A Predictor for Operator Input for Time-Delayed Teleoperation

Christian Smith*,[a], Patric Jensfelt[a]

[a] *Centre for Autonomous Systems, Royal Institute of Technology, Stockholm, Sweden.*

Abstract

In this paper we describe a method for bridging internet time delays in a free motion type teleoperation scenario in an unmodeled remote environment with video feedback. The method proposed uses minimum jerk motion models to predict the input from the user a time into the future that is equivalent to the round-trip communication delay. The predictions are then used to control a remote robot. Thus, the operator can in effect observe the resulting motion of the remote robot with virtually no time-delay, even in the presence of a delay on the physical communications channel. We present results from a visually guided teleoperated line tracing experiment with 100 ms round-trip delays, where we show that the proposed method makes a significant performance improvement for teleoperation with delays corresponding to intercontinental distances.

Key words: teleoperation, time delay, human motion models

1. Introduction

Teleoperation in robotics has been studied for more than half a century. One typical teleoperation scenario that has been around since the beginning consists of an operator controlling a robot manipulator with a joystick at the *master* site, while viewing video feedback from the remote *slave* site. This could for example be the free-motion part of tele-surgery, hazardous material handling, underwater robotics, or rescue robotics. In many of these cases, the perception and reasoning abilities as well as the task related skills of the human operator are of paramount importance for successful task completion. The task may be too complex for autonomous treatment.

When the distance between master and slave becomes long in comparison to the transfer speeds of control signals, time delays will be present in the control loop. Recently, as packet-switched networks like the internet are being used for teleoperation, time delays due to routing delays may also have a significant impact. These delays become a significant factor in designing telerobotic systems, as they introduce instabilities. This can either occur directly in the automatic controller as can be the case for bilateral teleoperation with force feedback, or indirectly by the human operator overcompensating for perceived errors due to delayed responses, giving rise to operator-induced oscillations [1, 2]. After a while, human operators typically adapt to this by decreasing motion speed, or by only moving short distances and then waiting for the result of a motion to be seen until moving again, drastically worsening the performance as compared to a

*Corresponding Author

Email addresses: `ccs@kth.se` (Christian Smith), `patric@kth.se` (Patric Jensfelt) non-delayed setup [3, 4]. Time delays also decrease the operator's sense of directly interacting with the remote environment — what is commonly referred to as transparency [5].

Different methods for dealing with this has been one of the main focuses of the field of teleoperation for several decades. Traditionally, delays on visual feedback systems have been handled by predictive methods or through increased autonomy - i.e. higher level control, where the necessary degree of autonomy increases with the time delay [6, 7, 8]. Most prediction-based delay compensation methods relay on predicting the feedback from the remote environment, which may be difficult to model and predict, especially if the main feedback is a video signal.

It is well known that humans are visually dominated [9], and there are clear models that suggest that much of human actions are feed-forward driven rather than feedback driven [10]. Under such a control regime one would expect that motion for interaction is largely prospective, in particular for dynamic situations. The fundamental question addressed by the present paper is: given this knowledge, would it be possible to estimate and predict the motion input by the user with sufficient accuracy to allow it be to used to generate a predictive motion control signal for the slave device? If one could do this, it would be possible to compensate for time delays and decrease the performance losses they cause without having to predict the observations of the remote environment.

This would allow us to approach teleoperation scenarios with little or no explicit knowledge of the dynamics of the remote site, as all understanding of the remote environment is handled by the human operator. In previous work, we have shown that the user input prediction approach is possible for simple point-to-point reaching motions [11, 12, 13]. In the present paper, we extend this to continuous motions.

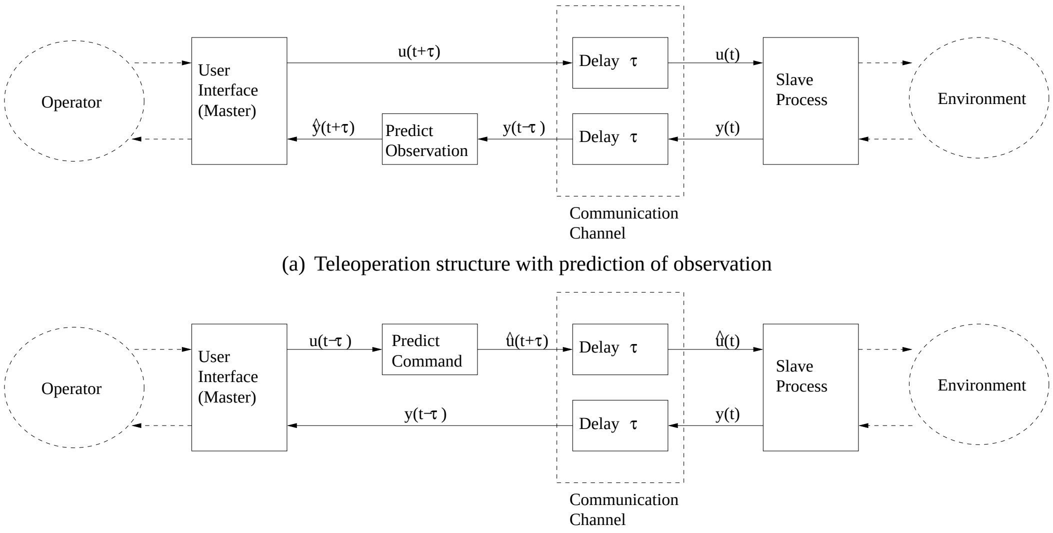
We study the prediction of commands issued by the user under significant time pressure. The questions addressed are:

- What are good estimation models for predicting the future trajectory specified by the user?
- How can these models be integrated into the control system?
- What kind of performance can be achieved with a real system?

The paper is organized as follows: In Section 2 we describe the principles of our proposed method of user input prediction. Section 3 describes an implementation using superimposed minimum jerk polynomials to approximate human motion. Section 4 describes an experiment where we let 12 subjects perform a line-tracing task with a telerobotic system, where we compare the results with and without a 100 ms round-trip time delay, and with and without the input prediction system. Finally, Sections 5 and 6 present the conclusions and future work.

2. Proposed Method

Time-delayed teleoperation is a well-studied problem. For a more complete survey, see for example [14, 7, 8]. We here mention some main approaches that have been used for setups with visual feedback. With *task level* or *supervisory* control, the operator performs higher level control, such as e.g. task selection, while low-level control, e.g. obstacle avoidance, is performed autonomously at the remote site [15, 16, 17, 18, 19, 20, 21, 22]. Other systems use different kinds of *predictive* or *simulated* displays to present the operator with a real-time simulation of the remote site alongside the delayed feedback [23, 24, 25, 26, 27]. A variant is the *hidden robot* concept, where the remote robot is not displayed to the operator, but only the remote environment



(a) Teleoperation structure with prediction of observation

(b) Teleoperation structure with prediction of command signal

Figure 1: Teleoperation control structure with predictions to bridge time delay. In the present paper, u is position commands and y is visual feedback.

itself, which can be "felt" via haptic feedback to the user [28]. A special case of simulated display is *teleprogramming*, where the task is first performed in simulation until a satisfactory performance has been recorded, and replayed at the remote site [29]. Recent *model-mediated* methods simulate the remote environment locally for high-bandwidth interaction, and use sensor data from the remote site to estimate and update model parameters [30].

However, neither of these methods will be adequate if the measurements from the remote site are difficult to model and predict (for example an uncalibrated video signal), the task is too novel or difficult for autonomous treatment, and we still want good transparency for very fast motions, at the limit of the operator's cognitive performance.

Several published methods use different ways to predict the remote site, or put in control terms, one substitutes the unavailable future measurement y with the predictor $\hat{y}$. A schematic of this is shown in Figure 1(a).

We propose a different control structure. Instead of handling the round-trip delay by predicting the remote state with a simulation, the delay handling is moved to a model-based input predictor that predicts the operator's motions. The principal structure for this approach is shown in Figure 1(b). Predicting the operator's input over the one-way delay, and predicting the feedback from the remote site over the one-way return trip delay has been proposed previously, both with model-free prediction [31] and model-based prediction [32]. Handling the entire round-trip delay by predicting the operator's input is a novel approach to the best knowledge of the authors.

With this approach, video, audio, or other data from the remote site that may be difficult to predict, can be displayed as is, and there is no need for models of the remote site. Video feedback based control can therefore be performed with a camera with an unknown position in relation to the remote robot and task, as long as the camera shows an adequate view of the task space, enabling a human operator to interpret and react to the scene.

3. Implementation

In this section, we present a method to include a simulation of the operator's input in the control loop. This requires a model of the user's input that relates previous and present inputs to future ones, and a method to incorporate the predictions from the model into a control system.

3.1. Motion Models

A well-known model for explaining the kinematics of visually guided human reaching motions is the *minimum jerk* (MJ) model. It was first proposed for single-joint motions in [33], and later extended to include multi-joint planar motion in [34]. It was observed that the trajectory of voluntary arm motions, when described in extra-corporeal Cartesian space, follow certain constraints. The trajectories can be described by using a model that minimizes C, defined as the integral over time of the square sum of the third derivative of position, *jerk* (γ), for the duration of the motion (from t_0 to t_1), see Equation 1 [33].

$$C = \int_{t_0}^{t_1} \gamma(t)^2 dt \tag{1}$$

I.e, given a starting point, an end point and a time to move between the two, the trajectory that minimizes the jerk on this interval is the MJ trajectory. Figure 2(a) shows an MJ trajectory that has been fit to a measured human input signal from an experiment in [12].

More recently, other, more detailed models like minimum joint torque change, minimum force, or minimum energy have been proposed and shown to describe human motions more accurately [38, 35, 39]. However, the more detailed models require detailed knowledge of posture and intrinsic mechanical parameters of the human subject, parameters that are not observable when using a hand-held joystick-type device for motion input. On the other hand, the MJ model can be completely described in Cartesian space coordinates with no explicit knowledge of the subject's intrinsic mechanical properties, and is thus possible to implement with only the data from a joystick-type input device. Furthermore, when a subject is using a small input device with low inertia, it is reasonable to assume that no external forces other than gravity act on the user, and that posture does not change significantly during the motion. In this case, it has been shown that trajectories predicted by the MJ model do not differ significantly from ones predicted by the more advanced models [38].

All MJ trajectories share the property that the 6th derivative is zero for the duration of the motion, and that they thus can be described as 5th degree polynomials $x_{MJ}(t)$, as in Equation 2.

$$x_{MJ}(t) = b_5 t^5 + b_4 t^4 + b_3 t^3 + b_2 t^2 + b_1 t + b_0, \qquad t_0 \leq t \leq t_1 \tag{2}$$

If we also add the start and end points of the motion, $x(t_0)$ and $x(t_1)$, and state the position, velocity, and acceleration at these points, we get the following constraints on Equation 2.

$$\begin{aligned} x_{MJ}(t_0) &= x_0, & x_{MJ}(t_1) &= x_1 \\ \dot{x}_{MJ}(t_0) &= \dot{x}_0, & \dot{x}_{MJ}(t_1) &= \dot{x}_1 \\ \ddot{x}_{MJ}(t_0) &= \ddot{x}_0, & \ddot{x}_{MJ}(t_1) &= \ddot{x}_1 \end{aligned}$$

The above constraints will give us 6 equations, and we get a well-defined system to find the 6 parameters $b_0 \ldots b_5$. Thus, there is only one possible MJ trajectory for a given start and end, and it can be found by solving a simple system of linear equations. For the general case, the velocity and acceleration are zero at the start and end points, $\dot{x}_0 = \dot{x}_1 = \ddot{x}_0 = \ddot{x}_1 = 0$. Using this, along with the definition of x_0, we get 5 equations, and we can rewrite the equation as a function of b_5 and x_0:

$$\begin{aligned} x_{MJ}(t) = \quad & x_0 + b_5 \Big[t^5 - \tfrac{5}{2}(t_0 + t_1)t^4 + \tfrac{5}{3}({t_1}^2 + 4t_1t_0 + {t_0}^2)t^3 - \\ & 5(t_1{t_0}^2 - {t_1}^2t_0)t^2 + 5{t_0}^2{t_1}^2t - \\ & \tfrac{1}{6}{t_0}^5 + \tfrac{5}{6}{t_0}^4t_1 - \tfrac{5}{3}{t_0}^3{t_1}^2 \Big] \end{aligned} \tag{3}$$

Where the remaining constants are related as:

$$\begin{aligned} b_4 &= -\tfrac{5}{2}(t_0 + t_1)b_5 \\ b_3 &= \tfrac{5}{3}(t_0^2 + t_1^2 + 4t_0t_1)b_5 \\ b_2 &= -5(t_0^2t_1 + t_0t_1^2)b_5 \\ b_1 &= -5b_5t_0^4 - 4b_4t_0^3 - 3b_3t_0^2 - 2b_2t_0 \\ b_0 &= x_0 - (b_5t_0^5 + b_4t_0^4 + b_3t_0^3 + b_2t_0^2 + b_1t_0) \end{aligned}$$

It has been shown [33] that if the motion starts at $t_0 = 0$, Equation 3 can be written as

$$x_{MJ}(t) = \quad x_0 + (x_1 - x_0)\left(6(\tfrac{t}{d})^5 - 15(\tfrac{t}{d})^4 + 10(\tfrac{t}{d})^3\right) \tag{4}$$

where $d = (t_1 - t_0)$ is the duration of the motion. If we relieve the assumption that $t_0 = 0$, we get

$$x_{MJ}(t) = \quad x_0 + (x_1 - x_0)\left(6(\tfrac{t-t_0}{d})^5 - 15(\tfrac{t-t_0}{d})^4 + 10(\tfrac{t-t_0}{d})^3\right) \tag{5}$$

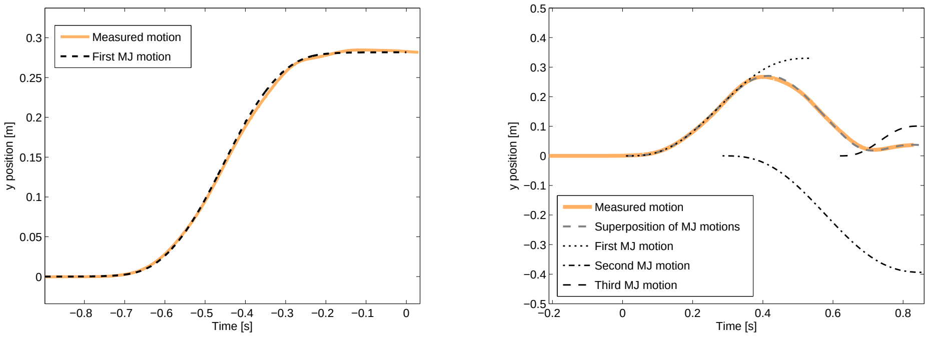
Thus, the entire trajectory is defined completely by the starting point x_0, the end point x_1, and the start and end times t_0 and t_1. For 3-dimensional motion, each dimension of the motion is described by such a polynomial, where the coefficients for the different dimensions are independent from one another, but t_0 and t_1 are the same [38].

The trajectories described by this simple MJ model are limited to one single MJ motion, which is a straight line of finite length. If a more complex motion is desired, or if the target of the motion is changed in mid-motion, the trajectory can be described by superpositioning several MJ trajectories. If the added MJ trajectory has an initial position, velocity, and acceleration of zero, this will still result in a continuous motion where the 6th derivative is zero, so the jerk is still minimized. For motions of this type, a common trait is that the tangential velocity tends to be lower as the radius of curvature decreases. Compound MJ motions have been described thoroughly in [40, 41, 42, 43]. In the motor control system of humans, a new submovement can be generated as often as once every 100 ms [44]. Figure 2(b) shows three such MJ trajectories superimposed to fit a human input signal from an experiment in [12].

MJ trajectories have been used extensively in the past to generate human-like motions for robots [45, 46, 43, 47, 48, 49, 50]. Recently, work has also been performed that fits MJ trajectories to user inputs in teleoperation, using environment knowledge to generate MJ-shaped virtual fixtures [51]. Also, MJ type trajectories have been used to estimate current operator input for generating force commands for admittance control [52].

3.2. Human Input Estimation

We fit the input data to an MJ model using least squares. The resulting polynomial is then extrapolated to obtain a prediction. This prediction is then tracked with a Kalman filter. The details are described below. The methods for detecting and fitting to MJ submotions have been described earlier in [11, 12, 13].



(a) One of the components (y) of the measured hand trajectory with MJ trajectory fitted. In this case the hand trajectory contains only one major MJ component

(b) One of the components (y) of the measured hand trajectory with MJ trajectory fitted. In this case the hand trajectory contains two major MJ components and one minor. The MJ components are shown superimposed to an approximation of original data.

Figure 2: MJ trajectories fit to observed hand motion data.

3.2.1. Detecting the MJ Submotion

The first step is detecting an MJ submotion, finding its start and end times and determining the magnitude. It is impossible to detect the start of a motion before it occurs, but the point in time at which the motion ends can to be found before it can be observed. Using Equation 5, we see that if the start of a motion is known, but not the end, there are two unknowns — the time and position at which the motion ends. Thus, in theory, if two points on the trajectory are known, the system of equations can be solved to find these two unknowns and thereby specify the entire motion exactly. Since this involves solving a 5th degree polynomial, the solution will be very sensitive to measurement error, especially in the time domain. However, if the start and end times are known, extrapolations that fit well with observed data are possible.

A robust way to detect the start time t_0 and the end time t_1 is proposed in the following. Since the motion itself is expected to follow a 5th degree polynomial curve, the velocity profile is expected to follow a 4th degree polynomial curve that starts and ends at zero value, with zero first derivative. Such a curve is symmetric around the apex. Empiric analysis of recorded motion data shows that by using an extended Kalman filter (EKF), the point of maximum velocity $t_{v_{max}}$ is easy to detect. A threshold value requires the peak velocity to be above a certain magnitude to be registered. Using a least square approach, the 4th degree velocity profile can be fit around this peak, fitting only to data before the peak, see Fig 3. The zeros of the polynomial found with this approach are used as candidates for the start and end of the motion, and used when fitting the motion data to an MJ trajectory.

A weakness in this approach is that half of the motion has to be observed before a prediction can be made. In order to facilitate earlier predictions, the EKF observer can be used to predict when $t_{v_{max}}$ will be reached. This involves estimating higher derivatives, and is prone to high uncertainties in the presence of observation noise. In practice this means that a stable estimate of $t_{v_{max}}$ can be achieved after approximately one third of the motion. The implementation used in the experiments described later in this paper therefore uses a $t_{v_{max}}$ predicted from the EKF until

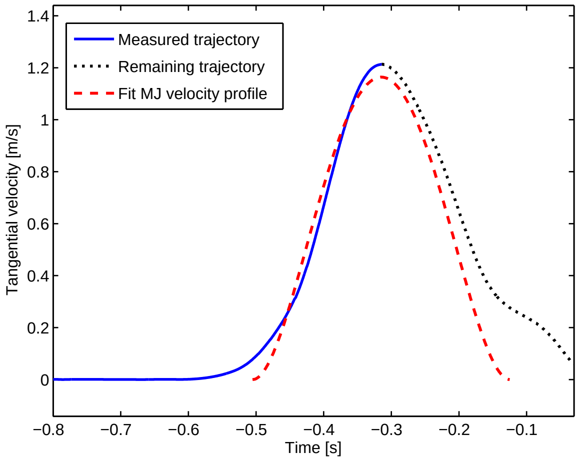


Figure 3: 4th degree MJ velocity profile fitted to Kalman filtered velocity data. The solid line is the data used for fitting, the dotted lined is the (unused) remainder of the measured velocity, and the dashed line represents the fit MJ velocity profile.

$t_{v_{max}}$ is reached, after which the algorithm switches to using the observed value.

When the peak velocity has been passed, at times $t > t_{v_{max}}$, the polynomial is subtracted from all incoming velocity measurements up to time t_1, and the algorithm tries to find the next velocity peak.

3.2.2. Fitting an MJ Submotion to Data

When t_0 and t_1 are known, along with the start position x_0 of the motion, Equation 3 only contains one unknown, b_5. By fitting to the latest measured data points with a least squares fitter, robust trajectory prediction for time $t + \tau$ is possible by calculating $x_{MJ}(t + \tau)$. For values where $(t + \tau) > t_1$, we let $x_{MJ}(t + \tau) \equiv x_{MJ}(t_1)$.

3.2.3. Tracking the Predictions

The predictions achieved this way are tracked by an extended Kalman filter. In the following treatment, we apply the same definitions and terminology as Welch [53]. We define the filter state $\mathbf{X}$ as

$$\mathbf{X} = \{x, y, v, \theta, \dot{v}\}, \tag{6}$$

where x and y are the Cartesian position coordinates, v is the tangential velocity, θ is the direction of motion, and $\dot{v}$ is the tangential acceleration. The motivation for this polar representation of velocity is that MJ trajectories are defined by 4th degree polynomials in tangential velocity, making the implementation more straightforward.

In this case, we assume that the direction of motion remains unchanged for each MJ submotion, so that the observation Z_{MJ} will be the value of the MJ polynomial added to the last measured position in the direction of motion:

$$Z_{MJ}(t) = \begin{bmatrix} x(t-\tau) + (x_{MJ}(t) - x_{MJ}(t-\tau)) \cdot \cos(\theta) \\ y(t-\tau) + (x_{MJ}(t) - x_{MJ}(t-\tau)) \cdot \sin(\theta) \\ \dot{x}_{MJ}(t) \\ \ddot{x}_{MJ}(t) \end{bmatrix} \tag{7}$$

where $\dot{x}_{MJ}$ and $\ddot{x}_{MJ}$ are the first and second derivatives of x_{MJ} with respect to time. The observation matrix H_{MJ} is then

$$H_{MJ} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{8}$$

For cases when there are no MJ predictions available, i.e. at the beginning of an MJ submotion, or during motions too slow to trigger the threshold value for the MJ detector, an ordinary EKF extrapolation from the (delayed) measured position data is used as observation. Given the delayed measurements $\mathbf{Y}_d$:

$$\mathbf{Y}_d(t) = \begin{bmatrix} y_{d,1} \\ y_{d,2} \\ y_{d,3} \\ y_{d,4} \\ y_{d,5} \\ y_{d,6} \end{bmatrix} = \begin{bmatrix} x(t-\tau) + w_1 \\ y(t-\tau) + w_2 \\ \dot{x}(t-\tau) + w_3 \\ \dot{y}(t-\tau) + w_4 \\ \ddot{x}(t-\tau) + w_5 \\ \ddot{y}(t-\tau) + w_6 \end{bmatrix} \tag{9}$$

where w_i is measurement noise, we model the observation Z_{EKF} as:

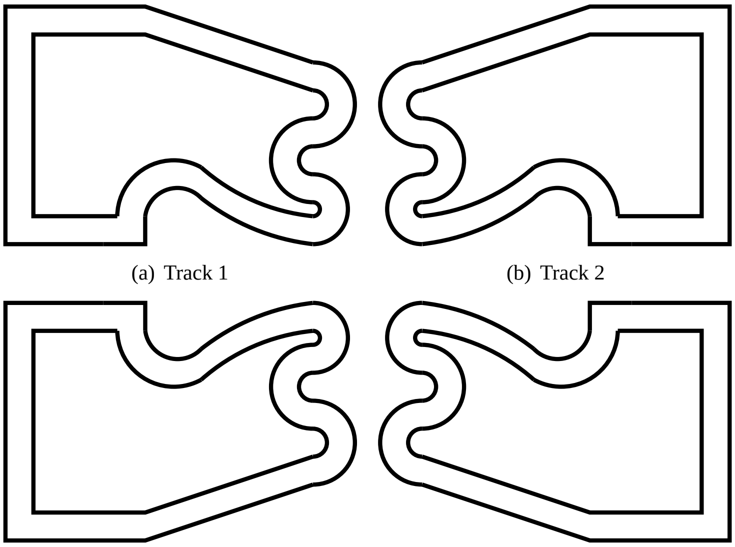
$$Z_{EKF}(t) = \begin{bmatrix} y_{d,1} + y_{d,3}\tau + y_{d,5}\tau^2/2 \\ y_{d,2} + y_{d,4}\tau + y_{d,6}\tau^2/2 \\ \sqrt{(y_{d,3} + y_{d,5}\tau)^2 + (y_{d,4} + y_{d,6}\tau)^2} \\ tan^{-1}((y_{d,4} + y_{d,6}\tau)/(y_{d,3} + y_{d,5}\tau)) \end{bmatrix} \tag{10}$$

The observation matrix H_{EKF} is then:

$$H_{EKF} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{11}$$

4. Evaluation Experiments

A prototype teleoperated drawing task was designed, and a setup was constructed so that an operator could control the robot with video feedback through arbitrary time-delays.



(a) Track 1

(b) Track 2

(c) Track 3

(d) Track 4

Figure 4: The tracks used in the tracing task.

4.1. Task Description

The users try to trace a path as fast as possible on a paper with a pen, using a robot controlled though a joystick-type interface. The task is significantly more difficult if a small time delay is added. A small pilot study showed that the task completion time increased by 40 to 50% with as little as 100 ms round-trip delay.

The path the subjects were given to trace contains a mixture of straights, sharp corners, and curves of varying radius. Four different tracks were constructed by mirroring the first track about the horizontal or vertical axis, see Figure 4. The tracks are defined by an inner and an outer line, and the task is to draw a trace between these boundaries.

4.2. Robot Platform

The robot used in this experiment is a fast lightweight manipulator with six degrees of freedom, see Figure 5. The robot can reach any point in the 30 cm × 42 cm (A3) workspace within 300 ms with the settings used in this experiment. The robot uses a combination of PID and computed torque control (CTC) to track setpoints with combined definitions in both position and velocity space, and is thoroughly described in earlier work [54].

The end effector used in this experiment is a permanent marker pen mounted in a short PVC pipe. The drawing area is a sheet of A3 paper fixed to a 5 mm aluminum sheet suspended by taught springs. In this setup, only two degrees of freedom are controllable by the operator: the Cartesian *x* and *y* dimensions of the tip of the pen in the plane of the paper.

The network interface of the robot is set so that all incoming commands are delayed for a preset time before being relayed to the robot controller. This preset time delay can easily be changed between experiments.

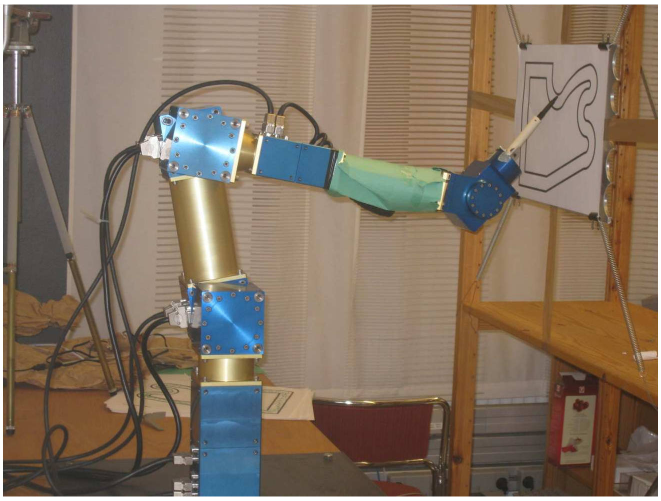


Figure 5: The manipulator with pen and drawing surface.

4.3. *User Interface*

The user interface (shown in Figure 6) used for the experiment consists of a video feed and a position input device.

The video is obtained with an inexpensive USB camera with 640 × 480 resolution and a framerate of 15 fps, which was mounted to point towards the drawing area. The unprocessed video image of the drawing surface is then shown on a 19-inch monitor.

An Omega haptic unit from Force Dimension serves as input channel for user hand motion. This device provides a large workspace, as well as high stiffness and force output. It is sampled at 2.2 kHz. Force feedback is used to define the limits of the workspace, so that drawing is possible on the entire 2 dimensional surface of the paper, but not outside. Gravity, friction, and inertia are canceled by the controller, so that within the boundaries no net forces act on the operator.

4.4. *Proof of Concept*

In an initial proof of concept, the task was performed without time delays, and all inputs were recorded. The proposed model was then applied offline to the measured inputs. For comparison, we also applied a predictor using the same EKF tracking but no MJ predictions, just the extrapolations from measured data. This is a near-optimal predictor without human motion models.

We observe that the MJ prediction errors are smaller than the errors caused by the delay, or the errors we get by applying the EKF predictor based purely on input measurements, See Table 1. The errors were measured at points were a valid MJ prediction was available. As described in 3.2, a valid prediction is available after $\frac{1}{3}$ of the submotion has been observed. With a time delay τ, this means that it is available at time $t_0 + \frac{1}{3}d + \tau$, and will remain valid until it is possible for a new submotion to dominate the trajectory, which according to empiric study of operator input was determined to be approximately 200 ms after the peak of the current MJ submotion.

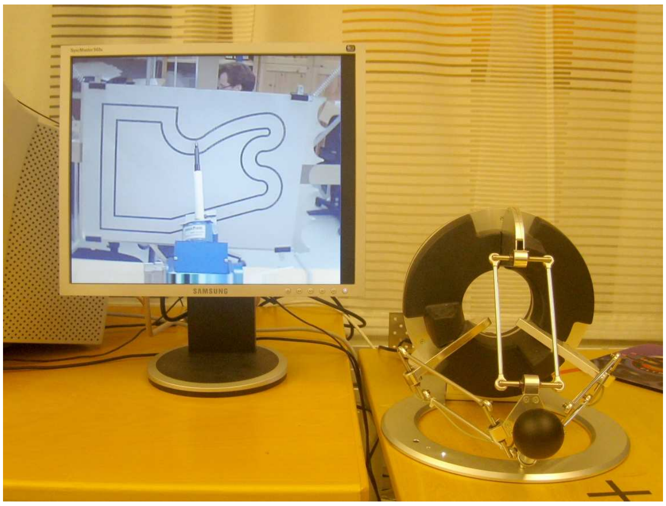


Figure 6: User interface hardware, with video display of remote site and input device.

Table 1: Comparison of mean errors of the MJ model and pure EKF predictions.

	Delayed signal	Non-MJ EKF	MJ Predictor
Mean error error E_i	12.1 mm	3.7 mm	2.9 mm

In the proof of concept trial, track 1 was circled 7 times, both clockwise and counter clockwise, for a duration of 100 seconds. During this time a total of 317 distinct MJ motions were detected by the system.

The error measurement E_i for each predictor function P_i was defined as follows:

$$E_i \equiv \frac{1}{n}\sum_{k=1}^{n} ||\mathbf{X}(t_k) - P_i(t_k)||^2 \tag{12}$$

Where $\mathbf{X}(t)$ is the true input signal at time t, with position given in millimeters, and t_k, $k = \{1 \ldots n\}$ are all time points where a valid MJ prediction exists. In order to quantify the error that was caused by the delayed signal without a predictor, a predictor function defined as the position at the time $t - \tau$ was used:

$$P_{delay}(t) \equiv \mathbf{X}(t - \tau) \tag{13}$$

where τ is the round-trip time delay, in this case 100 ms.

In terms of average error magnitude, the delayed signal has an average error of 12.1 mm, the signal from the non-MJ EKF 3.7 mm, and the signal from the MJ predictor 2.9 mm. For comparison, it can be noted that the average width of the track is 23 mm allowing an 11.5 mm error margin from the center of the track.

The magnitude of the average error for the pure EKF approach is 27% larger than that of the MJ approach. This may not be very large, but there is a difference in the quality of the error. Figure 7 shows both predictions as compared to the real signal and the delayed signal of the x coordinate. With the MJ curves there is only a small amount of oscillations as the input slows down after a period of higher velocity. This is where one of the strengths of the MJ-based prediction is shown as it enforces an MJ trajectory, which by construction does not oscillate. In this particular setup, this is an important property, as high frequency oscillations may be reproduced at lower frequencies but with higher amplitudes by the manipulator controller.

Note that while Figure 7 shows two entire superimposed MJ submotions, in online usage, due to the time dely, only the later part of each submotion would be available as a prediction. In the figure, the arrows pointing from the left show where each prediction would become available, and the arrows from the right show where the prediction would loose validity as a new submotion could possible become dominant. Since the MJ trajectory is also tracked with a Kalman filter, the influence of the MJ prediction remains for some time after this point.

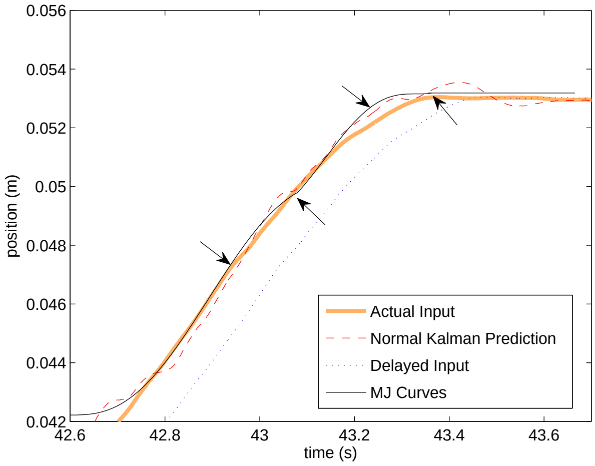


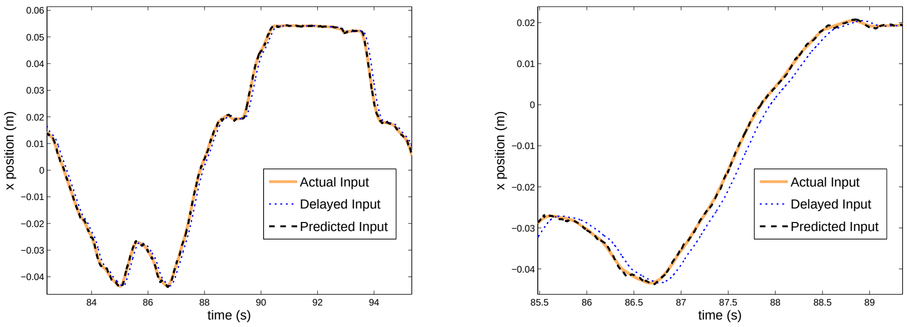
Figure 7: A comparison between MJ curves and pure EKF prediction. The MJ trajectory in the plot contains two submotions. The arrows pointing from the left show where each submotion would become available to an online prediction system, the arrows pointing from the right show where each prediction would loose validity, as it would be possible for a new submotion to dominate the trajectory.

An example of the final result of the compound prediction, using MJ predictions when these are available, and simple EKF type extrapolation from the delayed signal when no MJ predictions are available is shown in Figure 8. This plot shows the x coordinate as Track 4 is traced one lap clockwise. The transients caused by switching predictor depending on the availability of MJ predictions are effectively damped out by the system and are not noticeable to the operator.

4.5. *Experimental Procedure*

For evaluation purposes, experiments were carried out where subjects were asked to trace the path clockwise. First 10 laps for practice, then 10 laps that were recorded. This was repeated 3 times, with a different setting each time. Possible settings were:

A No communication delays added, no predictions. This is the control case that the other settings are compared to.



(a) Predictor performance with 100 ms delay.

(b) Predictor performance with 100 ms delay, closeup.

Figure 8: Example performance of prediction.

B 100 ms round-trip time delay added, no predictions. This case is used to determine the effects of delays on the completion time of this particular task.

C 100 ms round-trip time delay added, input prediction system active. This case is used to determine how well the predictions can cancel the negative effects of the time delay.

The order in which the three settings were presented to the subjects was permuted between subjects to cancel out overall learning effects. The subjects were told that “three different settings” would be evaluated, but not what those settings were. They were asked to trace as fast as they could without making mistakes. A total of 12 subjects were used. The subjects were all male, ages 23–31, and did not have prior experience with the experimental setup.

We measure “Objective telepresence”, as defined in [55]. This means that the task completion time is used as the performance measure. Of the measured tries, the times for successful trials are included in the analysis. Results for subjects who did not successfully complete at least 5 laps with each setting was not considered. Success is defined as not drawing outside the borders of the track during the lap. For this type of task, there exists an inherent trade-off between minimizing errors and completion time. By excluding results from tries with errors, the remaining data should have a lower variation in how this trade-off is balanced.

It should be noted that even in setting A, with no added communication delay, there are some inherent delays in the system. The time needed to process input data and calculate predictions is less than one millisecond, but the robot processes commands at 100 Hz, giving the “current” command a delay that varies from 0 to 10 ms. Similarily, the video display is updated at 15 Hz, so that the image being displayed will have a varying delay of 0–67 ms, plus the overhead for image acquisition and processing. None of these delays were treated in any special way, but were added to the total delay of the system for all three settings.

4.6. Results

For all subjects but one, the median completion time was lower for setting C, with the predictor, than for setting B. One subject even performed times that where better than the control setting A, See Figure 9. This is possibly the effect of some overall learning, with subjects performing relatively better on the last setting they were presented with. A small overall trend for

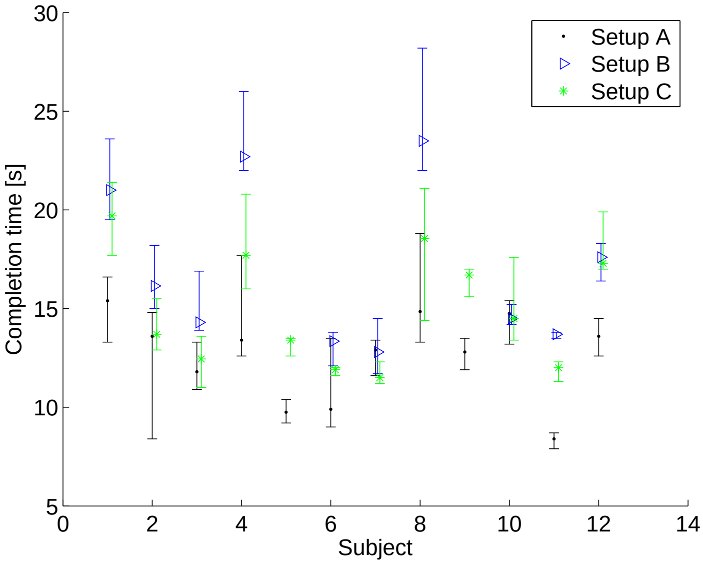


Figure 9: Task completion times for the 12 subjects. The plot shows median completion times and 90% confidence intervals. Note that subjects 5 and 9 did not complete the task for setup B, and are not included when calculating statistics.

improvement over time was observed. When corrected for setting differences, the subjects were on average 2% faster on the second trial, and 6% faster on the third. The average completion time was 12.86 s for setup A, 19.3% longer for setup C and 40.0% longer for setup B. Performing a Student's T-test on the results, and setting the significance level to 0.05, we find that the performance with setup A is significantly better than those of both setup B (with $p = 1.83 \cdot 10^{-12}$) and setup C (with $p = 3.54 \cdot 10^{-6}$). The performance with setup C was significantly better than that of setup B (with $p = 0.000187$).

Two subjects, 5 and 9, were only able to complete one successful lap with setting B, and are therefore excluded from these results. All others were able to complete at least 5 successful laps with each setting. Comments from several subjects stated that the main difficulties they perceived with settings B and C were oscillations, most subjects did not explicitly notice that there was a time delay.

5. Conclusions

In the present paper, we have described a method that uses predictions of user input to cancel the negative effects of time delays on free-motion teleoperation. The experimental results show that this is a valid approach, as the performance was significantly better when compared to a control case with the same delay but no predictive system. The main strength of this approach is that we only need a model of the master side of the telerobotic system, while the slave side can be left completely unmodeled. This means that once we have designed the master system, we can deploy a slave system in novel environments to perform novel tasks without the need for recalibration or redesign of the control system. Furthermore, once the predictive system is calibrated for the master system, the slave system can be replaced without having to recalibrate the predictive system, as long as the new slave system works in the same coordinate space and does not significantly differ in performance. This also allows for easy-to-deploy remote sensors, such as arbitrarily positioned uncalibrated cameras.

In the implementation presented here, the prediction system relies on MJ models, but any model that sufficiently well describes human motion could conceivably be used.

6. Limitations and Future Work

One strength of using the relatively simple MJ model for predicting human input is that since we do not use any knowledge of the particular task being performed, it should generalize to most possible vision guided tasks.

However, there are several limits to this approach. The first is that since new MJ submotions can be generated as often as every 100 ms, it is not realistic to make predictions much further into the future than this, as there is a real possibility that the operator will be moving along a trajectory that we have not yet detected. A 100 ms round-trip delay is comparable to the internet round-trip delay between Stockholm, Sweden and New York City, USA. However, when using the Internet, time delays can behave stochastically, something that would have to be considered and dealt with in an implementation over an internet connection.

A potential use of this approach even in the presence of longer delays, is where we have a plausible model for predicting the feedback from the remote environment as in traditional model-based control, but where non-linearities or unmodeled events make it difficult to predict over as long time delays as we would need. It is possible that in combination with the input predictor model described in the present paper, one could successfully predict both input and measurements, and thus bridge larger time delays than when just predicting one of the two. How this could be done remains to be studied.

Another limit is that the MJ model assumes free motion. Its assumptions do not hold under the presence of outer forces, e.g. contact forces. Therefore, this approach is not applicable without modification for tasks that require haptic interaction with the remote environment. However, as shown in Section 4.4, even without the MJ predictions, a pure EKF predictor has similar magnitude of tracking errors. It remains as future work to examine how this type of approach can be extended to tasks that include contact. It is possible that hybrid approaches, utilizing parts of this approach and parts of traditional delay-handling approaches may be the solution. One could conceive of using simple remote autonomy to avoid unwanted collisions. If an operator's typical reaction to contact forces is known, it could be possible to incorporate this in a predictor at the remote site.

Finally, there this is yet no treatment of stability issues for the MJ prediction approach. Although empirical trials have not shown any unstable behavior, a more rigorous analysis remains to be done.

Acknowledgments

The authors wish to thank professor Henrik Christensen and the Swedish Foundation for Strategic Research, through the Centre for Autonomous Systems, for providing the funding for this project.

References

[1] K. U. Smith, Delayed Sensory Feedback and Behavior, W. B. Saunders Co, Philadelphia, Pa., 1962.

[2] W. Ferrell, Remote manipulation with transmission delay (performance of simple and complex tasks affected by inserting transmission delay between master and slave elements of remote manipulator), IEEE Transactions on Human Factors in Electronics 6 (1965) 24–32.

[3] T. B. Sheridan, W. Ferrell, Remote manipulative control with transmission delay, IEEE Transactions on Human Factors in Electronics 4 (1963) 25–29.

[4] W. Ferrell, Delay force feedback, IEEE Transactions on Human Factors in Electronics 7 (1966) 449–455.

[5] S. Hirche, M. Buss, Human perceived transparency with time delay, STAR Advances in Telerobotics 31 (2007) 191–209.

[6] T. B. Sheridan, Telerobotics, Automatica 25 (4) (1989) 487–507.

[7] P. Arcara, C. Melchiorri, Control schemes for teleoperation with time delay: A comparative study, Robotics and Autonomous Systems 38 (2002) 49–64.

[8] P. F. Hokayem, M. W. Spong, Bilateral teleoperation: An historical survey, Automatica 42 (2006) 2035–2057.

[9] A. B. Schwartz, D. W. Moran, G. A. Reina, Differential representation of perception and action in the frontal cortex, Science 303 (2004) 380–384.

[10] A. Berthoz, The Brain's sense of movement, Perspectives in Cognitive Science, Harvard University Press, London, UK, 2000.

[11] M. Bratt, C. Smith, H. I. Christensen, Minimum jerk based prediction of user actions for a ball catching task, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE/RSJ, San Diego, Ca, USA, 2007, pp. 2710–2716.

[12] C. Smith, M. Bratt, H. I. Christensen, Teleoperation for a ballcatching task with significant dynamics, Neural Networks, Special Issue on Robotics and Neuroscience 24 (2008) 604–620.

[13] C. Smith, H. I. Christensen, Wiimote robot control using human motion models, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE/RSJ, Saint Louis, USA, 2009, pp. 5509–5515.

[14] T. B. Sheridan, Space teleoperation through time delay: Review and prognosis, IEEE Transactions on Robotics and Automation 9 (5) (1993) 592–606.

[15] D. Yoerger, J.-J. Slotine, Supervisory control architecture for underwater teleoperation, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Vol. 4, 1987, pp. 2068–2073.

[16] T. B. Sheridan, Telerobotics, automation, and human supervisory control, MIT Press, 1992.

[17] K. Goldberg, M. Mascha, S. Gentner, N. Rothenberg, C. Sutter, J. Wiegley, Desktop teleoperation via the world wide web, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Vol. 1, 1995, pp. 654–659.

[18] K. Brady, T.-J. Tarn, Internet-based teleoperation, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Seoul, Korea, 2001, pp. 644–649.

[19] J. J. Biesiadecki, P. C. Leger, M. W. Maimone, Tradeoffs between directed and autonomous driving on the mars exploration rovers, The International Journal of Robotics Research 26 (1) (2007) 91–104.

[20] B. K. Muirhead, Mars rovers, past and future, in: Proceedings of the IEEE Aerospace Conference, Vol. 1, 2004, pp. 128–134.

[21] G. Hirzinger, K. Landzettel, C. Fagerer, Telerobotics with large time delays, in: Proceedings of the International Conference on Intelligent Robots and Systems (IROS), 1994, pp. 571–578.

[22] M. R. Stein, R. P. Paul, Operator interaction, for time-delayed teleoperation, with a behavior-based controller, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Vol. 1, IEEE, 1994, pp. 231–236.

[23] A. K. Bejczy, W. S. Kim, S. C. Venema, The phantom robot: Predictive displays for teleoperation with time delay, in: Proceedings of IEEE International Conference on Robotics and Automation (ICRA), Vol. 1, 1990, pp. 546–551.

[24] T. Kotoku, A predictive display with force feedback and its application to remote manipulation system with transmission time delay, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Raleigh, NC, 1992, pp. 239–246.

[25] A. K. Bejczy, P. Fiorini, W. S. Kim, P. Schenker, Toward integrated operator interface for advanced teleoperation under time-delay, in: Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems (IROS), Vol. 1, 1994, pp. 563–570.

[26] G. Hirzinger, J. Heindl, K. Landzettel, Predictive and knowledge-based telerobotic control concepts, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Vol. 4, 1989, pp. 1768–1777.

[27] A. Kheddar, E.-S. Neo, R. Tadakuma, K. Yokoi, Advances in Telerobotics, Vol. 31 of STAR, Springer-Verlag, 2007, Ch. Enhanced Teleoperation Through Virtual Reality Techniques, pp. 139–159.

[28] A. Kheddar, C. Tzafestas, P. Coiffet, The hidden robot concept — high level abstraction teleoperation, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 1997, pp. 1818–1824.

[29] M. Hernando, E. Gambao, Advances in Telerobotics, Vol. 31 of STAR, Springer-Verlag, 2007, Ch. Teleprogramming: Capturing the Intention of the Human Operator, pp. 303–320.

[30] P. Mitra, G. Niemeyer, Model-mediated telemanipulation, The International Journal of Robotics Research 27 (2) (2008) 253–262.

[31] S. Clarke, G. Schillhuber, M. F. Zaeh, H. Ulbrich, Prediction-based methods for teleoperation across delayed networks, Multimedia Systems 13 (2008) 253–261.

[32] C. Smith, H. I. Christensen, A minimum jerk predictor for teleoperation with variable time delay, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE/RSJ, Saint Louis, USA, 2009, pp. 5621–5627.

[33] N. Hogan, An organizing principle for a class of voluntary movements, Journal of Neuroscience 4 (1984) 2745–2754.

[34] T. Flash, N. Hogan, The coordination of arm movements: An experimentally confirmed mathematical model, The Journal of Neuroscience 5 (7) (1985) 1688–1703.

[35] A. Hauck, M. Sorg, T. Schenk, What can be learned from human reach-to-grasp movements for the design of robotic hand-eye systems?, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Vol. 4, Detroit, Michigan, USA, 1999, pp. 2521–2526.

[36] N. Krüger, Entwicklung einer humanoiden ball-fang-strategie für ein roboter-hand-arm-system, Master's thesis, Technischen Universität München (2006).

[37] N. Miller, R. Mann, Detecting hand-ball events in video sequences, in: Proceedings of the Canadian Conference on Computer and Robot Vision (CRV), 2008, pp. 139–146.

[38] M. Kawato, Trajectory formation in arm movements: Minimization principles and procedures, in: H. N. Zelaznik (Ed.), Advances in Motor Learning and Control, Human Kinetics, Human Kinetics Publishers, Chanpaign Illinois, 1996, pp. 225–259.

[39] S. E. Engelbrecht, Minimum principles in motor control, Journal of Mathematical Psychology 45 (3) (2001) 497–542.

[40] N. Hogan, T. Flash, Moving gracefully: quantitative theories of motor coordination, Trends in Neurosciences 10 (4) (1987) 170–174.

[41] T. Flash, E. Henis, Arm trajectory modification during reaching towards visual targets, Journal of Cognitive Neuroscience 3 (1991) 220–230.

[42] P. Viviano, T. Flash, Minimum-jerk, two-thirds power law, and isochrony: converging approaches to movement planning, Journal of Experimental Psychology: Human Perception and Performance 21 (1) (1995) 32–53.

[43] T. Gat-Falik, T. Flash, The superposition strategy for arm trajectory modification in robotic manipulators, IEEE Transactions on Systems, Man, And Cybernetics–Part B: Cybernetics 29 (1) (1999) 83–95.

[44] T. E. Milner, A model for the generation of movements requiring endpoint precision, Neuroscience 49 (2) (1992) 487–496.

[45] K. J. Kyriakopoulos, G. N. Saridis, Minimum jerk path generation, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Vol. 1, Philadelphia, PA, USA, 1988, pp. 364–369.

[46] M. Kawato, Y. Uno, M. Isobe, R. Suzuki, Hierarchical neural network model for voluntary movement with application to robotics, IEEE Control Systems Magazine 8 (2) (1988) 8–15.

[47] A. Piazzi, A. Visioli, Global minimum-jerk trajectory planning of robot manipulators, IEEE Transactions on Industrial Electronics 47 (1) (2000) 140–149.

[48] H. Krebs, J. Palazzolo, L. Dipietro, M. Ferraro, J. Krol, K. Rannekleiv, B. Volpe, N. Hogan, Rehabilitation robotics: Performance-based progressive robot-assisted therapy, Journal of Autonomous Robots 15 (1) (2003) 7–20.

[49] E. L. Secco, A. V. G. Magenes, Minimum jerk motion planning for a prosthetic finger, Journal of Robotic Systems 21 (7) (2004) 361–368.

[50] J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, H. Inoue, Robotics Research, Vol. 15 of Springer Tracts in Advanced Robotics (STAR), Springer Berlin / Heidelberg, 2005, Ch. Motion Planning for Humanoid Robots, pp. 365–374.

[51] C. Weber, V. Nitsch, U. Unterhinninghofen, B. Farber, M. Buss, Position and force augmentation in a telepresence system and their effects on perceived realism, in: Proceedings of the World Haptics Third Joint EuroHaptics conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC), IEEE Computer Society, Washington, DC, USA, 2009, pp. 226–231.

[52] B. Corteville, E. Aertbelien, H. Bruyninckx, J. D. Schutter, Human-inspired robot assistant for fast point-to-point movements, in: Proceedings of IEEE International Conference on Robotics and Automation, ICRA, 2007, pp. 3639–3644.

[53] G. Welch, G. Bishop, An introduction to the kalman filter, Tech. Rep. TR 95-041, Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC (Apr 2004).

[54] C. Smith, H. I. Christensen, Using COTS to construct a high performance robot arm, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), IEEE, Rome, IT, 2007, pp. 4056–4063.

[55] R. Aracil, M. Buss, S. Cobos, M. Ferre, S. Hirche, M. Kuschel, A. Peer, Advances in Telerobotics, Vol. 31 of STAR, Springer-Verlag, 2007, Ch. The Human Role in Telerobotics, pp. 11–24.