Robust control of mechanical systems: a computational design study

Proceedings of the 30th Conference
on Decision and Control
Brighton, England • December 1991

Robust Control of Mechanical Systems: A Computational Design Study

Bram de Jager
Department of Mechanical Engineering, WH 2.137
Eindhoven University of Technology
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
Email: jag@wfw.wtb.tue.nl

ABSTRACT

Several methods for the design of robust controllers for non-linear systems, especially manipulators, are investigated, with emphasis on robustness for unmodeled dynamics.

This investigation is performed along the following lines. We select a system for testing the designs methods, make a simple model for this system, design controllers for this model with different design methods and design parameters, use these controllers to control parametrized versions of a more complicated model and evaluate with respect to the tracking performance, the robustness characteristics of the resulting controlled model, the computational complexity of the controllers and the ease of design.

The system to be controlled is a two degrees of freedom manipulator, with a rotational and a prismatic joint. The control task is to follow a circular object with trajectory control along the circumference and force control in the direction perpendicular to the circumference. For the trajectory a fixed computed torque with PD controller is used. The force is controlled by the controllers under investigation. These controllers are all based on a linearizing state-feedback with acceleration feedforward (computed torque), and a controller based on a H_2 or H_∞ design for the resulting linear system. A PD controller is used for reference. The design parameters or functions are used to get specific bandwidths for the controlled linearized design model, so this bandwidth is a universal design parameter. The model used for evaluation is the model used for the design plus first order models for the drives. The time constants of the drives are parameters, so we test robustness for parametrized unmodeled dynamics.

The simulation results indicate that there are no advantages in using H_∞ or H_2 controllers, instead of a PD controller. This is attributed to the decoupled nature of the controllers, resulting in a single input/single output design.

1 INTRODUCTION

To fulfill the increasing requirements on the dynamic behavior of systems, it is of advantage to control the system. In the design process of controllers often a model of the system is used. Models can be derived according to first principles or be based on measurements. A mix of both approaches is also usual. To describe the system adequately, a nonlinear model can be necessary. More often than not this model is only a rough approximation of reality. This should be taken into account during the control design process, and can be done by using a design method in which the model uncertainty can be specified, so that no iteration between design and evaluation is necessary. These methods are now available for linear systems, the H_∞ and μ-synthesis designs [1, 2], to name a few. At the moment they can only be used for linear models and the way the uncertainties must be specified is seldom directly related to real world uncertainties. Whether these methods can have a significant contribution to the design of robust controllers for non-linear systems is a subject of further investigation.

Approaches for fast and accurate control of non-linear systems are adaptive controllers [3] and robust controllers that are directly suitable for non-linear systems. A disadvantage of the first approach is the inherent inability to adapt to unmodeled dynamics, resulting in a robustness that is comparable to classical PD control [3], but the tracking error can be made much smaller.

The approach in this paper is to use first a linearizing state-feedback, which also decouples the model and then add a robust controller for the resulting linear time-invariant decoupled model. Then, we take advantage of the abundance of results for robust control of linear systems. Expected problems are:

- does there exist a linearizing state-feedback,
- when the model used in the linearizing state-feedback is not exact, will the resulting model still be linear, time-invariant and decoupled,
- what is the resulting uncertainty of the non-linear model after applying the state-feedback,
- how to express the uncertainty in the framework of the linear system design methods,
- can H_∞ methods be directly applied to non-linear models.

The first problem has been solved about 10 years ago [4]. The answer to the second problem is in general negative. The other problems have not been solved to the knowledge of the author, although some research has been done [5].

In this paper the problems mentioned are circumvented by a computational investigation of the quality of the controllers, by using them to control a model that is more complex than the design model. The conclusions are then based on an evaluation of the resulting simulation data. This, in fact, implies again a design where iteration between design and evaluation is necessary, which ideally should be avoided to speed up the design process.

This paper has the following structure. First, the models used to design and evaluate the controllers are given, followed by a discussion of the control task. Second, the general structure of the hybrid control scheme is given, in which the scrutinized controllers will be plugged in. Then the design of the controllers will be elaborated. This is followed by a discussion of the method to evaluate the designs. After that the simulation results are presented and discussed. The paper closes with conclusions and suggestions for further research.

2 COMPUTATIONAL EXPERIMENT SETUP

To assess the robustness of the controllers, different models are used for the design and the evaluation. So, a nominal *design model* and an *evaluation model* are introduced. The controllers are designed for the design model and evaluated for the evaluation model. The evaluation model is based on the design model with parametrized unmodeled dynamics added. The control task introduced is a hybrid position/force task along the circumference of a circular object.

2.1 Design model

Mechanical systems like robotic manipulators can often be described by the following system of n second order differential equations, in n degrees-of-freedom or joint coordinates q_i

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = f + J^T(q)F_e \qquad (1)$$

where $H(q)$ is the $n \times n$ positive definite inertia matrix, $C(q,\dot{q})\dot{q}$ the n vector of Coriolis and centrifugal forces, $g(q)$ the n vector of gravitational forces, f the n vector of generalized forces (forces or torques), $J(q)$ the $n \times n$ Jacobian of the manipulator and F_e the n vector of external forces on the manipulator. In this model each joint has its own motor. We neglect the dynamics of the motors, Coulomb and viscous friction, stiction, backlash and flexibility of the joints and links.

The design model chosen is a model for a two degrees-of-freedom manipulator, moving in the horizontal plane, with a rotational and a prismatic joint, a so called RT-robot. This choice is based on the simplicity of the system and the availability of an experimental RT-robot for possible future experiments. The model equations for the RT-robot of Fig. 1 are

$$\begin{aligned} &\theta_1\ddot{r} - (\theta_1 r - \theta_2)\dot{\varphi}^2 = \\ &\qquad F + F_{x_1}\cos\varphi + F_{x_2}\sin\varphi \\ &(\theta_1 r^2 - 2\theta_2 r + \theta_3)\ddot{\varphi} + 2(\theta_1 r - \theta_2)\dot{r}\dot{\varphi} = \\ &\qquad M - F_{x_1} r\sin\varphi + F_{x_2} r\cos\varphi \end{aligned} \qquad (2)$$

where r and φ are the prismatic and rotational degree-of-freedom, F_{x_1} and F_{x_2} the components of the external force F_e in x_1 and x_2 direction, M and F are the motor torque and force acting on the manipulator and θ_1, θ_2 and θ_3 are related to the physical parameters by

$$\theta_1 \quad = \quad m + m_l$$

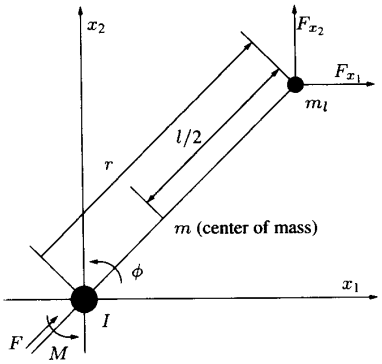

Figure 1: Schematic drawing of RT robot

$$\theta_2 = \frac{1}{2}ml \qquad (3)$$
$$\theta_3 = I + \frac{1}{3}ml^2.$$

For the parameter values used in the computations see Table 1.

parameter	value	unit
m	10	kg
m_l	5	kg
I	5	kg m^2
l	1	m
θ_1	15	kg
θ_2	10	kg m
θ_3	$5 + \frac{10}{3}$	kg m^2

Table 1: Parameters of the design model

2.2 Evaluation model

The model to evaluate the design is equal to the design model plus first order models for the dynamics of the motors. So instead of $f = \tau$ with f proportional to the controller output vector τ, we obtain

$$\begin{bmatrix} \tau_f & 0 \\ 0 & \tau_m \end{bmatrix} \dot{f} + f = \tau$$

with τ_f, τ_m the motor time constants. The motor time constants are chosen equal, between $\frac{1}{200}$ [s] and $\frac{1}{800}$ [s], with a nominal value of $\frac{1}{400}$ [s].

2.3 Control task

The control task is to follow a circular object with position control along the circumference and force control in radial direction. The stiffness K_e of the object is assumed to be known precisely and is equal to 10^6 [N/m]. We select this task because it is a challenging application for adaptive control schemes, when the object stiffness must be estimated. In this research the stiffness is assumed to be known and constant, justified by the aim to assess the robustness for unmodeled dynamics, to enable future comparison.

The desired trajectory in cartesian end-effector space is

$$x_d(t) = \begin{bmatrix} a + R_d \cos\psi_d \\ b + R_d \sin\psi_d \end{bmatrix}$$

with $R_d = r_e = 0.25$ [m] the radius of the object, $\psi_d = 2\pi t - \frac{\pi}{2}$ [rad] the desired angular position and $a = 0$ [m], $b = 0.5$ [m] the center of the object, see Fig. 2. The desired force F_n is equal to 100 [N]. The periodic nature of the task makes it easy to compute accurate error statistics, without influence of initial transients.

3 HYBRID CONTROL SCHEME

The scheme proposed in [6] is used for the hybrid control task. For the structure of the controller see the block diagram in Fig. 3. The meaning of the symbols in Fig. 3 is specified later.

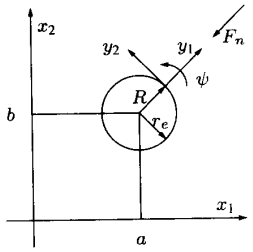

Figure 2: Desired trajectory

Three main blocks can be distinguished. First, the computed torque part, using the inverse dynamics of the design model. Second, a trajectory control loop, driven by the projection of the position error on the tangent to the object at the actual position. Third, a force control loop, driven by the radial projection of the force error.

The inverse dynamics of the design model is [6]

$$\tau = HJ^{-1}(u - \dot{J}\dot{q}) + C\dot{q} + g - J^T F_e \qquad (4)$$

with u the sum of the outputs u^x and u^f of the trajectory and force controller.

The trajectory controller is a standard PD-controller, with a feed forward loop for the desired acceleration $\ddot{x}_d$, so, with the inverse dynamics block it is in essence a computed torque controller with PD component. The controller has constant parameters and is given by

$$u^x = \ddot{x}_d + K_d^x \dot{\tilde{x}} + K_p^x \tilde{x} \qquad (5)$$

with $\tilde{x} = x_d - x$. The controller parameters are equal to

$$K_p^x = \begin{bmatrix} k_p^x & 0 \\ 0 & k_p^x \end{bmatrix} = \begin{bmatrix} 140^2 & 0 \\ 0 & 140^2 \end{bmatrix}$$

$$K_d^x = \begin{bmatrix} k_d^x & 0 \\ 0 & k_d^x \end{bmatrix} = \begin{bmatrix} 2 \cdot 140 & 0 \\ 0 & 2 \cdot 140 \end{bmatrix},$$

to get approximately critically damped dynamics with an undamped radial frequency of 140 [rad/s], well below the inverse of the smallest motor time constant.

The force controller is the subject of this investigation. Here, H_2, H_∞ and a reference PD-controller are used, with a feed forward path for $\ddot{F}_d$, so

$$u^f = -K_e^{-1}(\ddot{F}_d + K_d^f \dot{\tilde{F}} + K_p^f \tilde{F}) \qquad (6)$$

for the PD controller, with $\tilde{F} = F_d - F_e$. The term $\ddot{F}_d$, in end-effector coordinates, is equal to $\ddot{x}_d F_n / R_d$, when the desired force F_n in task space is constant. The term $K_d^f \dot{\tilde{F}} + K_p^f \tilde{F}$ is replaced by the output of the H_2 or H_∞ controller, both driven by $\tilde{F}$ only.

The matrix R^{-1} appearing in the control scheme is equal to

$$R^{-1} = \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix}$$

and is used to transform from cartesian end-effector space (x coordinates) to cartesian task space (y coordinates). The projection of the position and force errors is performed by the matrix

$$\Sigma = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

By this projection we obtain the force and position errors needed for the controllers.

Because the linearized model is decoupled, both the trajectory and force controllers are themselves decoupled. The controllers for the two diagonal blocks are identical. Refer to [6] for further details of the control scheme.

4 CONTROLLER DESIGN

Because the stiffness of the environment K_e is constant, the force controller can be designed as a position controller, the controllers designed should only be scaled with $-K_e^{-1}$, as in (6). To avoid a more complicated notation, in the following, therefore, a pure trajectory controller design is discussed and the superscript ^{x} is dropped.

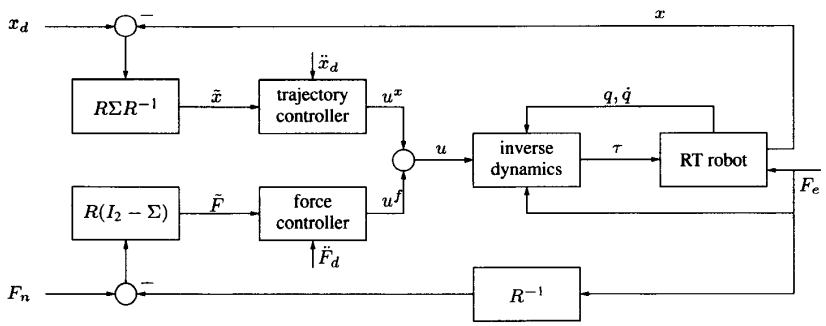

Figure 3: Hybrid control scheme

We design the controllers for the design model, augmented with the inverse dynamics and acceleration feedforward part of the control scheme. Because the controllers are designed in end-effector space, the design model equations in end-effector space are needed, they are

$$H^*\ddot{x} + J^{-T}(C\dot{q} + g) - H^*\dot{J}\dot{q} = J^{-T}f + F_e \tag{7}$$

where $H^*(q) = J^{-T}(q)H(q)J^{-1}(q)$. When the inverse dynamics (4) block is applied to the design model of the manipulator (7), for which $f = \tau$, there results

$$\ddot{x} = u, \tag{8}$$

a decoupled system of two second order differential equations. After transformation to state space we obtain

$$\dot{v}_i = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} v_i + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_i \tag{9}$$

for each degree of freedom x_i, with $v_i = [x_i,\ \dot{x}_i]^T$. This is a model for two integrators in series. After applying PD-feedback $u_{pd} = K_d\dot{\tilde{x}} + K_p\tilde{x}$ the controlled system has a transfer function matrix, where the diagonal entries are second order models with additional zeros

$$\frac{k_d s + k_p}{s^2 + k_d s + k_p}, \tag{10}$$

where the desired states x_d are inputs. The structure of the controlled system (8) in block diagram form is given in Fig. 4.

When the influence of the zero $s_z = -k_p/k_d$ in (10) is neglected, the undamped radial frequency ω_0 is $\sqrt{k_p}$ [rad/s] and the damping factor β is $\frac{k_d}{2\omega_0}$. The design parameters for the PD controller are selected to get a prescribed ω_0 and a damping factor $\beta = 1$, i.e., a critically damped second order system when s_z is neglected, so $k_p = \omega_0^2$ and $k_d = 2\omega_0$. For this choice of parameters, the actual bandwidth ω_b of the controlled model (10), depending on the definition of bandwidth, will be $\sqrt{3+\sqrt{10}}\omega_0 \approx 2.4824 \cdot \omega_0$ for $\alpha_b = \frac{1}{\sqrt{2}}$ and $\sqrt{2+\frac{3}{2}\sqrt{2}}\omega_0 \approx 2.0301 \cdot \omega_0$ for $\alpha_b = \frac{1}{\sqrt{2}} \max_\omega \alpha = \sqrt{\frac{2}{3}}$, where α_b is the magnitude α of the transfer function (10) at the bandwidth frequency ω_b. In the following discussion, we use the second definition of the bandwidth ω_b.

To bring the reference PD-controller in the same state space controller frame as the H_2 and H_∞ controllers, the D part of the PD-controller is approximated by a tame differentiator by using a series connection with a first order system with time constant $\frac{1}{\delta}$, so only the position and not the velocity error is used. The state space form of the PD controller becomes

$$\dot{z} = -\delta I_2 z + \tilde{x}$$

$$u_{pd} = \left[-\delta^2 K_d + \delta K_p\right] z + \delta K_d \tilde{x}.$$

To make the influence of the first order system small, choose the factor δ much larger, 10^5, than the largest design frequency ω_0.

The design of the H_2 and H_∞ controllers follows the lines given in [1]. Design the H_∞ controller $F(s)$ in such a way that the bandwidth ω_b of the resulting controlled system can be specified with the weight function $W_3(s)$ for the complementary sensitivity function $T(s)$. Select the weight function $W_1(s)$ for the sensitivity function $S(s)$ so that the sensitivity function is as small as possible, within the requirement that the H_∞ norm of the transfer function

$$\begin{bmatrix} W_1(s)S(s) \\ W_2(s)F(s)S(s) \\ W_3(s)T(s) \end{bmatrix} \tag{11}$$

is smaller than or equal to 1. We achieve this by choosing

$$W_1^{-1}(s) = \frac{1}{\rho}\frac{(\frac{\omega_1}{\omega_0})^{m_1}(\frac{s}{\omega_1}+1)^{m_1}}{(\frac{s}{2\omega_0}+1)^{m_1}} I_2, \quad m_1 = 1,\ldots,3$$

$$W_2 = 0$$

$$W_3^{-1}(s) = \frac{\alpha_3}{(\frac{s}{\omega_0}+1)^{m_3}} I_2, \quad m_3 = 1,\ldots,2$$

with $\omega_1 = \omega_0 / \sqrt[m_1]{10^{3+m_1}}$, and performing a search for maximal ρ (often called γ iteration), within the constraint that the H_∞ norm of (11) is ≤ 1. Use the factor α_3 in W_3^{-1} to force the amplitude α of W_3^{-1} through the point (ω_b, α_b). The amplitude of (10) coincides at this point. For $m_3 = 1$ the factor $\alpha_3 \approx 2$. Use the factors m_1 and m_3 to tune the slopes of S and T. The best results were obtained with $m_1 = 2, m_3 = 1$. This choice of the weight functions W_1^{-1} and W_3^{-1} gives S a slope of $+2$ for $\omega < \omega_0$, like a PD controller and T a slope of -1 for $\omega > \omega_0$, also like a PD-controller. Although a slope of -2 for T should make the system more robust for high frequency unmodeled dynamics, the results did then not match those of the selected weights.

The algorithm used for this calculation is the one given in [7]. A few modifications of the above problem are necessary to enable a solution for the H_∞ calculations. Shift the poles of the decoupled linearized model (9) to the right slightly. Also assign a small value to the weight W_2. Give both factors a value so small to enable the calculation, but without perceptible influence on the resulting controllers.

The same weight functions are used for the design of the H_2 controller. This design is performed according to the formulas given in [8], in essence formulas for the calculation of an H_∞ controller, but with their parameter $\gamma = 10^3$, because when $\gamma \to \infty$ the H_∞ controller approaches an H_2 controller for the same problem with the same weight functions. This strategy is chosen to prevent numerical problems, which occur in an algorithm for a direct H_2 design based on a combination of an optimal controller and an optimal Kalman filter (LQG) design. The same modifications as for the H_∞ design are used.

The dimensions of the controllers are 2, 6 and 8 for the PD, H_∞ and H_2 design.

5 CONTROLLER EVALUATION

The controllers are evaluated by simulating the evaluation model, with the hybrid controller for 2 [s]. The simulation data for the first second is disrecarded, because of transients originating from the incorrect settings of the initial conditions of the dynamic controller. From the simulation data of the second part of the simulation run the root mean square (RMS) values of the position and force error are used for evaluation.

To assess the robustness of the controller two system parameters are varied. First, the controllers are designed for different design frequencies ω_0. Second, the time constants of the unmodeled dynamics, included in the evaluation model are varied.

The transfer functions of the force controllers designed for $\omega_0 = 400$ [rad/s] are in Fig. 5.

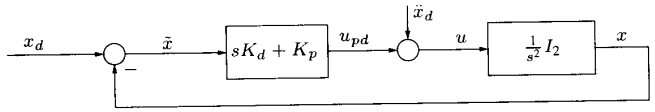

Figure 4: Block diagram of trajectory control loop with PD controller

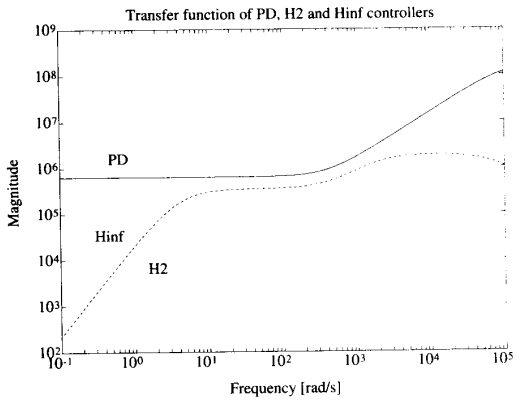

Figure 5: Transfer function amplitude of H_∞, H_2 and PD controllers

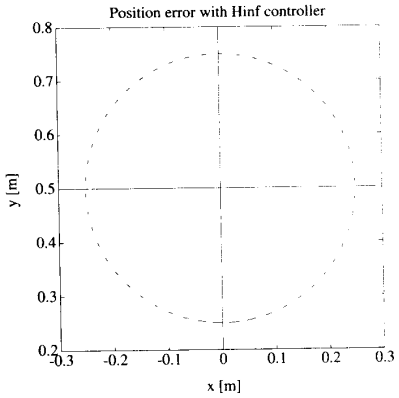

Figure 6: Position error

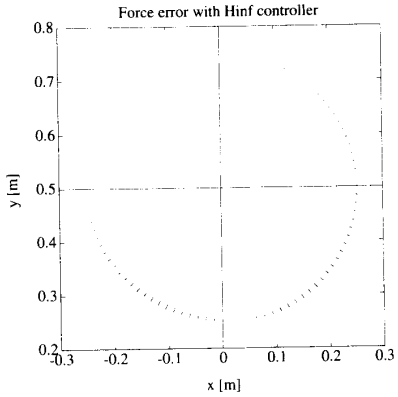

Figure 7: Force error

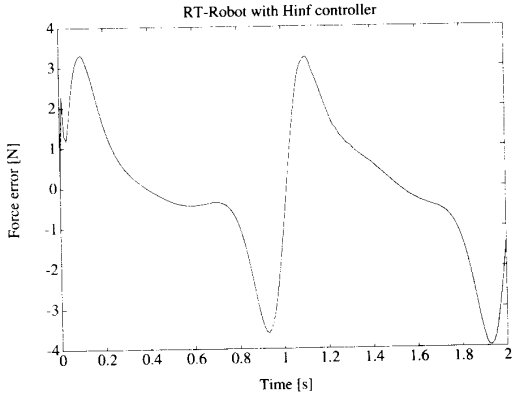

Figure 8: Force error against time

6 SIMULATIONS

The results are presented in three parts. First a sample of the time responses calculated during one of the simulations. Second an overview of the influence of the design frequency on the RMS errors, with fixed motor time constants. Third another overview, but now for a fixed design frequency but varying motor time constants. After that the results are discussed.

6.1 Results

A sample of the time responses is in Figs. 6 - 8. The results are obtained with a H_∞ controller designed for the nominal motor time constants and a nominal design frequency ω_0 of 400 [rad/s]. In Fig. 6 the position error for $t = 1 - 2$ [s] is projected on the reference trajectory. The position error is scaled with a factor of 400, but the position error is still very small, in the order of several [μm]. In Fig. 7 the force error for $t = 1 - 2$ [s] is presented, also projected on the reference trajectory. The length of the lines is a measure of the force error. The scaling is such that a line from the circumference to the center is equal to 100 [N], so the figure shows that the force error is also small, in the order of several [N]. Fig. 8, force error against time, shows that the system is still far from the stability limit, because of the fast damping of the oscillations, initiated by incorrect initial conditions of the controllers.

The RMS values of position and force error against design frequency ω_0 for nominal motor time constants of $\frac{1}{400}$ [s] are presented, for comparison purposes, in Fig. 9 and Fig. 10. The errors are large for small design frequencies, caused by loose control and diminishing with larger design frequencies because the control becomes more tight. The levels of the RMS values for the three control designs are different. Also for higher design frequencies, the controllers are unable to stabilize the system, so for $\omega_0 > 800$ [rad/s] no RMS results for the controllers are given.

The RMS values of position and force error against motor time constant, for a nominal design frequency of 400 [rad/s] and H_∞ controller, are presented in Figs. 11 and 12. The errors increase monotonically for larger time constants, caused by an earlier perceptible excitation of the unmodeled dynamics.

6.2 Discussion

The results presented above give rise to the following remarks:

- performance
  - the PD and H_∞ controllers have the best performance with respect to tracking and force error,
  - the robustness of the controllers is comparable, this can be attributed to the fact that the design problem is a SISO prob-

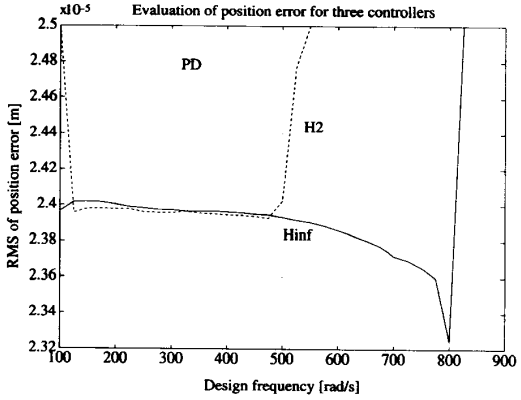

Figure 9: RMS values of position against design frequency

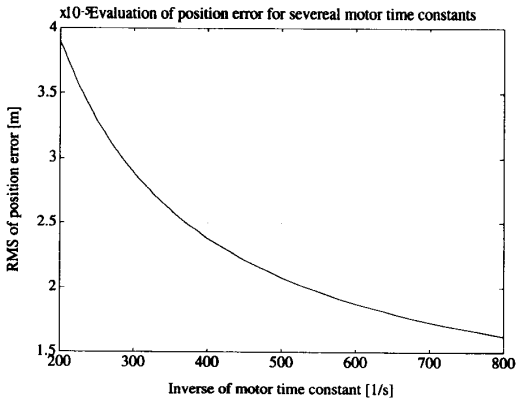

Figure 11: RMS values of position error against motor time constants

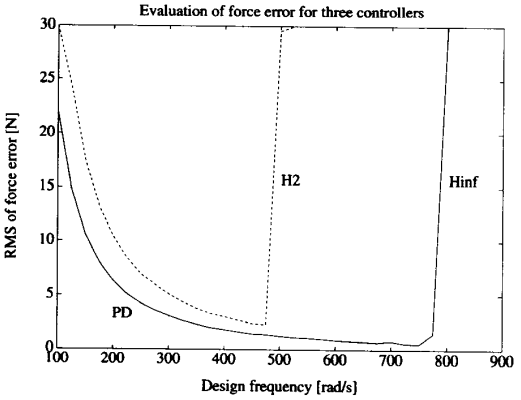

Figure 10: RMS values of force error against design frequency

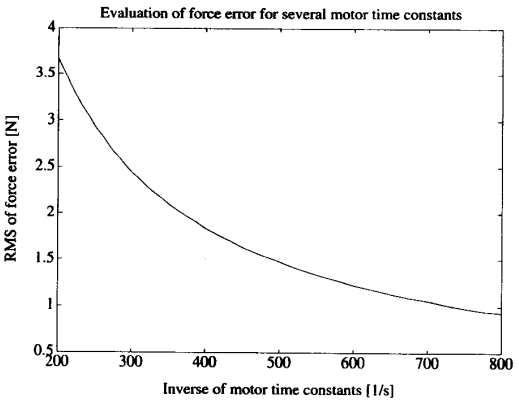

Figure 12: RMS values of force error against motor time constants

lem, so the only potential advantage left for the H_∞ and H_2 controllers are the loop shaping capabilities, which, for this problem could not improve on the PD controller,

- implementation
  - the computational complexity of the PD controller is the lowest, of the H_2 controller the largest, this is related to the controller order squared,
  - the computational complexity of the position and force control loop can be neglected compared with the computed torque and position and force error calculations, which require time consuming trigonometric function evaluations,
- design
  - the PD controller is the easiest to compute, but with the arrival of control system design (CSD) programs that already incorporate design algorithms for H_2 and H_∞ controller design [9] this is no longer an important issue,
  - the PD controller is the easiest to design, only the design frequency ω_0 is used as design parameter; for the H_∞ and H_2 controllers much more effort is needed to select suitable weight functions.

7 CONCLUSIONS

The main conjecture that can be drawn from the computational experiments described above is that there are no advantages in using H_2 or H_∞ controllers instead of a PD controller, for the system, unmodeled dynamics, control task and controller structure investigated. This conjecture probably can be generalized to control problems where the controllers are designed for a decoupled system, resulting in a SISO design problem. The number of problems that fits in this class is not negligible.

To give this paper even more scientific content, it is recommended [10] to try to refute the conjecture given above. A fast way to do this is to select another application with different models and control tasks and repeat the computations. A more fundamental approach is to supply a rigorous proof. Good luck.

REFERENCES

[1] B. A. Francis, *A Course in H_∞ Control Theory*. Berlin: Springer-Verlag, 1987.

[2] G. Stein and J. C. Doyle, "Beyond singular values and loop shapes," *AIAA J Guidance Control*, 1988.

[3] J.-J. E. Slotine and W. Li, "Adaptive manipulator control: A case study," *IEEE Trans. Automat. Control*, vol. AC-33, pp. 995–1003, Nov. 1988.

[4] A. Isidori, *Nonlinear Control Systems: An Introduction*. Berlin: Springer-Verlag, 2nd ed., 1989.

[5] A. van der Schaft, "On a state space approach to nonlinear H_∞ control," *Systems Control Lett.*, vol. 16, pp. 1–8, Jan. 1991.

[6] H. Asada and J.-J. E. Slotine, *Robot Analysis and Control*. Chichester: Wiley-Interscience, 1986.

[7] M. G. Safonov, D. J. N. Limebeer, and R. Y. Chiang, "Simplifying the H^∞ theory via loop-shifting, matrix-pencil and descriptor concepts," *Internat. J. Control*, vol. 50, pp. 2467–2488, Dec. 1989.

[8] J. C. Doyle, K. Glover, P. P. Khargonekar, and B. A. Francis, "State-space solutions to standard H_2 and H_∞ control problems," *IEEE Trans. Automat. Control*, vol. AC-34, pp. 831–847, Aug. 1989.

[9] R. Y. Chiang and M. G. Safonov, "Robust-control toolbox, for use with MATLAB." The MathWorks, Inc., June 1988.

[10] K. R. Popper, *Conjectures and Refutations, The Growth of Scientific Knowledge*. London and Henley: Routledge & Kegan Paul Ltd., fourth ed., 1972.