2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

Virtual Environment Teleoperation of a Hydraulic Forestry Crane

Simon Westerberg*, Ian R. Manchester*, Uwe Mettin*, Pedro La Hera*, Anton Shiriaev*,†

***Abstract*— A teleoperation system has been developed for a hydraulic crane, of the type used on a *forwarder* vehicle, which travels off-road and collects logs cut by a *harvester*. The system developed consists of a 3D virtual environment, which allows the user to input a desired position for the crane tip using either the mouse or a joystick. The desired position is then transmitted (via UDP/IP) to a local control system. The crane is a redundant manipulator, so movements of the individual links are calculated using a pseudoinverse method, and controlled using PIDs with friction compensation. Encoder data from the crane links are continuously sent back to the user side, and the crane's movement is visualized in the virtual environment. The system has been tested on a real forwarder crane, experimental results and a video of the system's performance are provided.**

I. Introduction

The Swedish forestry industry has a long-term goal of developing autonomous and semi-autonomous forestry vehicles [1], [2].

There are two main types of off-road vehicle used in the forestry industry: a *harvester*, which fells and delimbs the trees, and cuts the trunk into logs of a predetermined size, and a *forwarder*, which collects the logs in a tray, and carries them to the nearest road for collection. A forwarder is shown in Figure 1. Control of the forwarder can be divided into two distinct tasks: navigation of the vehicle itself, and operation of the on-board crane. It is the latter task which we consider in this paper.

This paper describes implementation and experimental testing of a virtual-environment-based remote control and visualization system for such a crane. The system presented here is to be considered a partial solution. At present, the virtual environment is constructed based on information from the sensors on board the crane, and does not include any dynamic sensing of objects within the environment. The latter is currently under development.

A. Virtual environment–assisted teleoperation

Virtual environments systems have been successfully used for teleoperational tasks. Gravez *et al.* [3] use a virtual model of a hydraulic arm and its surroundings to perform different tasks in a radioactive environment. Their system allows visual feedback as well as high-level motion control. There are several other reports of successful implementation of virtual-environment-based teleoperation systems for other classes of systems in the literature, see, e.g. [4], [5], [6] and references therein.

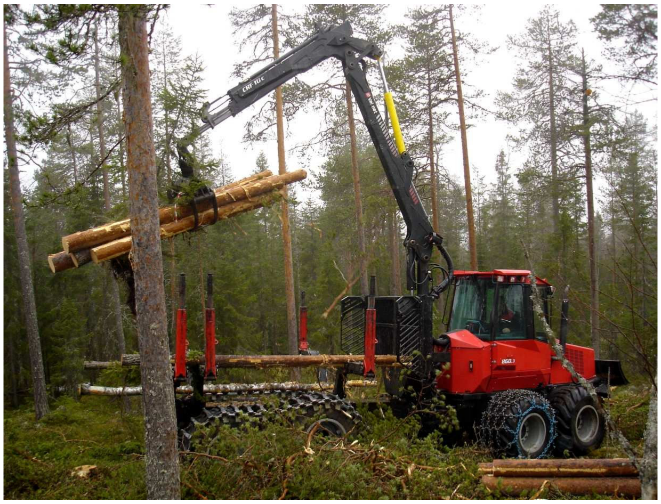

Fig. 1. A forwarder: the Komatsu 860.1.

The use of virtual environments in teleoperation has several advantages compared to teleoperation with streaming video as visual feedback [7]. One obvious advantage is the possibility to change the view of the virtual environment. When video is used for feedback, the physical cameras need to be attached to the machine and placed where they do not risk to be damaged by collisions. The virtual environment, however, can be fused from many different sensors at different locations, and the virtual cameras can be placed in any location and moved around freely. This gives the user a good overview and also the possibility to adapt the view to the task that is currently performed. E.g., if an obstacle is present, the operator can view the scene from another direction so that the line-of-sight is not blocked by the obstacle.

Additionally, the communication of streaming video requires large bandwidth in order to have reasonable image quality. This is an important issue in the forest, where Internet connections have a very limited capacity. With virtual environments, the information flow between the machine and the operator can be restricted to the most relevant features of the environment, such as the position and size of a log, whereas less important details in e.g. textures can be discarded.

Another advantage with virtual environments is the possibility to easily add visual operator assisting features. For instance, irrelevant details about the environment can be hidden, while more important information can be emphasized, in order to let the operator focus on important information.

This work has been supported by the Centre for Intelligent Off-Road Vehicles at Umeå University, Komatsu Forest AB, and the Kempe Foundation.

* Department of Applied Physics and Electronics, Umeå University, SE-901 87, Sweden.

† Department of Engineering Cybernetics, Norwegian University of Science and Technology, N-7491 Trondheim, Norway.

978-1-4244-1647-9/08/$25.00 ©2008 IEEE.

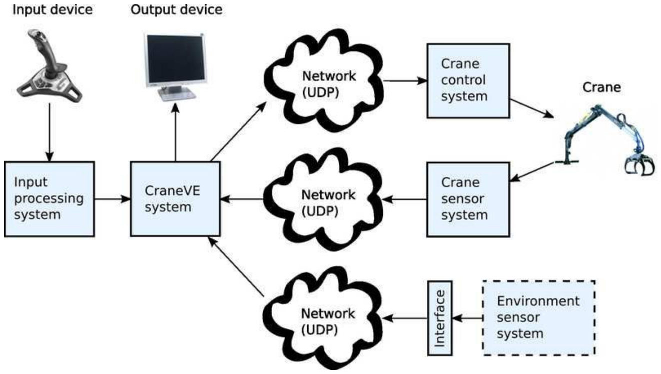

Fig. 2. The system architecture

Furthermore, information that is not visual in the physical world, like optimal crane-arm trajectories, can be added. By changing the modality of the information, non-visual object properties can be made visible in the virtual environment. E.g., color can be used to represent the temperature of an object.

This makes virtual environment useful not only for teleoperation. Even in normal operation, a virtual environment can give the operator information that is hard or impossible to see with his own eyes.

A further goal is to make the control of the crane more intuitive. In current forest machines, the driver must directly control six hydraulic valves using two 3-degree-of-freedom joysticks. Each joystick motion corresponds to one crane function, e.g. increasing the angle between two links, or extending the telescopic arm, etc. Learning to control the crane tip efficiently takes years of training [1]. A more intuitive way to control the crane is to directly specify the position of the crane tip, and generate the required hydraulics inputs by computer control.

When certain tasks are automated and thus performed without human intervention, a virtual environment is useful for monitoring the machine's actions. Furthermore, even if a crane is completely controlled by a driver, a virtual environment allows a forestry company to monitor the operation of a large fleet of vehicles in an intuitive manner.

II. System Architecture

Figure 2 shows the architecture of the teleoperation system. The system consists of two subsystems, connected through an IP network, either a LAN or the Internet.

One subsystem is at the location of the crane, with physical connections to sensors and actuators. Besides the sensor-equipped crane and the real-time device, this subsystem contains an ordinary PC. The PC is responsible for the communication between the real-time system and the network.

The second subsystem is at the location of the operator. It provides the operator with visual feedback, as well as with an interface for controlling the crane. For visualization and user interface, the CraneVE software was developed. The software is written in C++ and uses a scene graph as the virtual environment data structure, using the OpenSceneGraph library. This subsystem consists of a PC that runs the CraneVE software as well as the software that reads input from the user. The operator uses a joystick and/or a mouse to enter the input to the system. The output consists of visualization on a monitor.

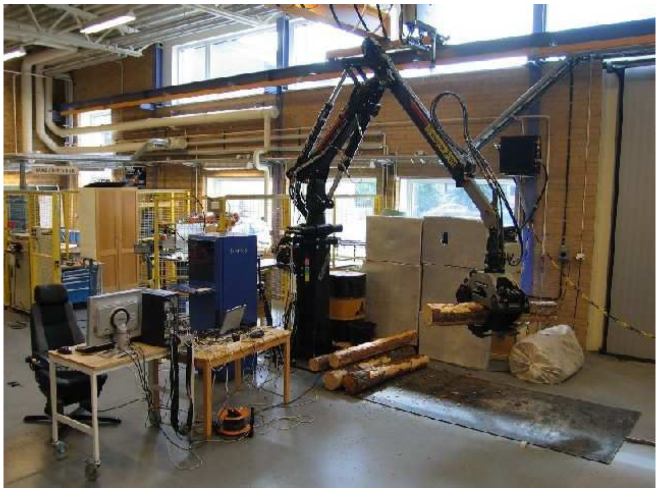

Fig. 3. The crane laboratory at Umeå University.

The crane PC and the operator PC are connected through an IP network. Data transmissions are sent using UDP, which has a low overhead and reduces the bandwidth use. In one direction, the transmitted data consists of sensor data, i.e. link angles and telescope extension. In the other direction, it consists of 3D positions that are used to control the crane. Since only a few bytes are used to represent the transmitted information, the bandwidth requirement is low, even if the sensor data are updated frequently.

III. Crane Control Hardware

The following experimental equipment has been installed at a laboratory at Umeå University:

- CRANAB 370RCR hydraulic forwarder crane,
- three joint-position encoders, 4000 pulses per turn, the telescope extension is measured by an encoder connected to a retracting wire,
- dSPACE MicroAutoBox real-time prototyping hardware, with on-board PowerPC CPU,
- RapidPro power amplification unit and signal conditioners.

The 370RCR crane is somewhat smaller than most cranes used on real forwarders, but was chosen due to space restrictions in the laboratory. It's operational principle and dynamics are, however, very similar. The experimental setup is shown in Figure 3.

IV. User Interface

The user interface is shown in Figure 4. The operator is presented with a view of the virtual model of the crane, as well as of the surrounding environment.

On the left side is the view of the virtual environment. The viewpoint can be moved around freely by the user using the

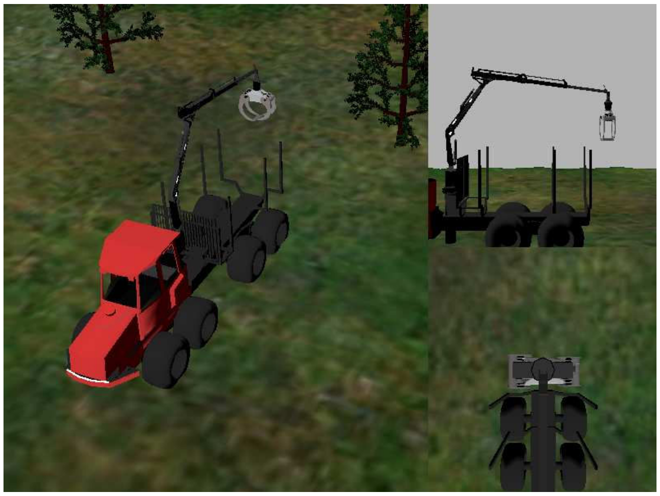

Fig. 4. The user interface for the crane teleoperation system.

mouse: the operator can move and rotate the camera, as well as zoom in and out. This allows the user to zoom out for an overview, or focus on details of the environment that are important for a certain task.

The other two view panes have positions and directions that are fixed relative to the crane. That is, for any configuration (rotation and translation) of the crane, the cameras are always directed towards the crane. The top right view pane shows the crane from the side, while the bottom right view pane shows the crane from above. This allows the operator to always have a detailed view over the crane and the environment near the crane.

Virtual environments are suited for the introduction of operator assisting features. As one example of this, visualization of the crane workspace has been implemented. This can be seen in Figure 5.

A. Target Specification

The user can specify the target position of the crane-tip in a number of different ways. One way is to use the mouse to click on the top right camera view. This will result in a position in the vertical plane. The second method is to use an ordinary joystick to move a pointer in the virtual environment. A 2-axis joystick is used and two buttons are used to simulate a third axis for full 3D motion. When the user presses a button, the current position of the pointer is sent to the control system. The crane motion that is required to move the crane tip from the current crane position to the target point is then calculated. The motion can be calculated to be optimal with regard to e.g. speed or energy efficiency. In the following section we describe a simple motion-planning algorithm which can be calculated on-line. If several subsequent target points are specified, the target points become waypoints on a target trajectory. The crane tip will then visit each waypoint in order.

The next step is to transfer more responsibility from the operator to the control system. This means that certain tasks or processes are automated and performed autonomously by the system. As an example of an activity where the computer has increased responsibility, a simple collision-avoidance algorithm has been implemented in the CraneVE software. When collision avoidance is enabled, the system performs collision detection between the user-specified trajectory and the objects in the scene. If the user specified trajectory is blocked by an object, a new trajectory is calculated such that the crane tip passes above the object and avoids the collision. The collision avoidance algorithm is explained in Figure 6.

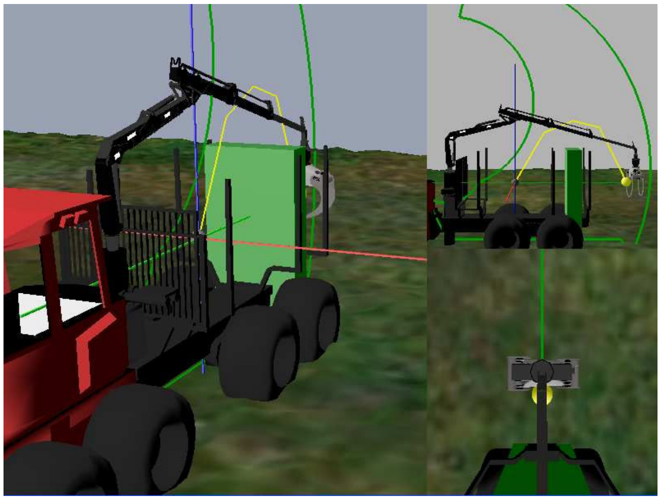

Fig. 5. High-level control with collision avoidance. The yellow path shows the trajectory generated by the collision avoidance algorithm in order to avoid the obstacle (the green box) while moving towards the tray.

V. Crane-Side Control

The configuration of the crane is represented by a three-dimensional vector, $q = [\theta_2, \theta_3, d]'$. From basic trigonometry one can derive the forward kinematics equations, giving the Cartesian coordinates from a known link configuration:

$$c := [x, y]' = f(q). \tag{1}$$

However, the low-level control objective is to reach a particular point in two-dimensional space, i.e. a particular Cartesian coordinate $c_\star = [x_\star, y_\star]'$. Since there are three degrees of freedom in the crane, this is an underdetermined problem, and hence has infinitely many solutions. We now describe the method we use to choose a corresponding $q_\star$ such that $f(q_\star) = c_\star$, it is conceptually similar to the methods presented in [8], [9], [10].

Differentiating the above equation, we get:

$$\dot{c} = F(q)\dot{q}, \tag{2}$$

where $F(q)$ is the Jacobian of $f(q)$. We wish to calculate a movement in q that would correct the current positioning error $c_\star - c$.

Again, generating $\dot{q}$ from $\dot{c}$ is an underdetermined problem, and has infinitely many solutions. We choose to find the solution which minimizes a configuration-dependent cost function of the form:

$$J = \dot{q}' P(q, \dot{c})^{-1} \dot{q}, \tag{3}$$

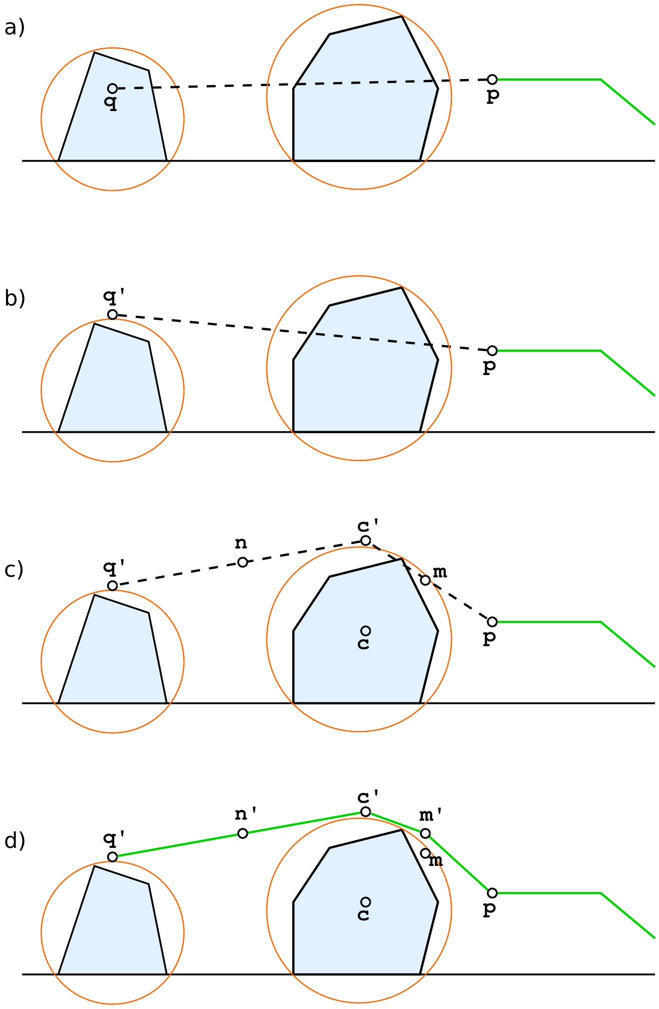

Fig. 6. The collision avoidance algorithm. (a) Assume that p is the end point of the current path. Let q be a new target point defined by the operator. (b) If q collides with an object, find a safe point q' above the bounding sphere of the object. (c) If the path $p \longrightarrow q'$ collides with an object, find a safe point c' above the center of the object. Let m and n be the middle point of each sub-path $p \rightarrow c'$ and $c' \rightarrow q'$ respectively. (d) Find safe points m' and n' above m and n such that they are outside of the bounding sphere of the colliding object. The final path will be $p \rightarrow m' \rightarrow c' \rightarrow n' \rightarrow q'$.

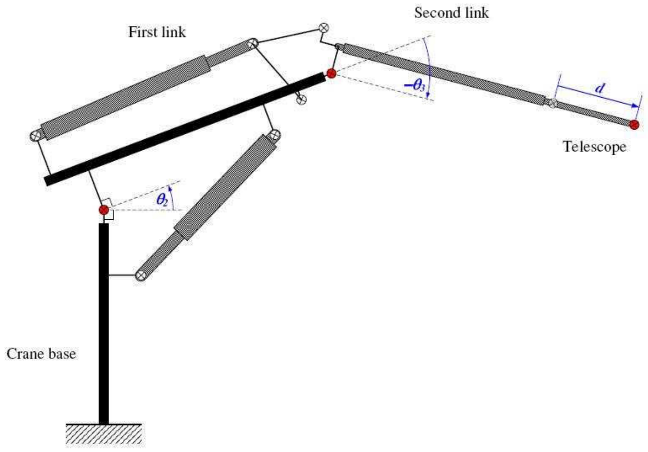

Fig. 7. Basic geometry of the crane.

where $P(q, \dot{c})$ is a positive definite matrix for all q and $\dot{c}$.

This choice of cost function is quite simplistic, but it has the advantage that the resulting solution can be calculated analytically:

$$\dot{q} = P(q, \dot{c}) F(q)' [F(q) P(q, \dot{c}) F(q)']^{-1} \dot{c}. \tag{4}$$

The matrix FPF' is two-by-two, so it is easy to calculate the right-hand side of (4) in terms of the elements of P and F. The formulas are too lengthy to present here, but they are not difficult for real-time computation, in comparison to an iterative scheme that another choice of cost function would require.

In our system, $P(q, \dot{c})$ is chosen to be of the following form:

$$P(q, \dot{c}) = \begin{bmatrix} w_1 p_1(q, \dot{c}) & 0 & 0 \\ 0 & w_2 p_2(q, \dot{c}) & 0 \\ 0 & 0 & w_3 p_3(q, \dot{c}) \end{bmatrix}. \tag{5}$$

The values w_1, w_2, w_3 represent the desired contribution from each link to the overall movement. If one of them is zero, the corresponding link is stationary and the other two links are used to generate the motion.

The functions $p_1(q, \dot{c}), p_2(q, \dot{c}), p_3(q, \dot{c})$ are there for "protection": if one of the links is approaching the limit of its working range, its contribution should be smoothly reduced towards zero, and the other links should take over. We now describe how they are calculated.

For each $i = 1, 2, 3$, let $q_i^{\max}$ and $q_i^{\min}$ be the maximum and minimum allowed values of q_i, let δ_i be a constant smaller than $(q_i^{\max} - q_i^{\min})/2$, and let $\phi_i(q)$ be a smooth function such that:

- $\phi_i(q_i^{\min}) = 0$,
- $\phi_i(q_i^{\max}) = 0$,
- $\phi_i(s) = 1$ for all $s \in [q_i^{\min} + \delta_i, q_i^{\max} - \delta_i]$

Then $p_i(q, \dot{c})$ is calculated like so:

Algorithm 1:

```
1) Set $p_1(q, \dot{c}) = p_2(q, \dot{c}) = p_3(q, \dot{c}) = 1$,
2) calculate $\dot{q}_\star^0$ from Equations (4) and (5),
3) for each $i = 1, 2, 3$, if
     • $q_i^{\max} - q_i < \delta_i$ and $\dot{q}_{\star,i}^0 > 0$,or
     • $q_i - q_i^{\min} < \delta_i$ and $\dot{q}_{\star,i}^0 < 0$
   then $p_i(q, \dot{c}) = \phi_i(q_i)$.
```

□

We have then the following closed-loop control algorithm, which generates control signals u_1, u_2, u_3 from a desired Cartesian set-point $c_\star$ and the measurements q_1, q_2, q_3:

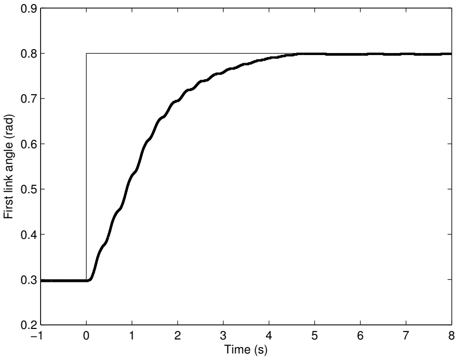

Fig. 8. Step response for the first link, θ_2.

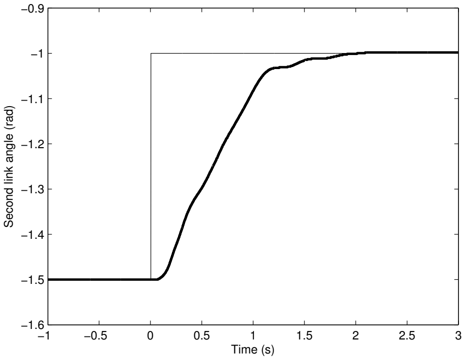

Fig. 9. Step response for the second link, θ_3.

Algorithm 2:

$$
\begin{aligned}
q_\star(0) &= q \\
\dot{q}_\star &= kP(q,\dot{c})F(q)'[F(q)P(q,\dot{c})F(q)']^{-1}(c_\star - f(q_\star)), \\
u_1 &= C_1(s)[q_1 - q_{\star,1}], \\
u_2 &= C_2(s)[q_2 - q_{\star,2}], \\
u_3 &= C_3(s)[q_3 - q_{\star,3}].
\end{aligned}
\tag{6}
$$

where k is a large positive gain, and $C_1(s), C_2(s), C_3(s)$ are PID controllers tuned for each link of the crane, and $P(q,\dot{c})$ is calculated at each sample time from Algorithm 1.

□

Each control signal is further modified to compensate for the large Coulomb friction present in the hydraulics actuators, see [11] for details, see also [12] for similar work.

The vector $q_\star$ quickly converges to a feasible target, satisfying $f(q_\star) = c_\star$. Following this, the individual link positions q_1, q_2, q_3 are driven more slowly to the corresponding values from $q_\star$.

VI. Experimental Results

A. Individual Link Positioning

In Figures 8, 9, and 10 we see the step responses of the individual links, as controlled by the PID controllers $C_1(s), C_2(s), C_3(s)$ along with the friction compensators.

It is clear that the first link is by far the slowest, which is reasonable since it must drive by far the greatest mass. The second link and the telescope converge at similar rates, but the telescope is much smoother and easier to control, since it has the least mass to drive.

We chose the control contributions in Algorithm 2 such that the links with the best control are used most. The following values were found to be reasonable in experiments: $w_1 = 0.5, w_2 = 1, w_3 = 1.5$.

Overall, we were able to make all links converge to the desired positions with zero steady-state errors, sufficient speeds, and without oscillations.

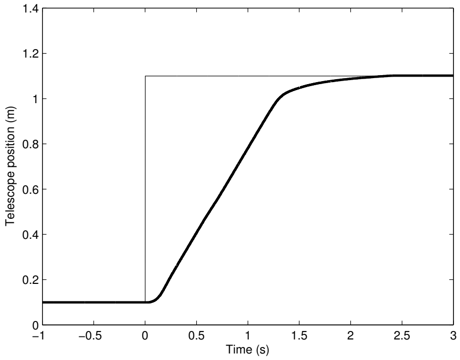

Fig. 10. Step response for the telescope, d.

B. Teleoperation Experiment

Accompanying this paper is a video of the teleoperation system in action (a higher resolution version is also available online [13]). The user clicks the mouse at a sequence of points in the virtual environment, and the crane can be observed moving to those points. Figure 11 shows the x and y setpoints and crane trajectories from the same experiment as featured in the video.

It can be seen that in general the system converges quickly (in two to four seconds) to the desired setpoint.

There is an interesting effect visible at around 17 seconds, and again at 30 seconds. When the desired motion is large in one direction, but small or zero in the other, we still can observe a noticable deviation in the latter dimension. This is because, with a large k in Algorithm 2, $q_\star$ converges very quickly to its final value satisfying $f(q_\star) = c_\star$, and then each link is driven to its desired value at a slower rate. In particular, the first link is much slower than the other two. This means that the end-effector will not in general follow a straight line path between two points in Cartesian space.

VII. Conclusions and Future Work

The virtual environment-based teleoperation system reported in this article can be considered proof of concept. The system is also useful as a tool for further research and development in the area of remote crane control.

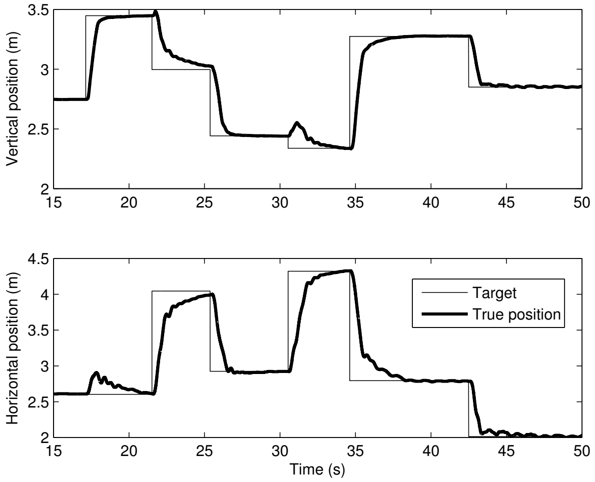

Fig. 11. Horizontal and vertical positioning during the teleoperation experiment.

However, the system is far from ready to be used in real forest machines. A virtual environment-assisted teleoperation system depends on the virtual representation being accurate, and the main problem right now is the lack of knowledge about obstacles in the environment. In order to correct this, we need a reliable method to detect and classify different objects near the crane. The classification problem is quite difficult: the system must be able to distinguish between rocks, trees, logs, branches and mud. It also needs to reliably detect any human inside the safety area.

A number of different technologies are under investigation to address this problem. Different sensor systems like stereo cameras, laser scanners and structured light systems [14] can be used to extract information from the environment.

Another direction of our current work is to implement and test a similar system on a real forwarder crane, provided by Komatsu AB. Due to reasons of confidentiality, we are not able to publish the results of these experiments, but a video showing an early test is available online [15].

References

[1] U. Hallonborg, "Förarlösa skogsmaskiner kan bli lönsamma (Unmanned forestry machines can be competitive)" (in Swedish), *Skogforsk Results*, No 9, 2003.

[2] M. Brander, D. Eriksson, B. Löfgren, "Automation of knuckleboom work can increase productivity", *Skogforsk Results*, No 4, 2004.

[3] P. Gravez, C. Leroux, M. Irving, L. Galbiati, A. Raneda, M. Siuko, D. Maisonnier, and J. D. Palmer, "Model-based remote handling with the MAESTRO hydraulic manipulator", *Fusion Engineering and Design*, vol. 69, no. 1, pp. 147–152, 2003.

[4] L. Flückiger, L. Piguet, and C. Baur, "Generic robotic kinematic generator for virtual environment interfaces", in *SPIE Telemanipulator and Telepresence Technologies*, Vol. 2901, pp. 186–195, 1996.

[5] R. Safaric, R. M. Parkin, C. A. Czarnecki, and D. W. Calkin, "Virtual environment for telerobotics", *Integrated Computer-Aided Engineering*, vol. 8, no. 2, pp. 95–104, 2001.

[6] L. A. Nguyen, M. Bualat, L. J. Edwards, L. Flckiger, C. Neveu, K. Schwehr, M. D. Wagner, and E Zbinden, "Virtual reality interfaces for visualization and control of remote vehicles", *Autonomous Robots*, vol. 11, no. 1, pp. 59–68, 2001.

[7] A. Kheddar, R. Chellali, and P. Coiffet, "Virtual environment-assisted teleoperation" in *Handbook of Virtual Environments: Design, Implementation and Applications*, 2001, ch. 48, pp. 959-998.

[8] A. A. Mohamed and C Chevallereau, "Resolution of Robot Redundancy in the Cartesian Space by Criteria Optimization", in *Proceedings of the IEEE International Conference on Robotics and Automation*, Atlanta, GA, 1993, pp. 646-651.

[9] L. Beiner, "Minimum-Force Redundancy Control of Hydraulic Cranes", *Mechatronics*, vol. 7, no. 6, pp. 537–547, 1997.

[10] L. Beiner and J. Mattila, "An improved pseudoinverse solution for reduntant hydraulic manipulators", *Robotica*, vol. 17, pp. 173–179, 1999.

[11] P. La Hera, U. Mettin, I. R. Manchester, A. Shiriaev, "Identification and Control of a Hydraulic Forestry Crane", *submitted to the 2008 IFAC World Congress*, available on request.

[12] M. E. Münzer, "Resolved Motion Control of Mobile Hydraulic Cranes", PhD Thesis, Aalborg University, Denmark, 2002.

[13] Video of crane virtual environment, available online at $www.tfe.umu.se/forskning/Control\_Systems/Movies/virtual\_environment\_exp.AVI$

[14] S Lee, J Choi, D Kim, J Na, S Oh, "Signal Separation Coding for Robust Depth Imaging Based on Structured Light", in *Proceedings of the IEEE International Conference on Robotics and Automation*, Barcelona, 2005.

[15] Video of experiment on forwarder, available online at $www.tfe.umu.se/forskning/Control\_Systems/Movies/forwarder\_crane\_exp.AVI$