob)*, 2016, pp. 157–162.

[2] Guy character rig by VMComix licensed under CC BY 3.0. [Online]. Available: http://www.blendswap.com/blends/view/24092

[3] J. De Schutter, T. De Laet, J. Rutgeerts, W. Decré, R. Smits, E. Aertbeliën, K. Claes, and H. Bruyninckx, "Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty," *The International Journal of Robotics Research*, vol. 26, no. 5, pp. 433–455, 2007.

[4] T. T. Andersen, "Optimizing the Universal Robots ROS driver." Technical University of Denmark, Department of Electrical Engineering, Tech. Rep., 2015.

[5] M. De Gier, "Control of a robotic arm: Application to on-surface 3D-printing," Ph.D. dissertation, TU Delft, Delft University of Technology, 2015.

[6] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, "Multi-Parametric Toolbox 3.0," in *Proc. of the European Control Conference*, 2013, pp. 502–510.