Implementing Force-Feedback in a Telesurgery Environment, Using Parameter Estimation

Hansen, Thomas; Henningsen, Claus ; Nielsen, Jens; Pedersen, Rasmus; Schwensen, John; Sivabalan, Senthuran; Larsen, Jesper Abildgaard; Leth, John-Josef

Published in:
2012 International Conference on Control, Automation and Information Sciences

DOI (link to publication from Publisher):
10.1109/CCA.2012.6402708

Publication date:
2012

Document Version
Accepted author manuscript, peer reviewed version

Citation for published version (APA):
Hansen, T., Henningsen, C., Nielsen, J., Pedersen, R., Schwensen, J., Sivabalan, S., Larsen, J. A., & Leth, J-J. (2012). Implementing Force-Feedback in a Telesurgery Environment, Using Parameter Estimation. In *2012 International Conference on Control, Automation and Information Sciences* (pp. 859 - 864). IEEE Press. https://doi.org/10.1109/CCA.2012.6402708

Implementing Force-Feedback in a Telesurgery Environment, Using Parameter Estimation

Thomas Hansen[1], Claus T. Henningsen[1], Jens J. M. Nielsen[1], Rasmus Pedersen[1],
John Schwensen[1], Senthuran Sivabalan[1], Jesper A. Larsen[1] and John J. Leth[1]

Abstract— **During minimal invasive telesurgery with surgical robots, surgeons rely on their vision to determine the forces applied to tissue. A force-feedback control system has been developed, in order to reduce the unnecessary forces applied by the surgeon. To avoid adding any additional hardware, the forces in the system have been estimated on the basis of the existing actuators, using parameter estimation techniques. The inevitable time-delays in the network, which imposes challenges in control design, are also estimated and compensated for within the control design.**

During tests, it has been shown that it is possible to implement a distributed networked controller, which is stable over a range of typical time-delays. This shows, that the applied parameter estimation technique is indeed a viable solution for implementing force-feedback in telesurgery.

I. INTRODUCTION

The use of robotics has become an important tool for certain surgical procedures, leaving patients with significantly shorter recovery time due to the possibility of minimal invasive surgery. There are still many possibilities in the use of robotics that are not yet being fully utilised. Having already separated surgeon from patient by use of electrical signals, it seems like a natural step in this time of advanced telecommunications to further geographically distance them. This could result in patients having easier access to expertise within specific surgical fields instead of having to travel to other cities or even countries for the best treatment available. Additionally, even small hospitals would be able to offer surgery for a wider range of illnesses without having to employ several surgeons, each specialising in a certain procedure.

However, by simply remotely controlling a surgical robot, the surgeon is restricted by not having the sense of touch. Adding force-feedback would relieve the surgeon of having to estimate the force visually from the strain caused on the tissue of the patient. To avoid increased production costs, obtaining a measure for the forces applied using only the hardware already in place for robotic surgery would be favourable. This would also ease the implementation of force-feedback on existing systems.

The concept of telesurgery is one that has been researched for many years and has resulted in the world's first transatlantic surgery in 2001 by [1], with the surgeon sitting in an office in New York and performing a cholecystectomy[1] on a women in Strasbourg. This was done over a dedicated fiber-optic connection with a mean round trip time of 155 ms. Further studies into the effects of time-delay in a telesurgical environment has been done such as [2], who concludes that delays of 400 ms or below had no significant impact on neither task time nor error-rate when performing surgical like tasks. [3] have studied the combination of time-delay and force-feedback concluding that in a suturing task the skill of the surgeon outweighed the negative effects of 165 and 270 ms delay. Neither did force-feedback increase completion time for experienced surgeons, however it did significantly reduce forces exerted upon the sutured fabric. Both of these studies confirm [4] who concludes that time-delays of 600 ms significantly increases task times. In addition they also show that asynchrony in video and control feedback improves performance compared to further delaying one of the signals to obtain synchronised feedback.

This article will describe the work done to design and implement a stable force-feedback control system on a telesurgery prototype consisting of one end-effector and a surgeon interface, which in the following will be referred to as the joystick, connected through a network imposing a variable time-delay on the information transmitted. The communications protocol developed to handle the information flow between each station is designed to run on an IP-network and could therefore immediately be used over the Internet. However, the results discussed in this article are obtained on a Local Area Network with simulated delays. To avoid using expensive force and torque sensing equipment the forces experienced by the end-effector are found by parameter estimation, using measurements that would be available even if the system had not been designed for force-feedback.

In Section II, Methods, the prototype and environment will be further described along with any definitions and assumptions made during the design of the system. Furthermore, the ideas and methods behind both control design and the development of the parameter estimator will be described before introducing the practical tests, through which the results have been obtained. Section III, Results, will take a closer look upon the details of the data gathered in the specified tests of the system. Lastly, Section IV, Conclusion, will conclude on the obtained results while Section V, Perspectives, will leave suggestions regarding areas for further research and

---

[1] T. Hansen, C. T. Henningsen, J. J. M. Nielsen, R. Pedersen, J. Schwensen, S. Sivabalan, J. A. Larsen and J. J. Leth are with the Department of Electrical Engineerig, Aalborg Univeristy, 9220 Aalborg, Denmark

[1] Removal of the gallbladder.

development.

II. METHOD

During the design phase of the controllers and the model based parameter estimator, certain assumptions have been made. In order to obtain a linear model of the system, non-linearities of the physical system have not been taken into consideration, besides saturation of the power supply and constraints in the joystick and end-effector position. This will result in a simplified model and may pose a need for more manual tuning.

Dynamics of the end-effector have not been modeled, as they are assumed negligible compared to the dynamics of the motors driving it. The joystick is also assumed to have no dynamics, mainly because of the hand's influence on the system.

Latency in the system is defined as the round-trip time from the console sends a latency service packet to it receives a corresponding latency service packet. This is done by starting a timer on the console when the packet is added to the send queue and stopping the timer when the corresponding packet is received. When defining latency in this manner, the motor drivers processing time is not taken into consideration.

To implement force feedback it is utilized, that the current of a DC-motor is proportional to the torque it applies. By knowing the end-effectors motor current, this can be used as a reference point to the console control loop and thereby force-feedback can be applied without the use of force sensors.

A. Setup

For implementation and testing, a prototype of a remote surgical system has been developed. This has been done with the intention of constructing a modular and easy to expand system, which makes future modification and feature adding possible by module replacement and/or addition. To ensure this property the interfaces between the subsystems have been defined. An example of this modular development is that the interface between each workstation and each motor is processed by a separate motor control board. The designed setup is shown in Figure 1.

The two computers depicted are running a low latency Linux kernel, which does not guarantee hard real time operations, but has a considerably lower latency than the corresponding generic kernel [5]. The motor drivers are embedded systems which handle the interface between workstations and motors driving the end-effector and applying force feedback to the joystick, through a serial link. The end-effector used for the prototype system is from da Vinci Surgical Systems[2] and has four degrees of freedom. The joystick has been developed at Aalborg University, and also has four degrees of freedom.

The network interface between the two workstations is an Ethernet communication channel, utilizing the User Datagram Protocol(UDP) of the Internet Protocol Suite. All tests on the system have been performed on a local network, however the communication protocol has been developed for Internet communication. To simulate an unreliable network, like the Internet, that may impose varying time delay, the network emulation tool Netem has been used [6].

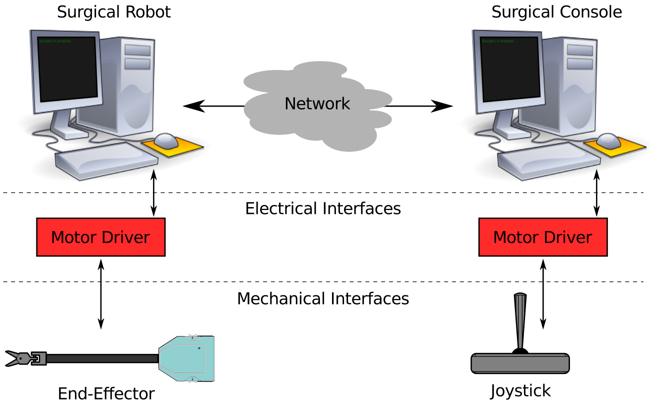

Fig. 1: *Setup of the developed remote surgical system prototype.*

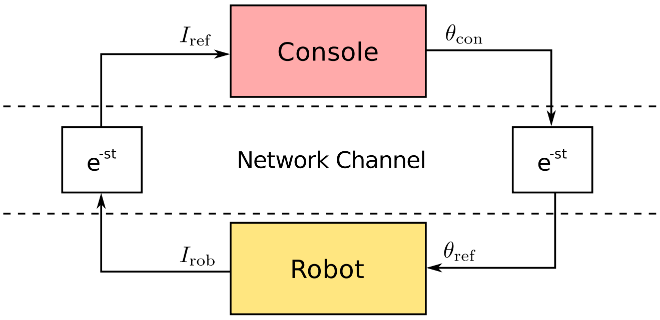

Fig. 2: *Concept overview of the control system.*

B. Control

The designed control scheme consists of two systems, namely console and robot. A concept overview of the control scheme is shown in Figure 2. The console control system's objective is to ensure that force applied by the robot is fed back to the user through the joystick. This feedback has been chosen to be a 1:1, based on the motor current in the robot control system.

The robot control system's objective is to ensure that the position of the robot corresponds to the position of the joystick. The relationship between joystick and robot position in the test setup is 1:1. This relationship in commercial products, like the da Vinci Surgical System, is 5:1 seen from the console side, as this filters shaky hand movements and makes it possible to conduct more precise procedures. What the optimal ratio should be is an entire study in itself, however, the developed controllers have been developed so they can take any ratio the user would desire. To control the internal loops as well as give a reference point to the other control loop, each plant model has one input and two outputs, which is illustrated in Figure 3 on the next page. Feedback

---

[2] `davincisurgery.com`

current from the robot to the console is negated because the force feedback should counteract the robots movement. From Figure 3, the closed loop transfer functions for the two inner loops are derived as stated in Eq. 1 and 2. Where H_c is the console closed loop and H_r is the robot closed loop.

$$H_\mathrm{c} = \frac{C_\mathrm{c} \cdot G_{1\mathrm{c}}}{1 + C_\mathrm{c} \cdot G_{2\mathrm{c}}} \tag{1}$$

$$H_\mathrm{r} = \frac{-C_\mathrm{r} \cdot G_{2\mathrm{r}}}{1 + C_\mathrm{r} \cdot G_{1\mathrm{r}}} \tag{2}$$

The open loop system equation for the outer feedback loop can be seen in Eq. 3. The open loop system is used for stability analysis.

$$H_\mathrm{ol} = H_\mathrm{c} \cdot H_\mathrm{r} \cdot \mathrm{e}^{-2\mathrm{st}} \tag{3}$$

1) Stability Analysis: To be able to guarantee a completely stable system it is not enough to simply look at the transfer function of the main loop. Considerations must also be made to all the other transfer functions arising from the multiple inputs and outputs of each subsystem. Using the principle of internal stability, the console and robot models are individually analysed for stability across all of their transfer function. This equates to checking the eigenvalues, i.e. poles, of the corresponding state matrix representation of the system.

Once the subsystems are proven stable, the overall system can be proven stable simply by looking at a single transfer function through the system [7]. This function, which in this system has been defined as the open loop of $\frac{I_\mathrm{rob}}{I_\mathrm{ref}}$, is the one that will be used to determine the effects of time-delay.

With the individually designed controllers for the subsystems the resulting overall system was calculated to be stable, however with a phase margin of only 5°. This was deemed as insufficient due to the fact that model inaccuracies could effectively mean that the actual system was pushed into an unstable region. Therefore, a phase margin of 45° is chosen as a design goal. It is not adjusted according to its transient properties due to the assumption that the hand has an overshadowing dampening effect. To increase the phase margin a lead compensator has been designed in series with the overall loop.

To counteract the destabilizing effects of a pure time-delay a Smith predictor is implemented. It contains a model of the robot side of the system from the reference position to the output current, and, with an ideal model, will eliminate the effects of delay completely from a stability point of view. How the lead compensator and the Smith predictor are incorporated into the control scheme is depicted in Figure 4. In order to implement the Smith predictor the latency, measured in round trip time, must be known. To ensure that this information is provided to the control system, the developed communications protocol measures it.

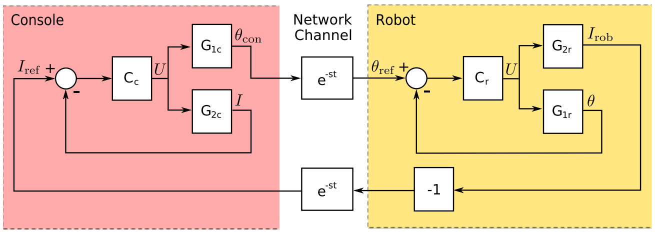

Fig. 3: *Expanded block diagram of the control system, with the different transfer functions. C_c is the console controller and C_r is the robot controller. $G_{1c,r} = \frac{\theta}{U}$, $G_{2c,r} = \frac{I}{U}$.*

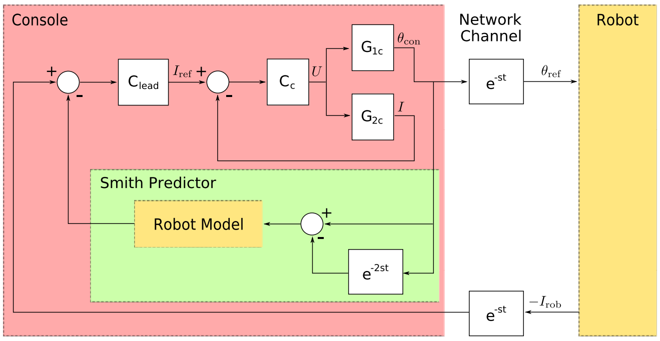

Fig. 4: *Illustration of how the lead compensator and Smith predictor is incorporated into the control scheme. The lead compensator makes sure that the open loop system has a phase margin of 45° or above and the Smith predictor handles the varying delays imposed by the network channel.*

C. *Parameter Estimation*

Due to the fact that the torque in a DC-motor is proportional to the current, it can directly be calculated when knowing the current. Ideally, the current should be found without adding any additional sensing equipment. Instead it would be beneficial if an already installed sensor could be used, such as the position sensor, which is a crucial sensor in robotic surgery. When the position and the sampling frequency is known, it is trivial to calculate the velocity.

A derivation of a state space DC-motor model can be found such that the states are the current, i, and the angular velocity, ω. This model needs an input, which is the motor voltage, u. The model in Eq. 4 and 5 is in continuous time.

$$\dot{\mathbf{x}} = \mathbf{A} \cdot \mathbf{x} + \mathbf{b} \cdot u, \qquad \mathbf{x} = \begin{bmatrix} i \\ \omega \end{bmatrix} \tag{4}$$

$$y = \mathbf{c} \cdot \mathbf{x}, \qquad y = \omega \tag{5}$$

The model makes it possible to estimate the current based on the input and the velocity. The estimation can be enhanced by applying the model into a Kalman filter. A Kalman filter is a linear optimal observer that estimates a state, given a model, an input and a measurement. Based on this information it estimates the state better than if only the model is used, as old input keeps influencing the estimate. Therefore, a Kalman filter is designed as done in [8]. In order to implement the model in the filter, a discrete model is needed, see Eq. 6 and 7. When the model is discretised, it is important to set a high enough sampling frequency, such that the dynamics of the

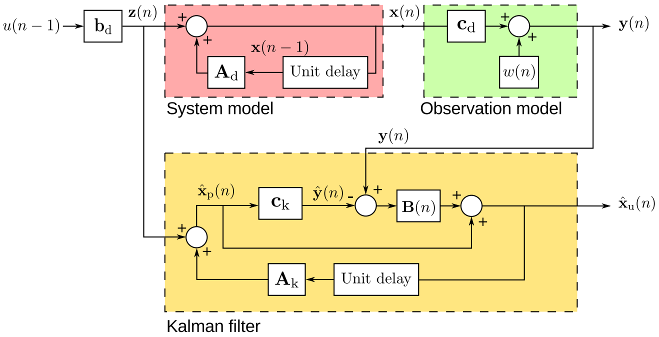

Fig. 5: *Illustration of how the system and observation model are connected to the Kalman filter.*

motor are not suppressed.

$$\mathbf{x}(n) = \mathbf{A}_\mathrm{d} \cdot \mathbf{x}(n-1) + \mathbf{b}_\mathrm{d} \cdot u(n-1) \tag{6}$$

$$y(n) = \mathbf{c}_\mathrm{d} \cdot \mathbf{x}(n) \tag{7}$$

The Kalman filter model is seen in Eq. 8, 9 and 10 and depicted in Figure 5, which shows how the Kalman filter uses the model to predict the states $\mathbf{x}(n)$.

$$\hat{\mathbf{x}}_\mathrm{p}(n) = \mathbf{A}_\mathrm{k} \cdot \hat{\mathbf{x}}_\mathrm{u}(n-1) + \mathbf{z}(n), \qquad \hat{\mathbf{x}} = \begin{bmatrix} \hat{i} \\ \hat{\omega} \end{bmatrix} \tag{8}$$

$$\hat{y}(n) = \mathbf{c}_\mathrm{k} \cdot \hat{\mathbf{x}}_\mathrm{p}(n), \qquad \hat{y} = \hat{\omega} \tag{9}$$

$$\hat{\mathbf{x}}_\mathrm{u}(n) = \hat{\mathbf{x}}_\mathrm{p}(n) + \mathbf{B}(n) \cdot (y(n) - \hat{y}(n)) \tag{10}$$

As seen, there is a direct connection between the discrete motor model and the system, observation and Kalman filter model. Where $\mathbf{A}_\mathrm{k} = \mathbf{A}_\mathrm{d}$, $\mathbf{c}_\mathrm{k} = \mathbf{c}_\mathrm{d}$. The real states $\mathbf{x}$ and y are estimated with $\hat{\mathbf{x}}_u$ and $\hat{y}$ by the Kalman filter. The noise, w, is the noise added the observations and the driver input $\mathbf{z} = \mathbf{b}_\mathrm{d} \cdot \mathrm{u}(n-1)$. The noise, w, and the driver, $\mathbf{z}$, are modelled as having a normal distribution with mean zero and variance σ_z^2 and σ_w^2. These variances are used to update the Kalman gain, $\mathbf{B}$, which minimises the error in the filter.

The Kalman filter takes two steps in order to predict and update its estimate of the states. These are depicted as $\hat{\mathbf{x}}_\mathrm{p}$ and $\hat{\mathbf{x}}_\mathrm{u}$, respectfully. The Kalman filter needs to be initialised. This is done by guessing on initial conditions and selecting variances. The current and angular velocity of the motors are expected to be zero at start up, therefore $\hat{\mathbf{x}}_u(0) = [0\ 0]^T$. The variance of w is set low and $\mathbf{z}$ high, this means that the measurement is highly trusted, and that noise is only a little suppressed. With this filter design it is possible to use the velocity and voltage to estimate the current and thereby also the torque.

D. *Test Description*

The results described in this article have been obtained through the following three tests of the system. In each case the tests have been performed using a local network at Aalborg University with the use of the network simulator Netem, for generating delays that are both known and reproducible[3]. When performing the tests all available joints have been used. Only data from one joint will be presented in this paper.

1) 1st Test - Position Tracking: Data from the tachometers of both console and robot are logged. The test is performed consecutively with delays of 0, 100, 200 and 400 ms. For each setting the joystick is rotated 45° to the right of centre position and once there the grip is loosened. After approximately two seconds the joystick is rotated 45° to the left of centre position before loosing the grip. An additional two seconds later the joystick is rotated back to the centre position and released. This will show how well the joystick's position is tracked by the robot and if stability is an issue within any of the tested time-delays.

2) 2nd Test - Torque Tracking: In this test the estimated current at the robot and the console is logged. The test is again performed consecutively with delays of 0, 100, 200 and 400 ms. During testing a force is applied to the joystick while keeping the end-effector fixated. This should result in a constant and equal current in both joystick and end-effector and will show how well torque applied to one of them is tracked by the other.

3) 3rd Test - Parameter Estimation: In this test the estimated motor current on the robot is logged, along with a measurement of the actual motor current. During the test a force is applied to the joystick, first in one direction and then in the other. Hereafter, the end-effector is fixated and the motion is repeated. From this test it should be evident how well the actual motor current is estimated. By first applying the force in one direction and then the other, it is also tested how well the direction of the current is estimated.

III. RESULTS

The results are presented as graphs of the transient behaviour to inputs described in the previous section. Only the tests where the round trip time is 0 and 400 ms are shown. This is done as 400 ms is the maximum allowable delay in the system, and for systems with less delay, the transient behaviour is better in both theory and tests.

A. *1st Test - Position Tracking*

First, a test with 0 ms round-trip time is seen in Figure 6. It shows how the robot tracks the position of the joystick. The steady state error seen on the position tracking is most likely due to an insufficient integral term, resulting in the more aggressive controller on the joystick compensating for the error once it is released. By fine tuning the robot's controller, the tracking could be improved without having the joystick move. Increasing the round-trip time to 400 ms results in the response seen in Figure 7. While the system is still stable, a damped oscillation is now present. The steady state behaviour appears to suffer from the same problem as without delay, that is the joystick is compensating for the slower integral term on the robot.

---

[3] The round-trip time between the two computers, is found to have mean 0.4 ms and standard deviation 0.1 ms. This delay is treated as zero during the tests.

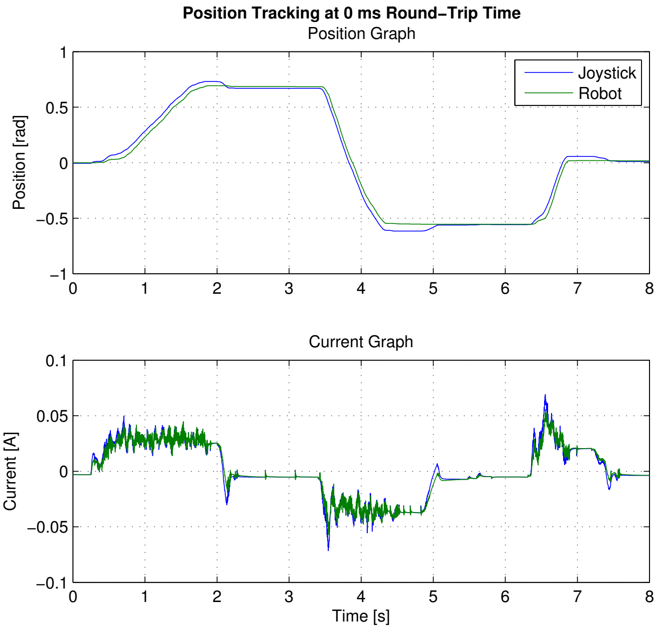

Fig. 6: *Test of system at 0 ms round-trip time and robot moving freely. The first plot shows how the position of the joystick and robot behaves, while the second plot shows the estimated current on the robot and the estimated current on the joystick.*

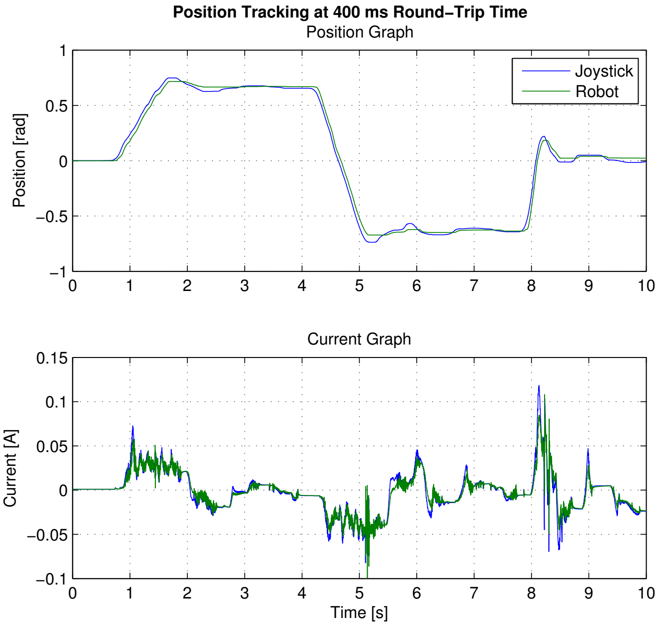

Fig. 7: *Test of system at 400 ms round-trip time and robot moving freely. The time-axis for the robot is shifted by 200 ms in order to improve comparability.*

B. 2^{nd} *Test - Torque Tracking*

When the movement of the robot is restrained and the test is performed without delay the systems' responses can be seen in Figure 8. There is a clear overshoot of approximately 40% with no significant steady state error. Additionally, it appears that the motor input reaches saturation as the position is tracked poorly compared to the previous test while the current for the most part is bounded at ±115 mA. Adding a round-trip time of 400 ms will result in Figure 9 on the next page. The most notable effect of the delay is again a damped oscillation, now for the current, whenever the joystick is moved. Though it is still stable.

C. 3^{rd} *Test - Parameter Estimation*

The parameter estimation test compares the estimated with the measured current on the robot side of the setup. The comparison is depicted in Figure 10. It shows that the current measured by use of the motor driver is only positive and cannot be below 50 mA. This is caused by the motor driver, and the way its current sensor works.

As seen, the current estimation tracks the measured current well. In the scenario where the robot is fixated (after 6 seconds) the estimation does not reach the measured current. This is assessed to be caused by simplifications in the model. Otherwise it is seen how the estimated current describes a lot of the dynamics of the measured.

IV. CONCLUSION

In this paper an implementation of force-feedback in a surgical robotic system by estimation of the applied force is introduced. The prototype setup has been developed in a modular manner consisting of standard COTS hardware. The software development platform is based on the Ubuntu low latency kernel, which does not guarantee real time operations, but reduces latency considerably compared to the generic kernel. To simulate different network delays, the network emulation tool Netem for Linux has been used.

A control design method that renders it possible to design the individual loops separately, by guaranteeing internal stability in them, has been utilized. Knowing that both control loops are stable, only a single path through the outer loop has been examined. In order to reach a satisfactory phase margin of 45° a lead compensator has been designed in series with the overall open loop system. To counteract the destabilizing effect of the network time delays a Smith predictor, containing a model of the robot system, is incorporated into the console control algorithm.

In order to provide force-feedback to the surgeon, without adding any additional sensing equipment, a Kalman filter estimating the motor current, has been introduced. The implemented filter is based on a system model as well as knowledge of the voltage input and position measurements.

The results obtained in this paper consolidate that it is possible to design a control scheme that is stable even with considerable delays imposed by the network channel. The results also show that it is indeed possible to implement force-

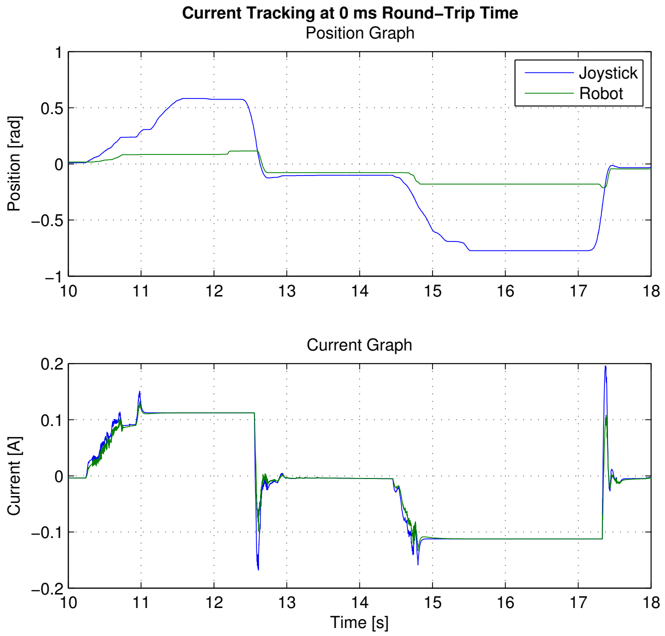

Fig. 8: *Test of system at 0 ms round-trip time and robot restrained.*

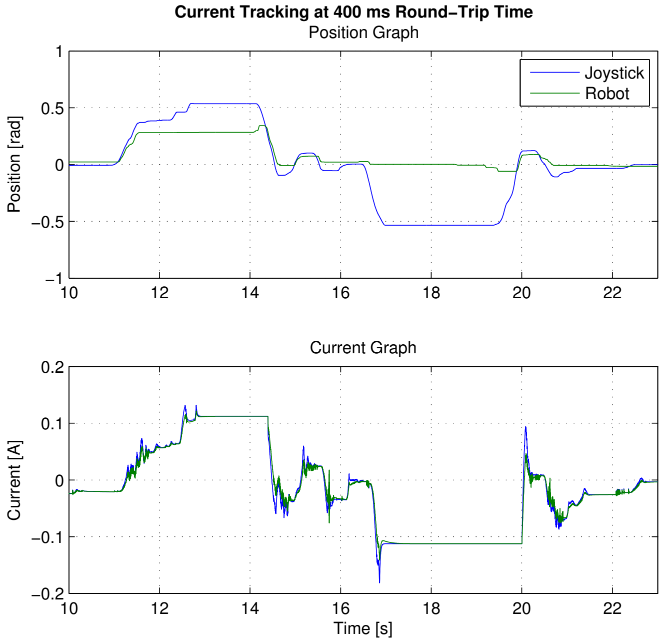

Fig. 9: *Test of system at 400 ms round-trip time and robot restrained. The time-axis for the robot is shifted by 200 ms in order to improve comparability.*

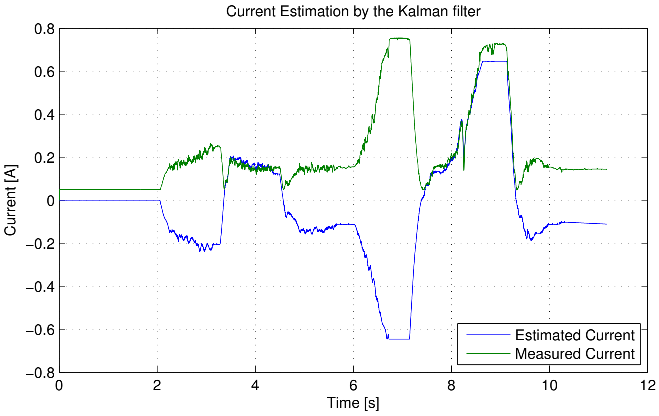

Fig. 10: *Test of the parameter estimator. The joystick is moved back and forth, and from time 6 s, the robot is fixated. The current sensor can only measure in absolute values and cannot show below 50 mA.*

feedback without any direct force or torque measurements, by the use of parameter estimation.

V. PERSPECTIVES

The application of 1:1 force-feedback to telesurgery is not necessarily an advantage in all situations. Using robotics, it is possible for the surgeon to perform under more comfortable conditions, such as sitting in a chair and avoid wearing caps, face mask and gloves. However, having to also apply the same force, instead of merely moving a joystick, could result in unnecessary fatigue. Depending on the procedure performed it could be beneficial to be able to reduce the amount of force fed back, at the surgeon's discretion, when it is irrelevant for the quality of the procedure. Taking this a step further, it could prove useful to make on-the-fly changes to feedback ratio in both directions. Meaning that the surgeon would also be able to enhance the sense of touch for when performing the most delicate procedures, incorporating the studies of [9], such that telesurgery in the future could utilize and enhance the surgeon's sense of touch instead of completely eliminating it as it currently does.

REFERENCES

[1] J. Marescaux, J. Leroy, F. Rubino, M. Smith, M. Vix, and D. Mutter, "Transcontinental robot-assisted remote telesurgery: Feasibility and potential applications," *Anals of Surgery*, vol. 235, no. 4, pp. 487–492, April 2002.

[2] R. Rayman, K. Croome, N. Galbraith, R. McClure, R. Morady, S. Peterson, S. Smith, V. Subotic, A. van Wynsberghe, and S. Primak, "Long-distance robotic telesurgery: A feasibility study for care in remote environments," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 2, no. 3, pp. 216–224, Sep 2006.

[3] J. Arata, H. Takahashi, S. Yasunaka, K. Onda, K. Tanaka, N. Sugita, K. Tanoue, K. Konishi, S. Ieiri, Y. Fujino, Y. Ueda, H. Fujimoto, M. Mitsuishi, and M. Hashizume, "Impact of network time-delay and force feedback on tele-surgery," *International Journal of Computer Assisted Radiology and Surgery*, vol. 3, no. 3, pp. 371–378, June 2008.

[4] J. M. Thompson, M. P. Ottensmeyer, and T. B. Sheridan, "Human factors in telesurgery: Effects of time delay and asynchrony in video and control feedback with local manipulative assistance," *Telemedicine Journal*, vol. 5, no. 2, pp. 129–137, 1999.

[5] A. I. Bogani, "Linux low latency kernel," launchpad.net/~abogani/+archive/lowlatency.

[6] T. L. Foundation, “Network emulation tool: Netem,” linuxfoundation.org/collaborate/workgroups/networking/netem.

[7] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control*. John Wiley & Sons, 1996.

[8] S. Padmakumar, A. Vivek, and R. Kallol, “A tutorial on dynamic simulation of dc motor and implementation of kalman filter on a floating point dsp,” *World Academy of Science, Engineering and Technology*, vol. 53, no. 126, pp. 781–786, 2009.

[9] G. D. Gersem, H. V. Brussel, and F. Tendick, “Reliable and enhanced stiffness perception in soft-tissue telemanipulation,” *The International Journal of Robotics Research*, vol. 24, no. 10, pp. 805–822, Oct. 2005.