Preprints of the 18th IFAC World Congress
Milano (Italy) August 28 - September 2, 2011

Model-Aided Controller Evaluation for a Surgical Assistance Robot to Track Patient Movements in Real Time

H.-C. Schneider * **J. Wahrburg** ** **H. Roth** ***

* *Center for Sensorsystems, University of Siegen, Germany (e-mail: schneider@zess.uni-siegen.de).*
** *Center for Sensorsystems, University of Siegen, Germany (e-mail: wahrburg@zess.uni-siegen.de).*
*** *Head of Control Engineering Department, Institute of Automation Technology, University of Siegen, Germany (e-mail: hubert.roth@uni-siegen.de).*

Abstract: This contribution describes the model-aided development and evaluation of a surgical robot motion controller that has been performed in order to improve the robot's dynamic behavior during optically navigated real time surgical instrument tracking. The navigation of the robot is realized using a camera-based optical 3D localizing system for *visual servoing* feedback. In this paper, we present a model-based controller that improves the dynamics of surgical instrument tracking, under the consumption of the limited measurement rate of common optical localizer systems and the use of a velocity-driven robot arm.

Keywords: Robot arms, Robot control, Robot dynamics, Robot navigation, Robot vision, Servomotor actuators, State-space methods, Model-based control, Feedforward control, Medical applications

1. INTRODUCTION

Recently, navigation systems dominate computer assisted surgery. These systems track the position of surgical instruments by means of a 3D localizing system and insert corresponding symbols into any preoperatively acquired image on the computer display. Originating from applications in neuro-, spinal- and Ear-Nose-Throat (ENT)-surgery, such systems have found wide acceptance in most bone-related surgical applications. However, surgical instruments are still guided manually when using common navigation systems. Thus hand tremor, slipping or the inhomogeneous bony structure of the patient can cause unintentional deviations in the result of a navigated surgical intervention. Where surgical navigation has its constraints, the extension to a navigated mechatronic assistance can help to further improve the overall precision, accuracy and especially the repeatability of surgical interventions. For instance, such mechatronic assistance can be assembled by means of a common six degrees of freedom robot arm which is integrated into the navigation system architecture. The interaction between the surgeon and the robotic system is a very important issue. Fully autonomous systems have lost acceptance in the surgical community because the surgeon must be in charge of the intervention instead of acting only as an observer. An alternative solution is to provide a semi-autonomous and cooperative system where benefits of both, human and machine, can be derived. One example for such system is modi*CAS* (Wahrburg et al. (2004)), where the robot arm just helps the surgeon to precisely and accurately place any surgical instrument relatively to the patient's anatomy and with respect to an image-based preoperative planning (X-Ray, computer tomography, 2D/3D C-Arm image). The placement of the surgical instrument is performed using navigation but the main interaction with the patient is still carried out by the surgeon along configurable degrees of freedom. In all situations, the navigated control of the system can be dominated by the surgeon using a haptic interface by means of a subsumption control architecture. Fig. 1 shows the environmental setup of the modi*CAS* system using a MHI PA10-6C (Mitsubishi Heavy Industries Portable Arm, 10kg wrist load capacity, 6 joints) as manipulator and a NDI-Polaris P4 (Northern Digital Inc., Canada) as optical localizer, where $\{arb\}$ describes the optically measured spatial pose of interest of the patient's anatomy, $\{ttp\}$ represents the corresponding pose of the surgical instrument that is measured by means of an optical reference body $\{rrb\}$, $\{ots\}$ is the base coordinate frame of the optical localizer, $\{tcp\}$ matches the wrist and $\{rbs\}$ the base of the robot arm.

One key feature of the modi*CAS* System is to keep constant any desired placement of the instrument $\{ttp\}$ relatively to the patient $\{arb\}$, even if the patient moves. Thus the necessity of rigid patient fixation will be omitted. A comparable system, for example, is *Cyberknife* (Muacevic et al. (2006)) in the field of robot assisted radiosurgery. In this paper we will investigate a novel control strategy in order to improve the overall dynamic performance of the modi*CAS* surgical instrument tracking.

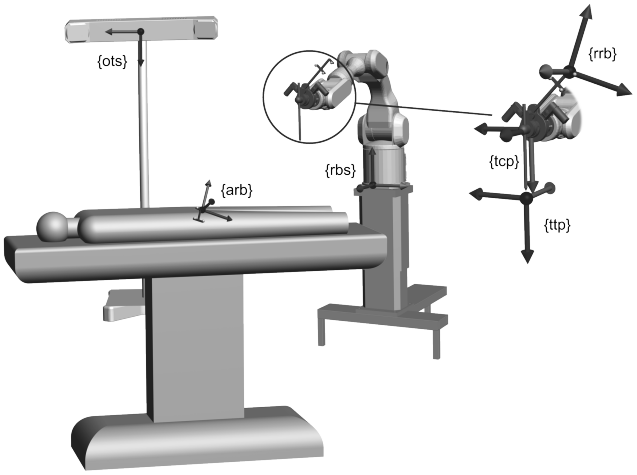

Fig. 1. Coordinate frames - {ots} localizer (optical tracking system), {arb} patient (aim reference body), {rrb} (robot reference body) {rbs} robot base, {tcp} robot wrist (tool center point), {ttp} tool tip

1	*control input*	**position based**	image based
2	*functional principle*	**look and move** (static/**dynamic**)	direct visual servoing
3	*sensor position*	inside-out	**outside-in**
4	*spatial degrees of freedom*	2D (monovision)	**3D (stereovision)**
5	*feedback*	endpoint open loop	**endpoint closed loop**

Table 1. Categorizing the visual servoing approach

2. REAL TIME INSTRUMENT TRACKING BY MEANS OF VISUAL SERVOING

Due to the fact that we use a camera-based optical 3D localizer system in order to feed back the robot's displacement during the instrument tracking procedure, we can characterize our system to be a so called *visual servoing system*. Visual servoing systems in general have been already described by Weiss et al. (1987) and dynamic effects were studied by Corke and Good (1996). Several approaches have just been categorized in detail, depending on the type of e.g. camera or control principle. For instance, a generalized overview is given in Kragic and Christensen (2002). Table 1 comprises the major features that define the type of a visual servoing system.

First, we distinguish between: *position based* systems, where the control law is based on completely reconstructed 3D position (and orientation) data; and *image based* systems, where the control output is directly processed from image data, e.g. from corner or edge features, respectively. Due to the fact that in general we use a common surgical 3D localizing system that typically delivers full-featured spatial information, we just focus on position based systems. *Second*, it plays an essential role, if: the robot motion control is directly and exclusively based on visual information from the cameras (*direct visual servoing*); or if the dynamic control of the robot is primarily based on its joint encoder measurements. Then, the visual feedback only performs capturing the object to be tracked, and calculating the corresponding position and orientation setpoint for the robot motion controller (*look and move*). We talk about *static* look and move, if the robot has reached steady state before the next setpoint is calculated, and *dynamic* look and move, if the setpoint update for the robot motion controller is sent continuously and faster, than the robot motion controller can compensate. Due to the relatively low measurement rates of common optical localizers (in our case 25Hz) it is reasonable to use the *dynamic look and move* approach, where the dynamic motion control of the robot is primarily based on joint encoder feedback. *Third*, a visual servoing system, where the utilized optical localizer is directly mounted to the robot wrist, labels *inside-out* (or eye in hand) system. The opposite is *outside-in*, where the localizer looks from an observer perspective. *Fourth*, we distinguish between: the use of *monovision* cameras (allows *2D-visual servoing*, e.g. for pick and place applications using SCARA robots); and the utilization of *stereovision* cameras that enable full *3D-visual servoing*, like the NDI-Polaris in our case. As the Polaris has a relatively wide baseline (distance between the two *'eyes'* of the 3D stereovision system), we assemble it as outside-in. *Fifth*, using such an outside-in look and move system lets us decide whether to do it endpoint open or endpoint closed. *Endpoint open loop* systems optically just capture the object to be tracked. The setpoint calculation for the robot motion controller then is based on a fix-calibrated spatial setup. *Endpoint closed loop* systems additionally use an optical reference capture of the robot's wrist pose for the setpoint calculation. We perform endpoint closed because that offers two important opportunities: First the visual servoing system can react on spatial setup modifications during tracking (like manually moving the robot basement or camera) and second it can compensate deviations between the nominal model and the real kinematics of any uncalibrated robot (that have been described by Bruyninckx and Shutter (2001)) by means of an accurate optical localizer.

In sum, the previous decisions lead to a visual servoing approach that may be categorized as *position based dynamic look and move* using *outside-in stereovision* in *endpoint closed loop* configuration. With respect to the spatial setup in Fig. 1 such system can be described using the block diagram in Fig. 2, where T_b^a is a 4×4 homogeneous matrix describing the spatial forward transformation from coordinate frame a to coordinate frame b, $\mathbf{q}_c\,[6 \times 1]$ is a vector consisting of the joint angle commands for the robot joint drives, $\mathbf{q}'_c\,[6 \times 1]$ holds the joint velocity commands, and $\mathbf{q}$,$\mathbf{q}'$ the corresponding feedbacks. Using this visual servoing principle, object tracking and kinematic or calibration error compensation is performed by the visual feedback loop, while the dynamic motion control of the robot is still carried out by the decentralized robot joint controllers. Further details on the functional principle of the presented structure can be reviewed in Schneider and Wahrburg (2010). Using such structure, the tracking dynamics highly depend on the design of the joint controllers. The following section describes the model-based design and tuning of the joint angle controllers, with respect to the technical constraints within the whole visual servoing system. The control synthesis is performed on the basis of a reliable simulation model that has been comprehensively described in Schneider and Wahrburg (2010) as well.

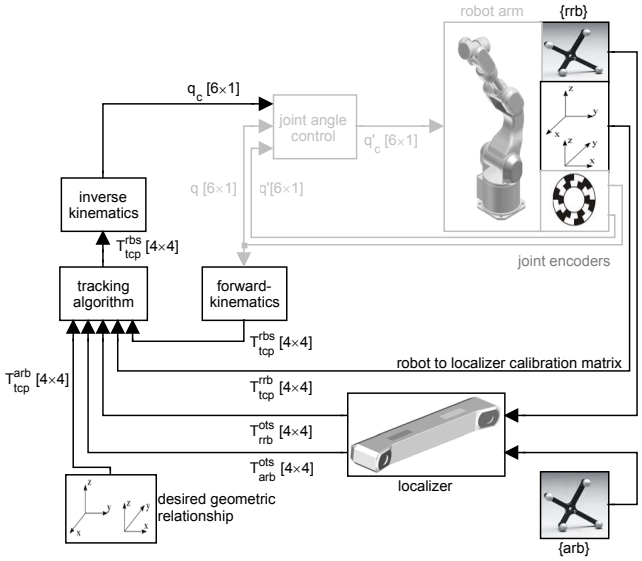

Fig. 2. Surgical instrument tracking by means of visual servoing

3. MODEL-BASED CONTROL DESIGN AND TUNING

In the field of industrial robotics, it is common to combine a P/PI cascade control structure, that itself consists of an underlying PI joint velocity controller and an overlying proportional joint angle controller, together with a velocity feedforward compensation in order to reduce the contouring error of robot arms. If the desired spatial trajectory is known in advance, one can simply derive the joint velocity trajectories off-line from the joint angle command trajectories, under the consumption that their resolution is sufficient or one can previously perform a proper interpolation between the settle points, respectively.

3.1 Proportional Feedforward Control

Performing visual servoing, we cannot calculate the velocity signals off-line but must them directly derive from the optical localizer measurements in real time. Thus, the feedforward gain has to be conservatively adapted to the measurement capabilities of the localizer. If one uses a velocity-driven robot with internal PI joint velocity controllers, a P/PI cascade with feedforward structure can be completed using the control scheme in Fig. 3, where q_c is the angular setpoint trajectory for one joint, q is the measured joint angular response trajectory, $\dot{q}_c$ is the velocity command for the underlying joint velocity controller, Δt_{rob}, Δt_{ots} are the sample times of robot and localizer and K_p, K_{ff} are the proportional controller and velocity feedforward gains. Assuming to have a virtually ideal localizer with sample and dead times close to zero, we could get really good results by just using a feedforward gain $K_{ff} = 1$ for every joint. Then we would only need a low gain feedback proportional controller to compensate remaining control errors. Fig. 4 illustrates a simulation using such a feedforward compensation, where sample time Δt_{ots} and dead time t_{ots} of the localizer both are $2ms$. The upper diagram shows a rotation of the tracked object around a spatially defined axis $\Delta\alpha$ and the corresponding angular response of the robot. The

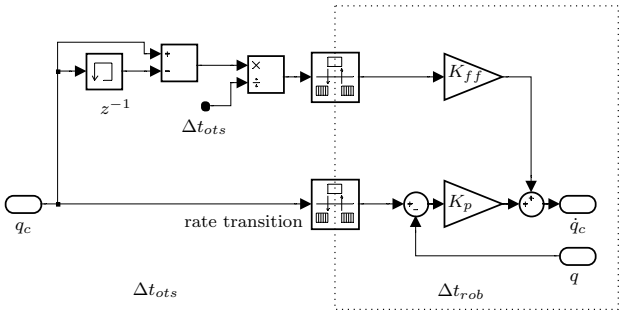

Fig. 3. Proportional joint angle controller with angular velocity feed forward compensation for one single robot joint drive

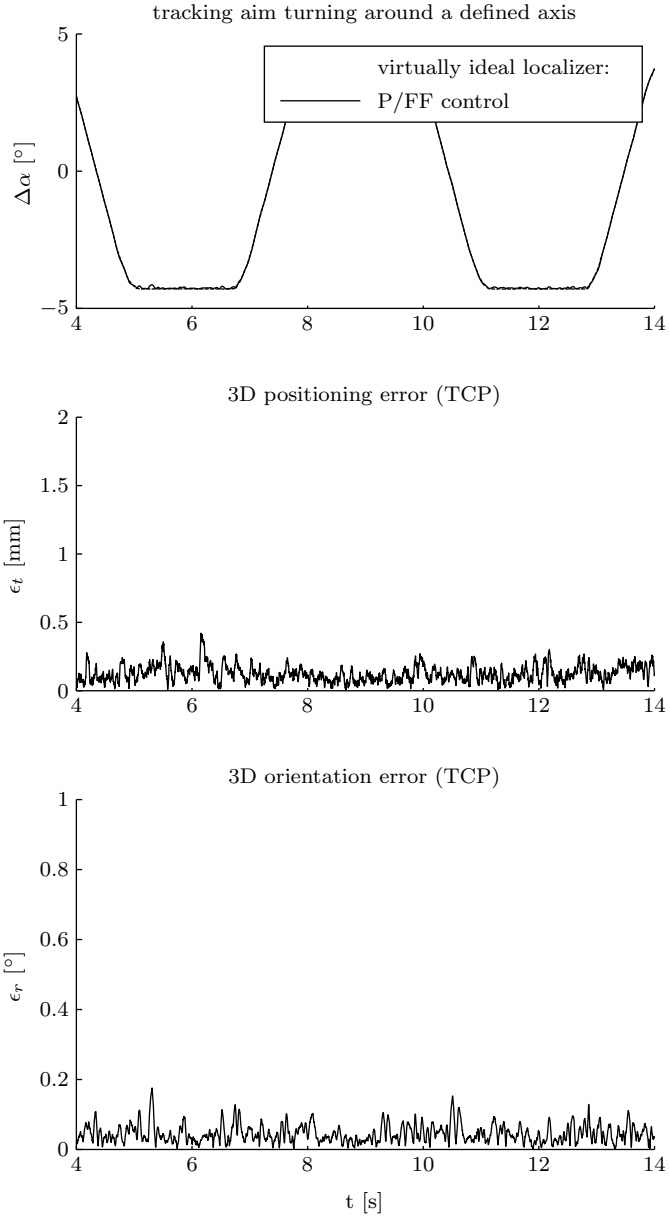

Fig. 4. Simulation of a virtually ideal localizer with $t_{ots} = \Delta t_{ots} = 2ms$ in combination with a proportional feedforward controller (P/FF, $K_{ff} = 1$) for all joint drives

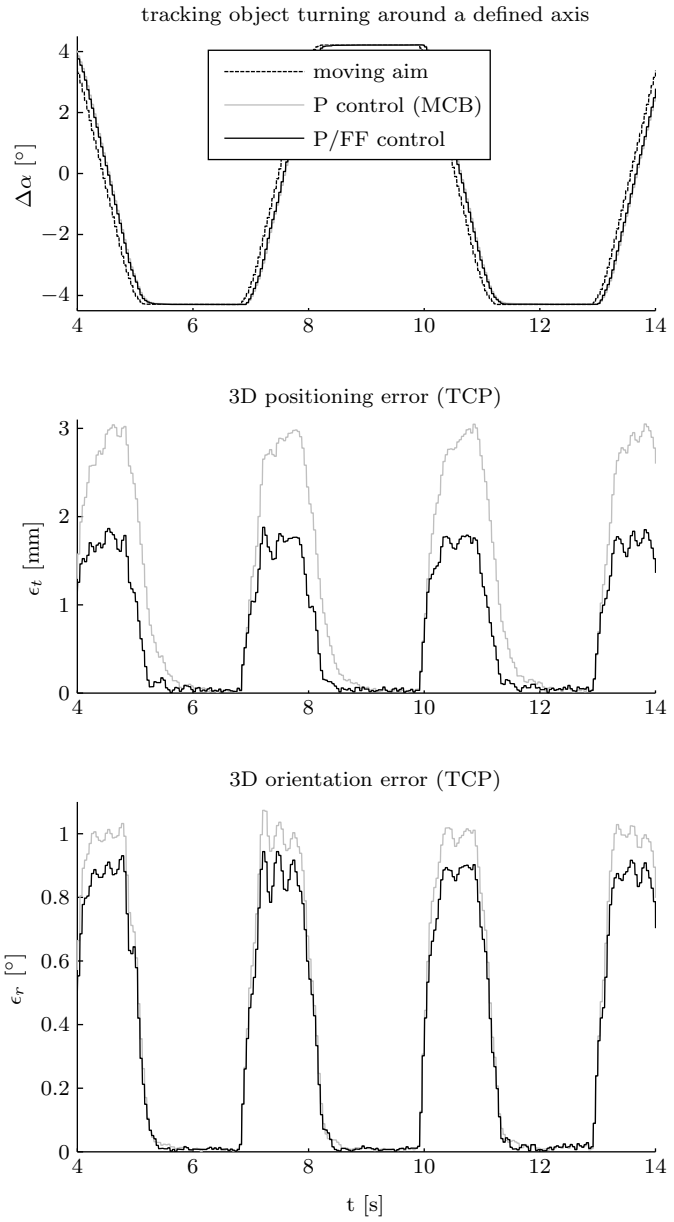

Fig. 5. Comparison between the optimized proportional controller with feedforward compensation (P/FF control) and the controller from the robot manufacturer's Motion Control Board (P control)

angular trajectories have been derived through axis-angle separation of the 3D orientation data. The middle diagram shows the corresponding spatial position error ϵ_t and the lower diagram shows the corresponding spatial orientation error ϵ_r. We see that the dynamic tracking error does not exceed the stochastic stationary error which is caused by localizer measurement noise. In practice, we need to tune the feedforward loop more conservatively because our real localizer has a longer dead time (33ms) and a lower sample rate ($25Hz$ instead of 500Hz). Under the consumption of zero localizer noise, Fig. 5 shows proportional feedforward control with parameters optimized for minimum quadratic step response error and zero overshoot. Certainly, under such real timing conditions the tracking error increases with respect to the simulation in Fig. 4. However, it is significantly lower than using the proportional controller from the robot manufacturer's motion control board. Table 2

joint	S_1	S_2	E_1	E_2	W_1	W_2
phase margin [°]	76,6	79,3	79,4	84,2	87,0	86,1

Table 2. Phase margins

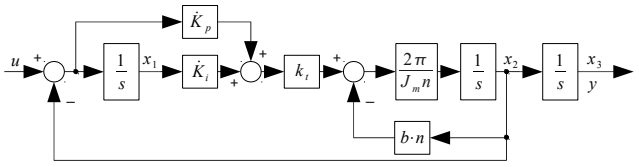

Fig. 6. Simplified model of one single robot joint drive

gives the phase margin of the controller with plant for each robot joint (S1,2=Shoulder; E1,2=Elbow; W1,2=Wrist).

In the long term, the key to a lower tracking error will be to provide a smoother and less delayed tracking object velocity trajectory, in order to max out the capability of feedforward compensation. On-line smoothing strategies like Kalman-Filtering (Schneider and Wahrburg (2008)) did not produce tracking results that are fast as well as securely precise enough for our surgical application, yet (e.g. due to overshoot in the prediction). However, in the following section we will present a control strategy that further reduces tracking error, if no better motion trajectory, than that which we can actually derive from the Polaris measurements, is available.

3.2 Pole Placement Controller

The dynamics of velocity-driven robot joint actuators with high gear ratios and internal PI velocity controllers can be robustly simulated using the model structure in Fig. 6, where $\dot{K}_p$ and $\dot{K}_i$ are the gains of the internal PI velocity controller, k_t is the current to torque constant, J_m is the cumulated inertia of the joint drive, n is the gear ratio and b is viscous friction coefficient. Due to the high gear ratio $n = 50$, any disturbance on the gear side of a robot joint drive is significantly downscaled to the motor side and quickly compensated by the internal PI velocity controller. These facts enable us to well predict the joint's angular position and velocity trajectory using such simple model and to neglect pay load and arm configuration of the robot. All model parameters have been derived jointwise by means of an identification procedure that has been described in Schneider and Wahrburg (2010). One limitation of this strategy is that dynamic effects within the mechanical robot arm structure (e.g. resonances) cannot be directly considered during the control design process. Concerning further details on modeling electrical robot joint drives, please refer to Sciavicco and Siciliano (2000). Our model includes the state variables and the output variable

$$\dot{x}_1 = -x_2 + u \tag{1}$$

$$\dot{x}_2 = \frac{2\pi}{J_m \cdot n}\left[\dot{K}_i k_t x_1 - \left(\dot{K}_p k_t + b \cdot n\right) x_2 + \dot{K}_p k_t u\right] \tag{2}$$

$$\dot{x}_3 = x_2 \tag{3}$$

$$y = x_3. \tag{4}$$

Defining the state vector

$$\mathbf{x} = [\, x_1 \;\; x_2 \;\; x_3 \,]^T, \tag{5}$$

we get the time continuous state space model

$$\dot{\mathbf{x}} = \underline{F}\ \mathbf{x} + \mathbf{g}\ u \tag{6}$$

$$y = \mathbf{h}^T \mathbf{x} \tag{7}$$

with the system matrix

$$\underline{F} = \begin{bmatrix} 0 & -1 & 0 \\ \dfrac{2\pi \dot{K}_i k_t}{J_m \cdot n} & -\dfrac{2\pi\left(\dot{K}_p k_t + b \cdot n\right)}{J_m \cdot n} & 0 \\ 0 & 1 & 0 \end{bmatrix}, \tag{8}$$

the input vector

$$\mathbf{g} = \begin{bmatrix} 1 & \dfrac{2\pi \dot{K}_p k_t}{J_m \cdot n} & 0 \end{bmatrix}^T, \tag{9}$$

and the output vector

$$\mathbf{h}^T = [\,0\ 0\ 1\,]. \tag{10}$$

Discretization of the state space model The state space controller to be implemented will run on a computer platform. Thus the continuous model described in 1 – 10 must be converted to the discrete model

$$\mathbf{x}(k+1) = \underline{\Phi}\ \mathbf{x}(k) + \gamma\ u(k) \tag{11}$$

$$y(k) = \mathbf{h}^T \mathbf{x}(k). \tag{12}$$

The system matrix of the discrete model $\underline{\Phi}$ can be derived by

$$\underline{\Phi} = e^{\underline{F}\Delta t_{rob}}, \tag{13}$$

where $\Delta t_{rob} = 2ms$ is the sample time of the robot motion controller interface ($f_{rob} = 500Hz$). The input vector γ of the time discrete model can be derived by

$$\gamma = \int_0^{\Delta t_{rob}} e^{\underline{F}\eta} d\eta \times \mathbf{g} \tag{14}$$

as well. A detailed explanation of the mathematical time discretization of state space models can be found in Franklin et al. (1997).

Modeling of joint encoder lags by extension to a delayed-state model In laboratory experiments, we have determined that the robot joint encoder output delay can be approximated with $t_{rob} = 6ms$. Due to the sample time of $\Delta t_{rob} = 2ms$, the output delay is modelable by extending the discrete model to a delayed-state model with six state variables, where three state transitions are zero order hold elements

$$\mathbf{x}_{xt}(k+1) = \underline{\Phi}_{xt}\ \mathbf{x}_{xt}(k) + \gamma_{xt}\ u(k) \tag{15}$$

$$y(k) = \mathbf{h}_{xt}^T \mathbf{x}_{xt}(k). \tag{16}$$

The state vector is extended to

$$\mathbf{x}_{xt} = \left[\, \mathbf{x}^T\ x_4\ x_5\ x_6 \,\right]^T \tag{17}$$

with

$$x_4(k+1) = x_3(k) \tag{18}$$

$$x_5(k+1) = x_4(k) \tag{19}$$

$$x_6(k+1) = x_5(k) \tag{20}$$

$$y(k) = x_6(k), \tag{21}$$

as well as the system matrix is extended to

$$\underline{\Phi}_{xt} = \begin{bmatrix} \underline{\Phi} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \\ \mathbf{h}^T & 0 & 0 & 0 \\ \mathbf{0}_{1\times3} & 1 & 0 & 0 \\ \mathbf{0}_{1\times3} & 0 & 1 & 0 \end{bmatrix} \tag{22}$$

joint	S_1	S_2	E_1	E_2	W_1	W_2
λ_1	$0,984$	$0,984$	$0,984$	$0,933$	$0,96$	$0,93$
λ_2	$0,947$ $+0,045i$	$0,953$ $+0,042i$	$0,938$ $+0,04i$	$0,96$	$0,96$	$0,96$
λ_3	$0,947$ $-0,045i$	$0,953$ $-0,042i$	$0,938$ $-0,041i$	$0,85$	$0,85$	$0,86$
λ_4	0	0	0	0	0	0
λ_5	0	0	0	0	0	0
λ_6	0	0	0	0	0	0

Table 3. Desired closed loop poles

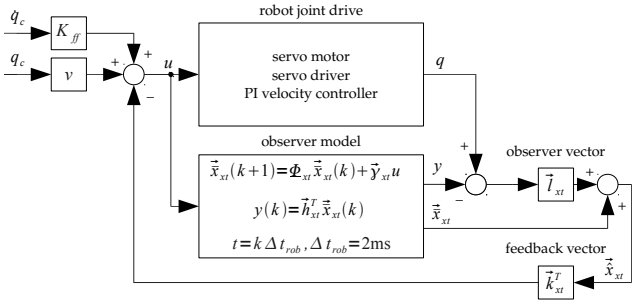

Fig. 7. Block diagram: pole placement controller with state observer and feedforward compensation

and the input and the output vectors are extended with three zero elements

$$\gamma_{xt} = \left[\, \gamma^T\ 0\ 0\ 0 \,\right]^T \tag{23}$$

$$\mathbf{h}_{xt}^T = \left[\, 0\ 0\ 0\ \mathbf{h}^T \,\right]. \tag{24}$$

3.3 Control Law and Pole Placement

We define the control law

$$u = v \cdot q_c - \mathbf{k}_{xt}^T \mathbf{x}_{xt}, \tag{25}$$

where v is the prefilter, q_c is the desired joint angle setpoint and $\mathbf{k}_{xt}^T$ is the feedback vector that is determined from the desired system poles using *Ackermann's* formula.

Under the same optimization criterion like for the presented proportional feedforward controller, we identified the optimal pole locations shown in table 3. All elements $k_4 \ldots k_6$ of the feedback vector $\mathbf{k}_{xt}^T$, that is derived from the desired pole locations in table 3, are zero, so that in steady state we get

$$u(k+1) = v \cdot q_c(k+1) - \begin{bmatrix} k_1 \\ k_2 \\ k_3 \\ 0 \\ 0 \\ 0 \end{bmatrix}^T \cdot \begin{bmatrix} 0 \\ 0 \\ q_c(k+1) \\ q(k) \\ q(k-1) \\ q(k-2) \end{bmatrix} = 0, \tag{26}$$

such that the prefilter can be determined to be

$$v = k_3. \tag{27}$$

The pole placement controller is completed with a state observer (as the joint's angular position q is the only used measurement variable) and a feedforward compensation like shown in Fig. 7. Table 4 holds the derived feedforward gains K_{ff}. Using the resulting controller within our visual servoing framework, we can achieve the results illustrated in Fig. 8 in comparison to the previous proportional feedforward controller. If we further compare the step responses of the joint controllers (proportional feedforward

Gelenk	S_1	S_2	E_1	E_2	W_1	W_2
K_{ff}	0,66	0,68	0,75	1,4	1,4	1,15

Table 4. Pole placement controller - feed forward gains

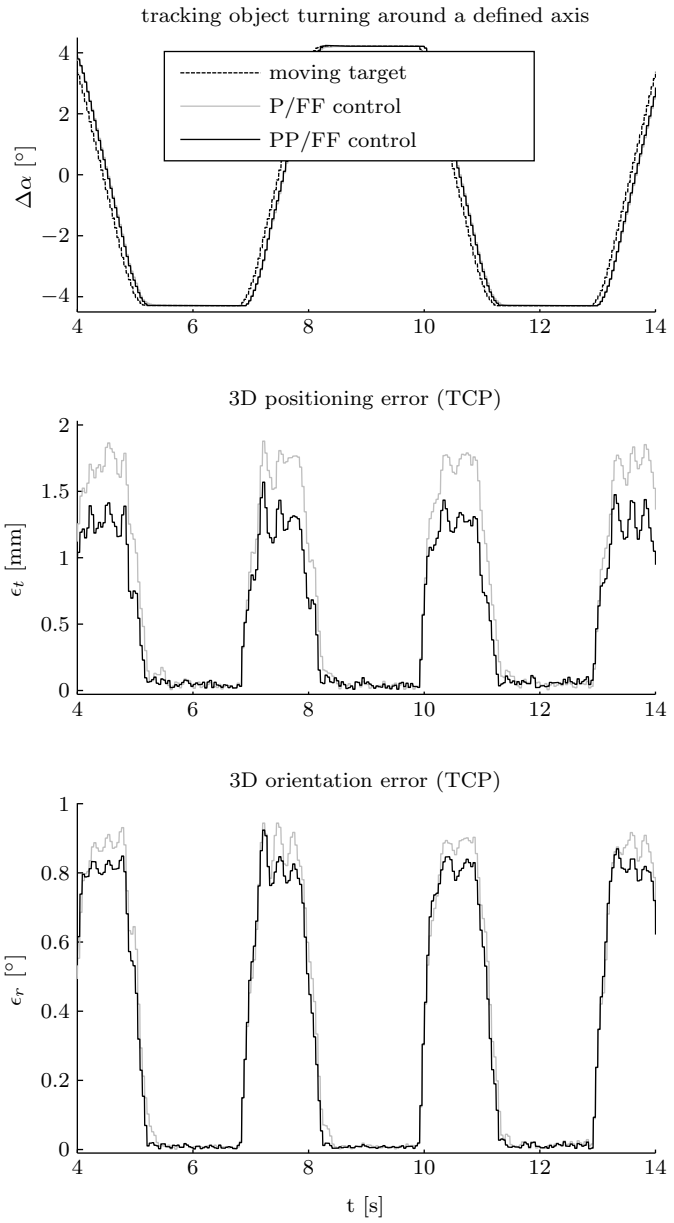

Fig. 8. Comparison: proportional feedforward control (P/FF control) and pole placement with feedforward compensation (PP/FF control)

and pole placement with feedforward) in Fig. 9, then we can figure out that the pole placement controller shows less oscillation and a faster compensation than the proportional one for every joint.

4. SUMMARY AND OUTLOOK

In this paper we presented a pole placement control structure that can improve a dynamic look and move visual servoing system based on a velocity-driven robot arm, if the sample time of the utilized localizer is constrained. In the next future we will perform further simulations as well as laboratory experiments on the robustness of the

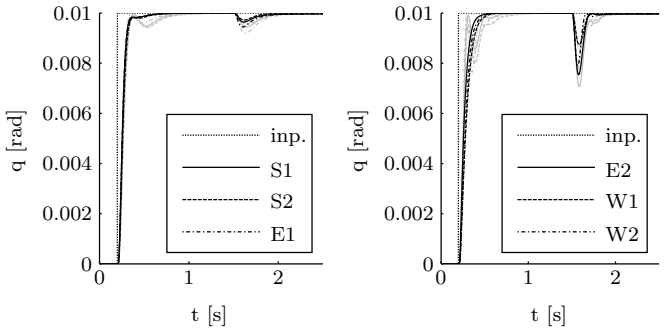

Fig. 9. Step responses of all six joint drives: proportional feedforward control (P/FF control, gray) and pole placement with feedforward compensation (PP/FF control, black)

controller and focus on how a variation of the localizer's measurement behavior (lag, noise, miscalibration) can influence the overall dynamics of the instrument tracking procedure using the presented control structure.

REFERENCES

Bruyninckx, H. and Shutter, J. (2001). Introduction to intelligent robotics. Technical report, Katholieke Universiteit de Leuven.

Corke, P. and Good, M. (1996). Dynamic effects in visual closed-loop systems. *IEEE Transactions on Robotics and Automation*, 12(5), 671–683.

Franklin, G., Powell, D., and Workman, M. (1997). *Digital control of dynamic systems*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 3 edition.

Kragic, D. and Christensen, H. (2002). Survey on visual servoing for manipulation. Technical report, School of Computer Science and Communication, Centre for Autonomous Systems, Numerical Analysis and Computer Science, Stockholm, Sweden.

Muacevic, A., Staehler, M., Drexler, C., Wowra, B., Reiser, M., and Tonn, J. (2006). Technical description, phantom accuracy, and clinical feasibility for fiducial-free frameless real-time image-guided spinal radiosurgery. *Journal of Neurosurgery: Spine*, 5(4), 303–312.

Schneider, H. and Wahrburg, J. (2008). A Novel Real Time Core for Dynamics and Safety Enhancement of a Navigated Surgical Assistance Robot. *at – Automation – Methods an Applications of Control and Information Technology*, 56(9), 483–493.

Schneider, H. and Wahrburg, J. (2010). *Simulation Model for the Dynamics Analysis of a Surgical Assistance Robot*. Intech, Vukovar, Croatia.

Sciavicco, L. and Siciliano, B. (2000). *Modelling and Control of Robot Manipulators*. Springer, London, Great-Britain.

Wahrburg, J., Gross, I., Knappe, P., Pieck, S., Knzler, S., and Kerschbaumer, F. (2004). An interactive mechatronic assistance system to support surgical interventions. In *International Congress Series*, volume 1268, 431–436. Elsevier.

Weiss, L., Sanderson, A., and Neuman, C. (1987). Dynamic sensor-based control of robots with visual feedback. *IEEE Journal of Robotics and Automation*, 3(5), 404–417.