Dynamic Positioning of Surface Vessel: Multivariable Frequency Analysis and Controller Design

Erling Aarsand Johannessen

Department of Engineering Cybernetics
The Norwegian Institute of Technology
University of Trondheim
N-7034 Trondheim, Norway

Telephone:	+47 7359 4348
Telefax:	+47 7359 4399
E-mail:	Erling.Johannessen@itk.unit.no

Project Work in
Course 43917 Multivariable Frequency Analysis

March 27, 1996

Abstract

The control problem in dynamic positioning is presented. A detailed description of vessel model and disturbances is given.

Multivariable frequency analysis is applied to plant and disturbance model. In particular, singular value analysis and relative gain array is considered. Criteria for assessing nominal performance, robust stability and robust performance is given.

PID, $\boldsymbol{H}_2$ and $\boldsymbol{H}_\infty$ controllers are designed. All satisfy robust stability. Only the $\boldsymbol{H}_2$ and $\boldsymbol{H}_\infty$ designs satisfy nominal performance. Robust performance is not satisfied for any of the designs, due to the fact that uncertainty was not explicitly taken into account during design. This could have been done in $\boldsymbol{H}_\infty$.

Time simulations of all three designs are provided. They show good rejection of disturbances for $\boldsymbol{H}_2$ and $\boldsymbol{H}_\infty$.

Preface

This report summarizes a project work in the course 43917 Multivariable frequency analysis.

The main reference is:

Sigurd Skogestad, *Multivariable feedback control — Analysis and design using frequency-domain methods*, University of Trondheim - NTH, 1994.

Other references are cited.

Acknowledgement

Per J. Nicklasson has provided assistance with Matlab details.

Erling Aa. Johannessen

Contents

Figures

Tables

1 Introduction and problem statement

1.1 What the problem is all about

The main objective is to control the position and course of a surface vessel which is subject to disturbances:

- current
- wind
- waves

The vessel is equipped with r thrusters (propellers).

An illustration of the control problem is given in figure 1.

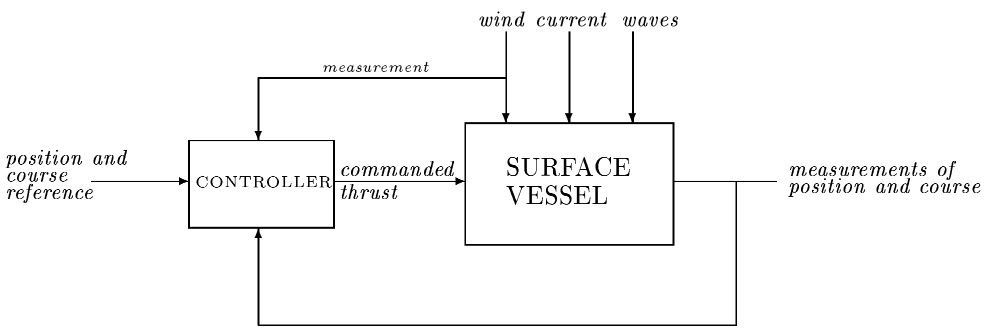


Figure 1: *Control problem*

1.2 Inputs and outputs

1.2.1 Disturbances

- *Wind* can be divided in a dominating low frequency component and high frequency gust.
- *Current* is a low frequency phenomenon.
- *Waves* cause disturbance in a certain frequency band, depending on the *sea state*. The frequency with the dominating disturbance is called the *wave frequency*.

1.2.2 Measurements

Vessel position, (x, y).

Vessel course, ψ.

Wind speed and direction.

1.2.3 Noise

High frequency noise is present in all measurements.

1.2.4 Controlled outputs

Low frequency components (x_L, y_L), ψ_L of vessel position and course shall be controlled. Low frequency in the sense that motion due to high frequency wave disturbances shall not be compensated, as this would cause unnecessary wear and tear on the thrusters.

1.2.5 Setpoints

Vessel position and course.

1.2.6 Manipulated inputs

Resulting force and moment from r individual thrusters (propellers), i.e. force in the surge (forward) and sway (sideways) directions and moment about the vertical axis (yaw).

1.3 Control objective

1. The control error (difference between setpoints and controlled outputs) shall be small (in some sense). In particular, low frequency disturbances shall be rejected. This suggests that integral action should be applied.
2. The transfer function from wave disturbances to manipulated inputs shall be small (in some sense), i.e. the control input shall not be influenced by these disturbances.
3. The control input shall not be influenced by measurement noise.

1.4 Sources of model uncertainty

To obtain high performance, some kind of model of the wave disturbances should be included in the control design. Here we will have uncertainty in:

- wave frequency
- amplitude of the disturbance

Other sources of model uncertainty:

- hydrodynamic coefficients in vessel model
- thruster model (input uncertainty)

1.5 Expected control problems

No particular control problems are expected.

This problem is usually solved using LQG control. It will be interesting to compare this (H_2) with H_∞. The latter method will provide a structured way of approaching model uncertainty. In particular, robust performance will be of interest.

1.6 Simplifications

- To simplify matters somewhat, wind disturbances will not be considered further.
- The influence of wave disturbances is not stressed in controller design and simulations. Set-point changes and rejection of constant disturbances is focused.

2 Vessel model

Most of this section is taken from [1]. Background material may also be found in [2].

2.1 Kinematics

To describe the motion of the vessel we will need two coordinate systems:

- A vessel fixed frame of reference, $x_B y_B z_B$, called B.
- An earth fixed frame of reference, xyz, called 0.

These are shown in figure 2.

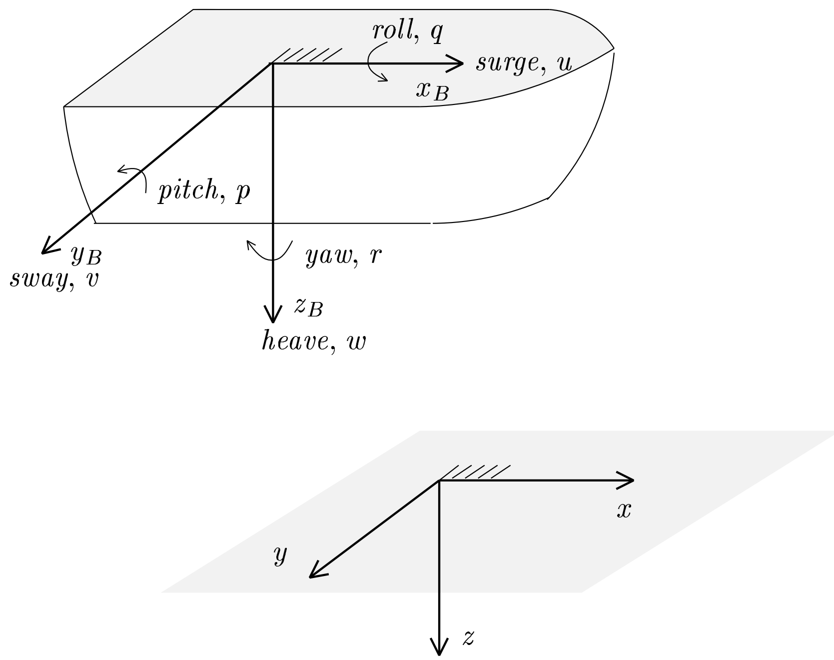


Figure 2: *Frames of reference*

Three degrees of freedom are of importance in dynamic positioning:

- surge
- sway
- yaw

Velocites may be expressed in B, i.e.

$$\boldsymbol{\nu} = \begin{pmatrix} u \\ v \\ r \end{pmatrix} \tag{1}$$

These velocities cannot be integrated in B to give position and orientation, but will have to be transformed to 0.

$$\dot{\boldsymbol{\eta}} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{pmatrix} = \boldsymbol{J}(\psi)\boldsymbol{\nu} = \begin{pmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} u \\ v \\ r \end{pmatrix} \tag{2}$$

2.2 Low frequency vessel model

$$\boldsymbol{M}\dot{\boldsymbol{\nu}}_L + \boldsymbol{D}(\boldsymbol{\nu}_L - \boldsymbol{\nu}_C) = \boldsymbol{\tau}_L + \boldsymbol{w}_L \tag{3}$$

where

$$\boldsymbol{\nu}_L = \begin{pmatrix} u_L \\ v_L \\ r_L \end{pmatrix}$$

is the low frequency velocity vector,

$$\boldsymbol{\nu}_C = \begin{pmatrix} u_C \\ v_C \\ r_C \end{pmatrix}$$

is the current velocity, $\boldsymbol{\tau}_L$ is a vector of control forces and moment, and $\boldsymbol{w}_L$ is a vector (of zero-mean Gaussian white noise processes) describing unmodelled dynamics and disturbances.

Now, we assume

$$\boldsymbol{M} = \begin{pmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - Y_{\dot{v}} & mx_G - Y_{\dot{r}} \\ 0 & mx_G - Y_{\dot{r}} & I_z - N_{\dot{r}} \end{pmatrix} \tag{4}$$

where x_G is the x-coordinate of the center of gravity of the vessel.

The damping matrix is given by

$$\boldsymbol{D} = \begin{pmatrix} -X_u & 0 & 0 \\ 0 & -Y_v & -Y_r \\ 0 & -N_v & -N_r \end{pmatrix} \tag{5}$$

Suppose

$$\boldsymbol{\eta}_d = \begin{pmatrix} x_d \\ y_d \\ \psi_d \end{pmatrix}$$

is the desired position and orientation, $\psi \approx \psi_d$, and the earth fixed frame of reference is oriented so that $\psi_d = 0$. Then $\boldsymbol{J}(\psi) \approx \boldsymbol{I}$.

We then have the following state space model:

$$\dot{\boldsymbol{x}}_L = \boldsymbol{A}_L\boldsymbol{x}_L + \boldsymbol{B}_L\boldsymbol{\tau}_L + \boldsymbol{E}_L\boldsymbol{w}_L \tag{6}$$

where

$$\boldsymbol{x}_L = (x_L, y_L, \psi_L, u_L, v_L, r_L)^{\mathrm{T}}$$

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{0} & -\boldsymbol{M}^{-1}\boldsymbol{D} \end{pmatrix}, \quad \boldsymbol{B}_L = \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{M}^{-1} \end{pmatrix}, \quad \boldsymbol{E}_L = \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{M}^{-1} \end{pmatrix}$$

2.3 High frequency wave model

A linear approximation to the Pierson-Moskowitz spectrum [3] is given by

$$h(s) = \frac{K_w s}{s^2 + 2\zeta\omega_0 s + \omega_0^2}. \tag{7}$$

The parameter K_w depends on the sea state, while ζ and ω_0 are tuning parameters.

We have the following model for the wave disturbances in the three degrees of freedom:

$$\begin{aligned} \dot{\xi}_x &= x_H \\ \dot{x}_H &= -2\zeta\omega_0 x_H - \omega_0^2 \xi_x + K_{wx} w_x \end{aligned} \tag{8}$$

$$\begin{aligned} \dot{\xi}_y &= y_H \\ \dot{y}_H &= -2\zeta\omega_0 y_H - \omega_0^2 \xi_y + K_{wy} w_y \end{aligned} \tag{9}$$

$$\begin{aligned} \dot{\xi}_\psi &= \psi_H \\ \dot{\psi}_H &= -2\zeta\omega_0 \psi_H - \omega_0^2 \xi_\psi + K_{w\psi} w_\psi \end{aligned} \tag{10}$$

This gives the following state space model

$$\dot{\boldsymbol{x}}_H = \boldsymbol{A}_H \boldsymbol{x}_H + \boldsymbol{E}_H \boldsymbol{w}_H \tag{11}$$

where

$$\boldsymbol{x}_H = (\xi_x, \xi_y, \xi_\psi, x_H, y_H, \psi_H)^{\mathrm{T}}, \quad \boldsymbol{w}_H = (w_x, w_y, w_\psi)^{\mathrm{T}}$$

$$\boldsymbol{A}_H = \text{block diag}\left\{ \begin{pmatrix} 0 & 1 \\ -\omega_0^2 & -2\zeta\omega_0 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ -\omega_0^2 & -2\zeta\omega_0 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ -\omega_0^2 & -2\zeta\omega_0 \end{pmatrix} \right\}$$

2.4 Current model

2.4.1 Varying velocity and direction

Let V_C denote current velocity and β_C denote current direction.

These are assumed to be slowly varying parameters, i.e

$$\dot{V}_C = w_{V_C} \tag{12}$$

$$\dot{\beta}_C = w_{\beta_C} \tag{13}$$

where w_{V_C} and w_{β_C} are zero-mean Gaussian white noise processes.

Transformed to B we get

$$\begin{aligned} u_C &= V_C \cos(\beta_C - \psi_L - \psi_H) \qquad (14) \\ v_C &= V_C \sin(\beta_C - \psi_L - \psi_H) \qquad (15) \\ \dot{r}_C &= w_r \qquad (16) \end{aligned}$$

Note that r_C is a non-physical quantity.

We then have the following state space model:

$$\dot{\boldsymbol{x}}_C = \boldsymbol{E}_C \boldsymbol{w}_C \qquad (17)$$

where

$$\boldsymbol{x}_C = (V_C, \beta_C, r_C)^{\mathrm{T}}, \quad \boldsymbol{w}_C = (w_{V_C}, w_{\beta_C}, w_r)^{\mathrm{T}}, \quad \boldsymbol{E}_C = \boldsymbol{I}$$

2.4.2 Simplified linearized model

Linearizing equations (14) and (15), we get:

$$\begin{aligned} \Delta u_C &= \cos(\beta_{C0} - \psi_{L0})\Delta V_C - V_{C0}\sin(\beta_{C0} - \psi_{L0})(\Delta\beta_C - \Delta\psi_L - \Delta\psi_H \qquad (18) \\ \Delta v_C &= \sin(\beta_{C0} - \psi_{L0})\Delta V_C + V_{C0}\cos(\beta_{C0} - \psi_{L0})(\Delta\beta_C - \Delta\psi_L - \Delta\psi_H \qquad (19) \\ & \qquad (20) \end{aligned}$$

We will use $V_{C0} = 0$, and neglect r_C.

With this simplified model we can write

$$\boldsymbol{\nu}_C = \tilde{\boldsymbol{E}}_C \boldsymbol{w}_C, \qquad (21)$$

where

$$\boldsymbol{w}_c = \Delta V_C \qquad (22)$$

and

$$\tilde{\boldsymbol{E}}_C = \begin{pmatrix} \cos(\beta_{C0} - \psi_{L0}) \\ \sin(\beta_{C0} - \psi_{L0}) \\ 0 \end{pmatrix} \qquad (23)$$

2.5 Thruster model

Thruster dynamics is approximated by a time constant in each degree of freedom, i.e

$$\dot{\boldsymbol{\tau}}_L = \boldsymbol{A}_{thr}(\boldsymbol{\tau}_L - \boldsymbol{\tau}_{com}), \qquad (24)$$

where $\boldsymbol{\tau}_{com}$ is commanded thrust and

$$\boldsymbol{A}_{thr} = \mathrm{diag}\left\{-\frac{1}{T_X}, -\frac{1}{T_Y}, -\frac{1}{T_N}\right\}. \qquad (25)$$

Assuming pitch controlled thrusters, commanded thrust is given by

$$\boldsymbol{\tau}_{com} = \boldsymbol{TKu} \tag{26}$$

where $\boldsymbol{u} \in \mathbb{R}^r$ (r individual thrusters), given by

$$\boldsymbol{u} = (|p_1|p_1, |p_2|p_2, \dots, |p_r|p_r,)^{\mathrm{T}} \tag{27}$$

is the control variable. p_i $(i = 1, \dots, r)$ are the so-called *pitch ratios.*

$\boldsymbol{K}$ is a diagonal matrix of thruster force coefficients, i.e

$$\boldsymbol{K} = \operatorname{diag}\{K_1(n_1), K_2(n_2), \dots, K_r(n_r)\}. \tag{28}$$

We note that the force coefficients depends on the propeller revolution, n_i.

The matrix $\boldsymbol{T} \in \mathbb{R}^{3\times r}$ describes the geometric configuration of the thrusters, and is perfectly known.

Example 1 ([1], vessel with 6 thrusters)

$$\boldsymbol{T} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ \ell_1 & -\ell_2 & -\ell_3 & -\ell_4 & \ell_5 & \ell_6 \end{pmatrix} \tag{29}$$

△

2.6 Measurement model

The following physical parameters are measured:

- x-position
- y-position
- course

This gives the following measurement model:

$$y_1 = x_L + x_H + n_1 \tag{30}$$
$$y_2 = y_L + y_H + n_2 \tag{31}$$
$$y_3 = \psi_L + \psi_H + n_3 \tag{32}$$

2.7 Total vessel model

$$\dot{\boldsymbol{x}} = \boldsymbol{Ax} + \boldsymbol{Bu} + \boldsymbol{Ew} \tag{33}$$
$$\boldsymbol{y} = \boldsymbol{Cx} + \boldsymbol{n} \tag{34}$$

where

$$\boldsymbol{x} = \left(\boldsymbol{x}_L^{\mathrm{T}}, \boldsymbol{x}_H^{\mathrm{T}}, \boldsymbol{\tau}_L^{\mathrm{T}}\right)^{\mathrm{T}}, \tag{35}$$

$$\begin{pmatrix} \dot{\boldsymbol{x}}_L \\ \dot{\boldsymbol{x}}_H \\ \dot{\boldsymbol{\tau}}_L \end{pmatrix} = \begin{pmatrix} \boldsymbol{A}_L & \boldsymbol{0} & \boldsymbol{B}_L \\ \boldsymbol{0} & \boldsymbol{A}_H & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{A}_{thr} \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_L \\ \boldsymbol{x}_H \\ \boldsymbol{\tau}_L \end{pmatrix} + \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ -\boldsymbol{A}_{thr}\boldsymbol{T}\boldsymbol{K} \end{pmatrix} \boldsymbol{u} + \begin{pmatrix} \boldsymbol{E}_C & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{E}_H \\ \boldsymbol{0} & \boldsymbol{0} \end{pmatrix} \boldsymbol{w}, \tag{36}$$

$$\boldsymbol{w} = \left(\boldsymbol{w}_C, \boldsymbol{w}_H^{\mathrm{T}}\right)^{\mathrm{T}}, \tag{37}$$

$$\boldsymbol{E}_C = \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{M}^{-1}\boldsymbol{D}\tilde{\boldsymbol{E}}_C \end{pmatrix}. \tag{38}$$

A block diagram is given in figure 3.

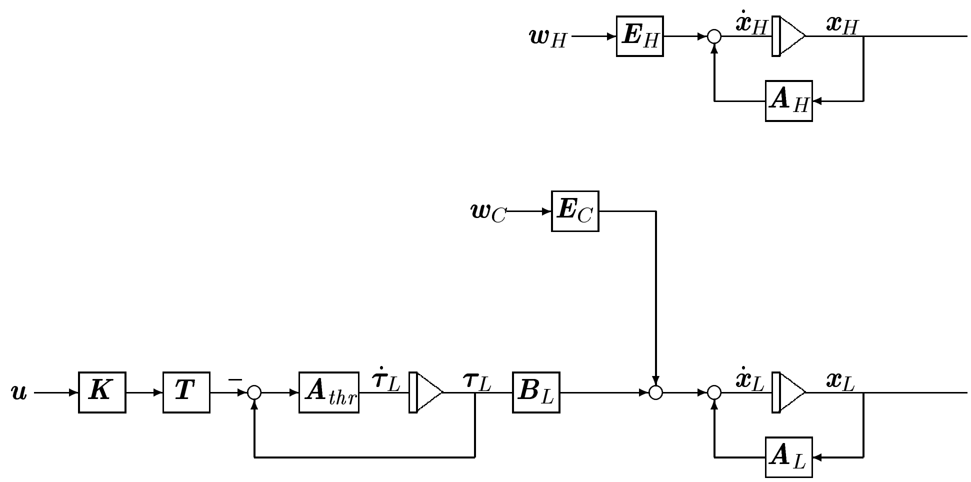


Figure 3: *Block diagram*

2.8 Numerical values and scaling

In [1] data for a vessel is given:

$$\begin{aligned} \boldsymbol{M}''\dot{\boldsymbol{\nu}}'' + \boldsymbol{D}''\boldsymbol{\nu}'' &= \boldsymbol{T}''\boldsymbol{K}''\boldsymbol{u}'' & (39) \\ \dot{\boldsymbol{\nu}}'' &= \tilde{\boldsymbol{A}}''\boldsymbol{\nu}'' + \tilde{\boldsymbol{B}}''\boldsymbol{u}'', & (40) \end{aligned}$$

where

$$\boldsymbol{T}'' = \begin{pmatrix} 1.0000 & 0 & 0 & 0 & 0 \\ 0 & 1.0000 & 1.0000 & 1.0000 & 1.0000 \\ 0 & -0.3937 & -0.3937 & 0.4514 & 0.3215 \end{pmatrix}.$$

The following values are estimated:

$$\boldsymbol{K}'' = \mathrm{diag}\,\{0.0260, 0.0140, 0.0140, 0.0015, 0.0053\}$$

$$\boldsymbol{M}'' = \begin{pmatrix} 1.1274 & 0 & 0 \\ 0 & 1.8902 & -0.0744 \\ 0 & -0.0744 & 0.1278 \end{pmatrix}$$

$$\tilde{\boldsymbol{A}}'' = \begin{pmatrix} -0.0367 & 0 & 0 \\ 0 & -0.0933 & -0.0073 \\ 0 & 0.1882 & -0.3310 \end{pmatrix}$$

All quantities with double prime $(\cdot)''$ are scaled with the so-called bis system [1], see table 1.

Table 1: *Bis system*

Unit	*Scaling factor*	
Length	L	(L: vessel length)
Mass	$\rho\nabla_0$	(water density × vessel displacement)
Moment of intertia	$\rho\nabla_0 L^2$	
Time	$\sqrt{L/g}$	(g: acceleration of gravity)
Reference area	$2\nabla_0/L$	
Position	L	
Angle	1	
Linear velocity	$\sqrt{Lg}$	
Angular velocity	$\sqrt{g/L}$	
Linear acceleration	g	
Angular acceleration	g/L	
Force	$\rho g\nabla_0$	
Torque	$\rho g\nabla_0 L$	

We <u>select</u> the following maximum deviations in outputs, disturbances etc.:

$$\begin{aligned} |\Delta \boldsymbol{w}_C|_{max} &= 0.5 \text{ m/s} \\ |\Delta \boldsymbol{w}_H|_{max} &= 1.0 \\ |\Delta \boldsymbol{u}_i|_{max} &= 1 \qquad \text{(normalized pitch angle)} \\ |\Delta \boldsymbol{e}_x|_{max} &= 5 \text{ m} \\ |\Delta \boldsymbol{e}_y|_{max} &= 5 \text{ m} \\ |\Delta \boldsymbol{e}_\psi|_{max} &= 3^\circ \end{aligned}$$

Thus, the scaled transfer function matrices of the plant is given by

$$\boldsymbol{G}(s) = \boldsymbol{D}_e \boldsymbol{C}(s\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{B}\boldsymbol{D}_u \qquad (41)$$

$$\boldsymbol{G}_d(s) = \boldsymbol{D}_e \boldsymbol{C}(s\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{E}\boldsymbol{D}_d \qquad (42)$$

2.9 Thruster configuration

The thruster configuration is shown in figure 4.

Note that the main propellers run in parallell, and thus constitutes <u>one</u> control input (u_1).

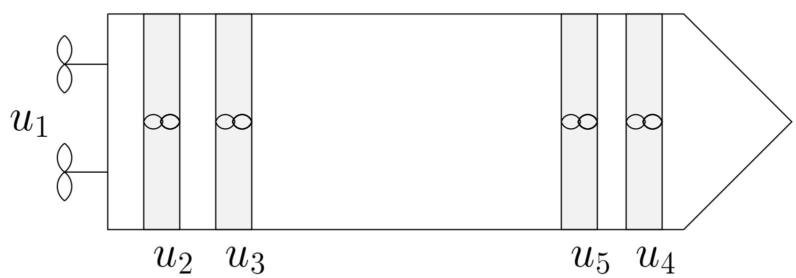


Figure 4: *Thruster configuration*

3 Analysis

3.1 Gain and phase element by element

Plot of gain and phase of elements in $\boldsymbol{G}$ is shown in figures 5 and 6.

Plot of gain and phase of elements in $\boldsymbol{G}_d$ is shown in figures 7 and 8.

3.2 Poles and zeros

Poles and zeros of elements in $\boldsymbol{G}(s)$ is given in table 2.

Poles and zeros of elements in $\boldsymbol{G}_d(s)$ is given in table 3.

Poles of $\boldsymbol{G}(s)$:

```
  -1.1664e-01
  -3.5603e-02
  -1.3168e-02
  -2.0000e-01
  -2.0000e-01
  -2.0000e-01
  -1.0000e-10
  -1.0000e-10
  -1.0000e-10
```

Zeros of $\boldsymbol{G}(s)$: (`tzero`)

```
None
```

Poles of $\boldsymbol{G}_d(s)$:

```
 -1.1664e-01
 -1.0000e-10
 -3.5603e-02
 -1.3168e-02
 -3.1416e-01 + 5.4414e-01i
 -3.1416e-01 - 5.4414e-01i
 -3.1416e-01 + 5.4414e-01i
 -3.1416e-01 - 5.4414e-01i
 -3.1416e-01 + 5.4414e-01i
 -3.1416e-01 - 5.4414e-01i
```

Zeros of $G_d(s)$: (`tzero`)

```
 -1.1213e-15 + 6.2718e-16i
 -1.1213e-15 - 6.2718e-16i
```

Table 2: *Poles and zeros of elements in $\boldsymbol{G}(s)$*

G_{ij}		1	2	3	4	5
1	Poles	-1.0000e-10 -1.3168e-02 -2.0000e-01	None	None	None	None
	Zeros	None	None	None	None	None
	Steady state gain	$-\infty$	0	0	0	0
2	Poles	None	-2.0000e-01 -1.0000e-10 -3.5603e-02 -1.1664e-01	-2.0000e-01 -1.0000e-10 -3.5603e-02 -1.1664e-01	-1.0000e-10 -1.1664e-01 -2.0000e-01 -3.5603e-02	-1.0000e-10 -1.1664e-01 -2.0000e-01 -3.5603e-02
	Zeros	None	-1.3657e-01	-1.3657e-01	-1.0371e-01	-1.0699e-01
	Steady state gain	0	$-\infty$	$-\infty$	$-\infty$	$-\infty$
3	Poles	None	-1.0000e-10 -3.5603e-02 -1.1664e-01 -2.0000e-01	-1.0000e-10 -3.5603e-02 -1.1664e-01 -2.0000e-01	-1.0000e-10 -3.5603e-02 -2.0000e-01 -1.1664e-01	-2.0000e-01 -1.0000e-10 -3.5603e-02 -1.1664e-01
	Zeros	None	-2.3545e-02	-2.3545e-02	-4.5224e-02	-4.8496e-02
	Steady state gain	0	∞	∞	$-\infty$	$-\infty$

Table 3: *Poles and zeros of elements in* $\boldsymbol{G}_d(s)$

$G_{d,ij}$		1	2	3	4
1	Poles	-1.0000e-10 -1.3168e-02	-3.1416e-01 + 5.4414e-01i -3.1416e-01 - 5.4414e-01i	None	None
	Zeros	None	0	None	None
	Steady state gain	∞	0	0	0
2	Poles	-1.1664e-01 -3.5603e-02 -1.0000e-10	None	-3.1416e-01 + 5.4414e-01i -3.1416e-01 - 5.4414e-01i	None
	Zeros	-1.2405e-01	None	0	None
	Steady state gain	∞	0	0	0
3	Poles	-3.5603e-02 -1.1664e-01	None	None	-3.1416e-01 + 5.4414e-01i -3.1416e-01 - 5.4414e-01i
	Zeros	None	None	None	0
	Steady state gain	-1.4410e+00	0	0	0

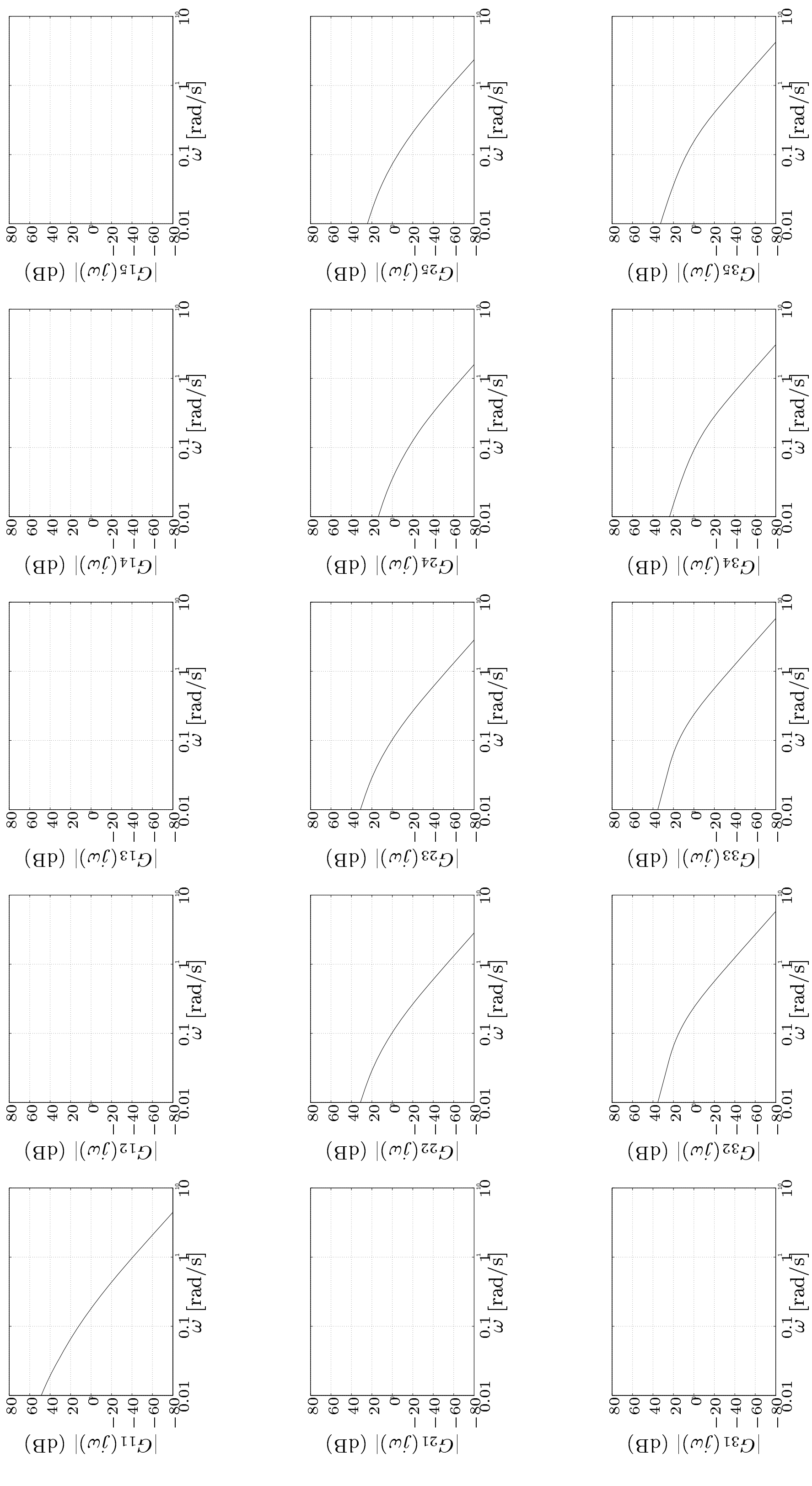


Figure 5: *Amplitude of elements in* $\boldsymbol{G}(j\omega)$

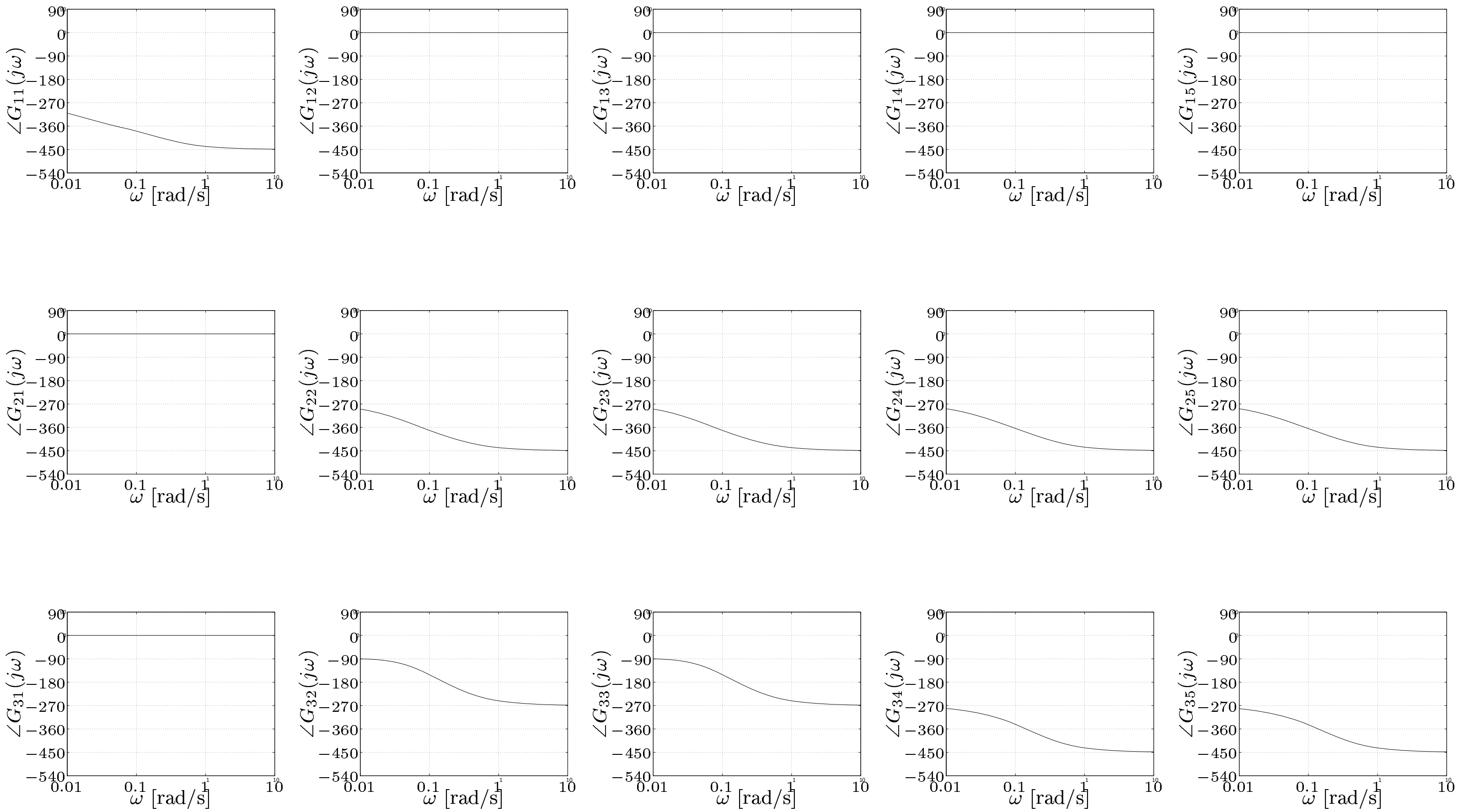


Figure 6: *Phase of elements in* $\boldsymbol{G}(j\omega)$

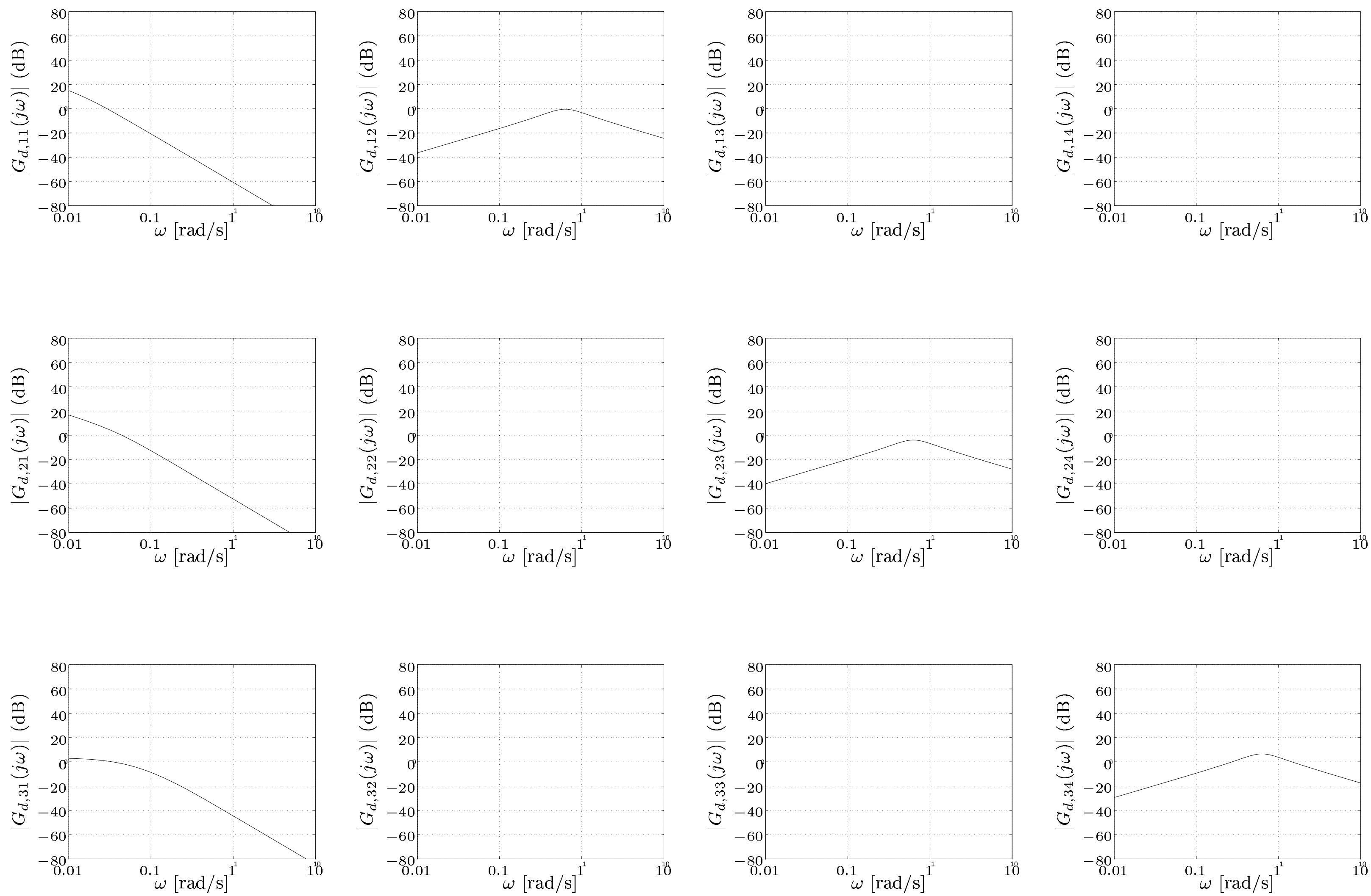


Figure 7: *Amplitude of elements in* $\boldsymbol{G}_d(j\omega)$

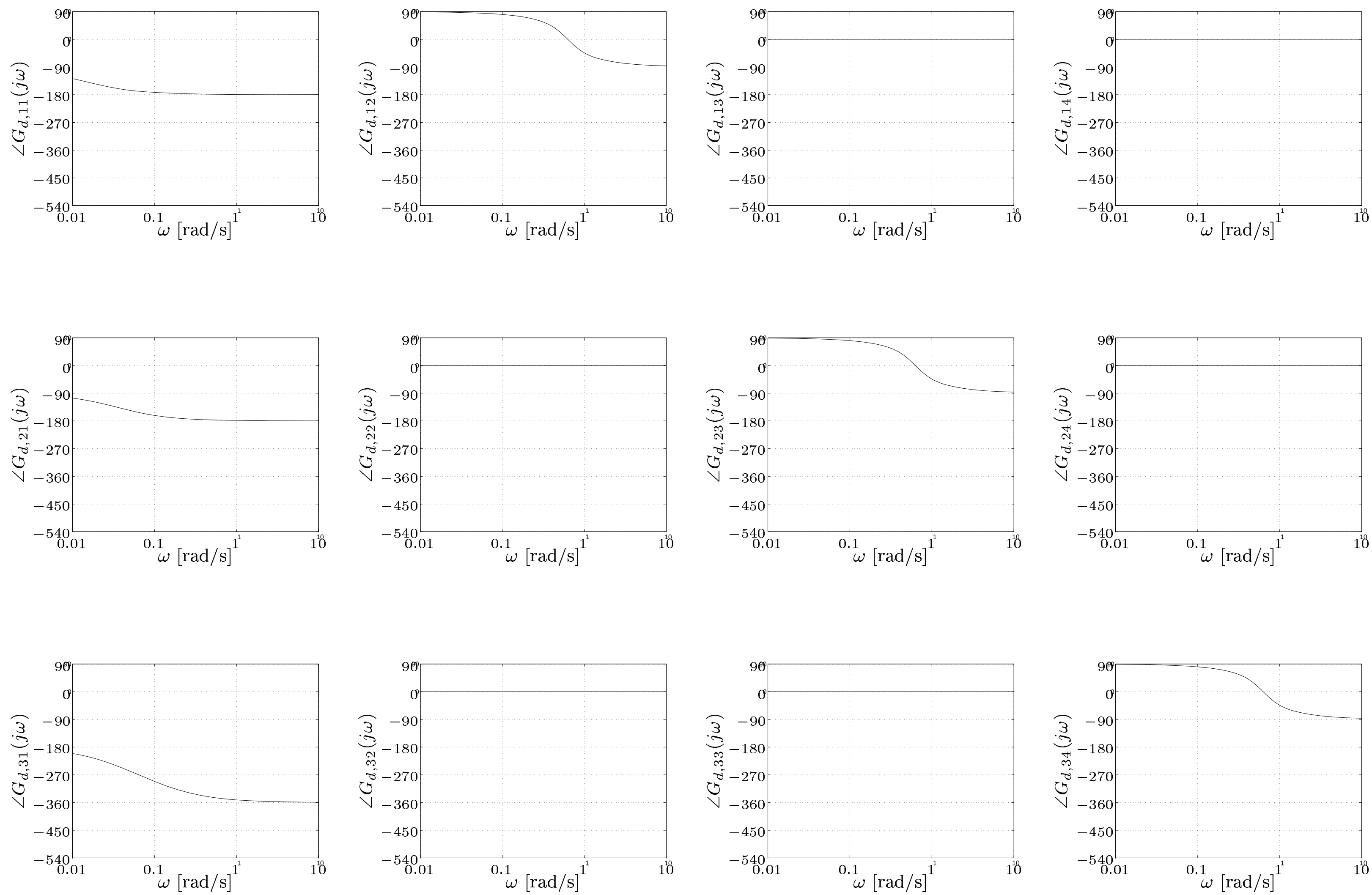


Figure 8: *Phase of elements in* $\boldsymbol{G}_d(j\omega)$

3.3 Singular value analysis

The singular values of $\boldsymbol{G}(j\omega)$ and $\boldsymbol{G}_d(j\omega)$ is shown in figure 9.

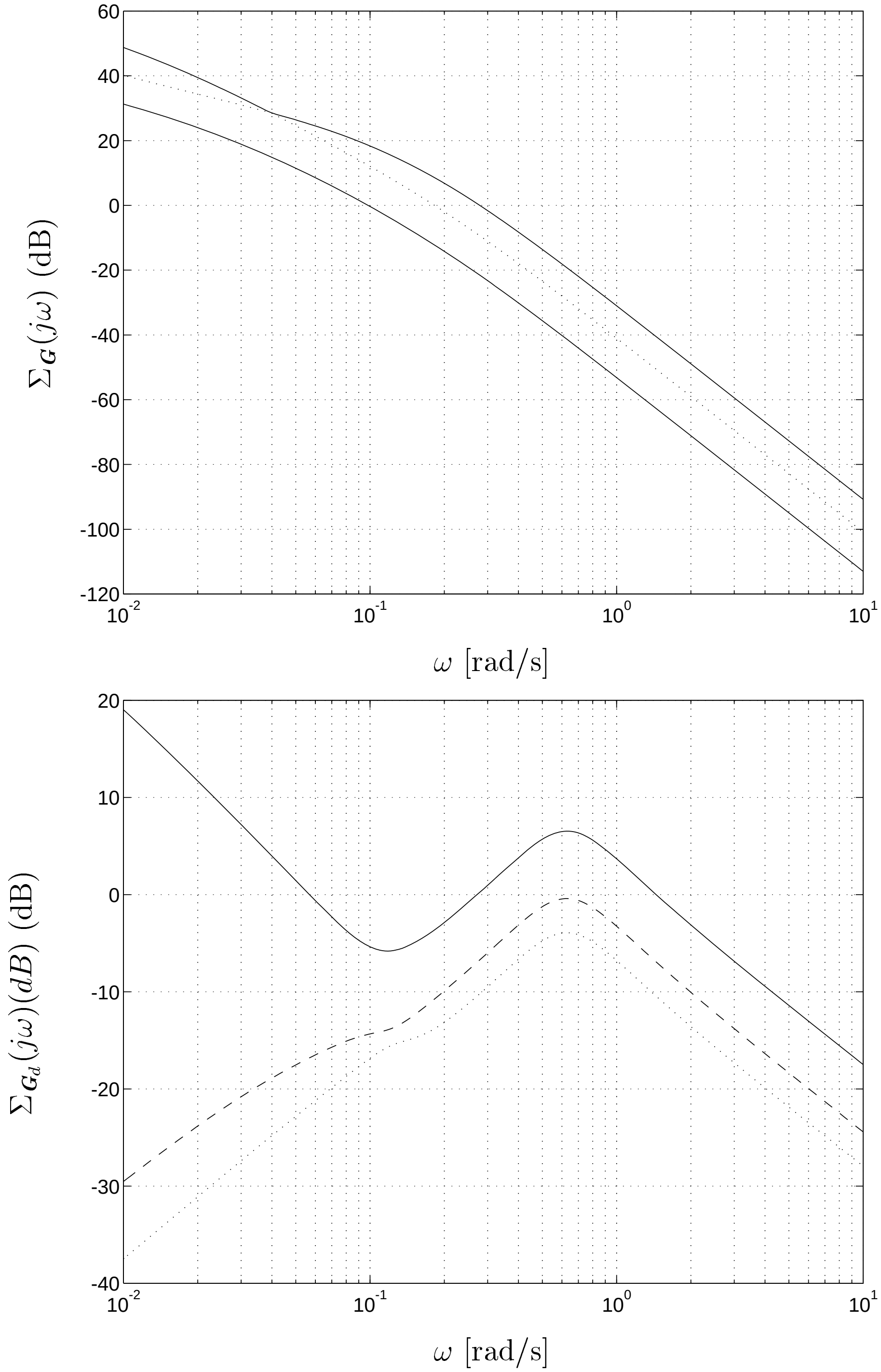


Figure 9: *Singular values of* $\boldsymbol{G}(j\omega)$ *and* $\boldsymbol{G}_d(j\omega)$

From this plot we immediately see that a bandwidth of 0.08 rad/s will be reasonable to reject the constant disturbance. We also see that it will <u>not</u> be possible to reject the disturbance caused by wave induced forces.

3.4 Relative gain array (RGA)

It is reasonable to use the pseudo-inverse of the thrust allocation matrix, $\boldsymbol{T}$, as decoupler.

This will produce a 3×3 plant. Figure 10 shows the magnitude of the elements in the RGA matrix as a function of frequency.

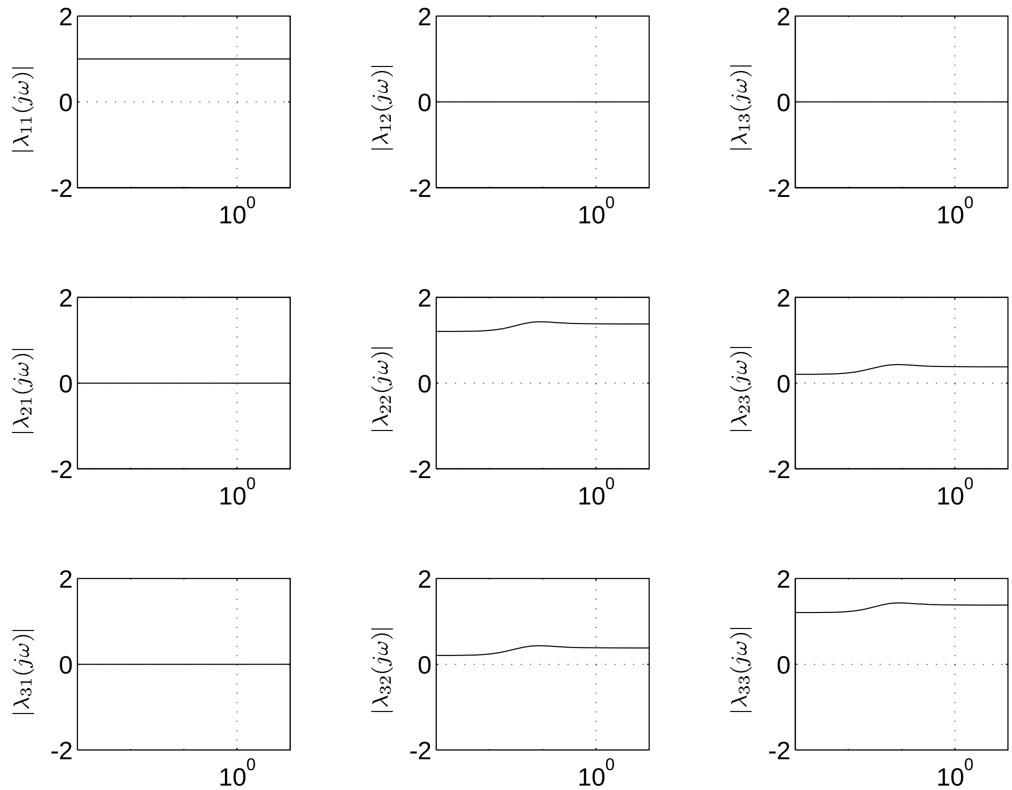


Figure 10: *Relative gain array of* $\boldsymbol{G}(s)\boldsymbol{T}^\dagger$

4 μ analysis

In this section, criteria for robust stability, nominal performance and robust performance are given.

In the next section, we will perform some controller designs, and check whether these criteria are met.

4.1 Performance weights and uncertainty weights

4.1.1 Selection of $\boldsymbol{W}_e$

The performance weight, $\boldsymbol{W}_e$, was selected to test whether the desired bandwidth of 0.08 rad/s was achieved.

Thus,

$$\boldsymbol{W}_e(s) = 0.5\frac{1 + 8 \cdot 10^{-2}s}{1 + 10^{-6}s}\boldsymbol{I}_{3\times 3} \tag{43}$$

A plot of this weight is shown in figure 11.

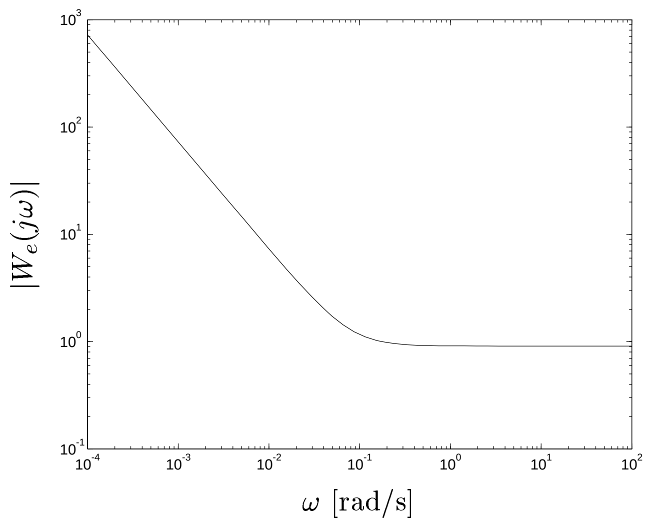


Figure 11: *Performance weight*

4.1.2 Selection of $\boldsymbol{W}_I$

The weight on input uncertainty, $\boldsymbol{W}_I$ reflects 5% uncertainty at low frequency and 100% uncertainty at high frequencies. The region of transition is chosen somewhat arbitrarily.

Thus,

$$\boldsymbol{W}_I(s) = \frac{1 + 0.5s}{1 + 10s}\boldsymbol{I}_{5\times 5} \tag{44}$$

A plot of this weight is shown in figure 12.

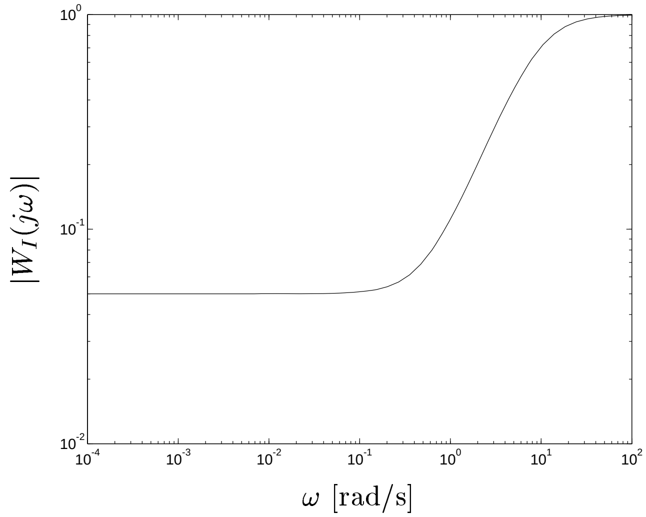


Figure 12: *Disturbance weight*

4.1.3 Selection of $\boldsymbol{W}_r$ and $\boldsymbol{W}_d$

We selected the following weight on references and disturbances:

$$\boldsymbol{W}_r(s) = 0.1\boldsymbol{I}_{3\times 3} \tag{45}$$

$$\boldsymbol{W}_d(s) = \begin{pmatrix} 0.01\frac{1}{1+0.01s} & 0 & 0 & 0 \\ 0 & 10^{-4} & 0 & 0 \\ 0 & 0 & 10^{-4} & 0 \\ 0 & 0 & 0 & 10^{-4} \end{pmatrix} \tag{46}$$

Rejection of disturbances caused by water current is thus more emphasized than response to set point changes, as the former part of the controller action is considered more important.

This is discussed further in section 7.

The disturbance from waves is given little weight, as this disturbance cannot be compensated.

4.2 Structure used in analysis

Consider figure 13 where a block diagram of the control system with weights is given.

With

$$w = \begin{pmatrix} \boldsymbol{r} \\ \boldsymbol{d} \end{pmatrix}, \tag{47}$$

$$z = \boldsymbol{W}_e \boldsymbol{e}, \tag{48}$$

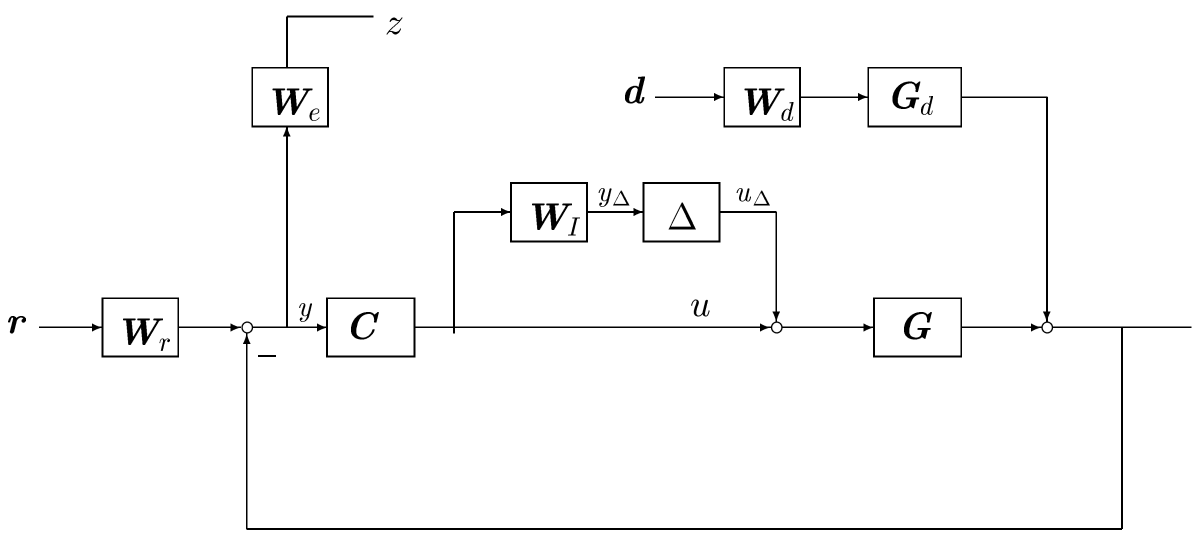


Figure 13: *Control system with weights*

$$K = \boldsymbol{C} \tag{49}$$

the block diagram in figure 13 can be transformed into the general structure of figure 14.

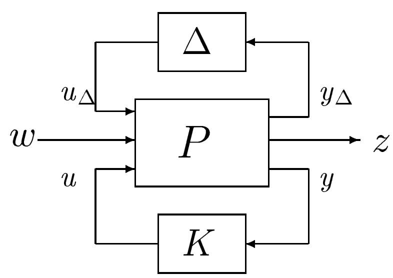


Figure 14: *General block diagram*

We then have

$$P = \left(\begin{array}{c|cc|c} 0 & 0 & 0 & \boldsymbol{W}_I \\ \hline -\boldsymbol{W}_e\boldsymbol{G} & \boldsymbol{W}_e\boldsymbol{W}_r & -\boldsymbol{W}_e\boldsymbol{G}_d\boldsymbol{W}_d & -\boldsymbol{W}_e\boldsymbol{G} \\ \hline -\boldsymbol{G} & \boldsymbol{W}_r & -\boldsymbol{G}_d\boldsymbol{W}_d & -\boldsymbol{G} \end{array}\right) \tag{50}$$

In this analysis the controller a given, and we will consider the following:

- robust stability — is the closed loop system stable when uncertainty is considered ?
- nominal performance — do we really achieve the desired performance ?
- robust performance — do we achieve the desired performance when uncertainty is considered as well ?

We consider structured uncertainty — diagonal on the input, and μ will have to be used.

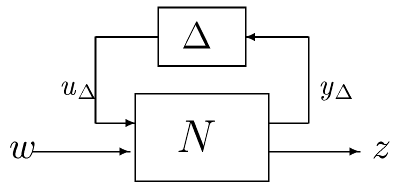


Figure 15: *Structure for μ analysis*

In figure 15 N is given by

$$
\begin{aligned}
N &= F_l(P,K) = P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21} && (51)\\
&= \left(\begin{array}{c|cc} -\boldsymbol{W}_I\boldsymbol{CG}(\boldsymbol{I}+\boldsymbol{CG})^{-1} & \boldsymbol{W}_I\boldsymbol{C}(\boldsymbol{I}+\boldsymbol{GC})^{-1}\boldsymbol{W}_r & -\boldsymbol{W}_I\boldsymbol{C}(\boldsymbol{I}+\boldsymbol{GC})^{-1}\boldsymbol{G}_d\boldsymbol{W}_d \\ \hline -\boldsymbol{W}_e(\boldsymbol{I}+\boldsymbol{CG})^{-1} & \boldsymbol{W}_e(\boldsymbol{I}+\boldsymbol{GC})^{-1}\boldsymbol{W}_r & -\boldsymbol{W}_e(\boldsymbol{I}+\boldsymbol{GC})^{-1}\boldsymbol{G}_d\boldsymbol{W}_d \end{array}\right) && (52)\\
&= \left(\begin{array}{c|cc} -\boldsymbol{W}_I\boldsymbol{CGS}_I & \boldsymbol{W}_I\boldsymbol{CSW}_r & -\boldsymbol{W}_I\boldsymbol{CSG}_d\boldsymbol{W}_d \\ \hline -\boldsymbol{W}_e\boldsymbol{S}_I & \boldsymbol{W}_e\boldsymbol{SW}_r & -\boldsymbol{W}_e\boldsymbol{SG}_d\boldsymbol{W}_d \end{array}\right) && (53)\\
&= \left(\begin{array}{c|c} N_{11} & N_{12} \\ \hline N_{21} & N_{22} \end{array}\right) && (54)
\end{aligned}
$$

where

$$
\begin{aligned}
\boldsymbol{S} &= (\boldsymbol{I}+\boldsymbol{GC})^{-1} && (55)\\
\boldsymbol{S}_I &= (\boldsymbol{I}+\boldsymbol{CG})^{-1} && (56)
\end{aligned}
$$

4.3 Robust stability

Criterion for robust stability:

$$
\begin{array}{c}
\text{RS} \\
\Updownarrow \\
\mu_\Delta(M) \quad < \quad 1, \quad \forall\omega,
\end{array} \qquad (57)
$$

where

$$M = N_{11} = -\boldsymbol{W}_I\boldsymbol{CG}(\boldsymbol{I}+\boldsymbol{CG})^{-1} \qquad (58)$$

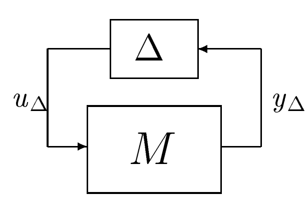


Figure 16: *Structure for assessing RS*

4.4 Nominal performance

Criterion for nominal performance:

$$
\begin{array}{rcl}
& \text{NP} & \\
& \Updownarrow & \\
\|N_{22}\|_\infty & < & 1, \quad \forall\omega,
\end{array}
\tag{59}
$$

where

$$
N_{22} = \left(\; \boldsymbol{W}_e(\boldsymbol{I} + \boldsymbol{G}\boldsymbol{C})^{-1}\boldsymbol{W}_r \quad -\boldsymbol{W}_e(\boldsymbol{I} + \boldsymbol{G}\boldsymbol{C})^{-1}\boldsymbol{G}_d\boldsymbol{W}_d \; \right)
\tag{60}
$$

4.5 Robust performance

Criterion for nominal performance:

$$
\begin{array}{rcl}
& \text{NP} & \\
& \Updownarrow & \\
\|F_u(N,\Delta)\|_\infty & < & 1, \quad \forall \|\Delta\|_\infty \leq 1, \\
& \Updownarrow & \qquad (61) \\
\mu_{\hat{\Delta}}(N) & < & 1, \quad \forall\omega \qquad (62)
\end{array}
$$

where

$$
\hat{\Delta} = \begin{pmatrix} \Delta & 0 \\ 0 & \Delta_P \end{pmatrix}
\tag{63}
$$

Δ_P is a full (7×3) matrix of fictitious uncertainty used to represent the performance requirement.

5 Controller design

5.1 PID controller

5.1.1 Design

We will use a PID controller in each degree of freedom, combined with $\boldsymbol{T}^\dagger$ as a decoupler/thruster allocation function.

As the RGA analysis shows, there is not much coupling between the surge motion and the two other, sway and yaw. There is, however, coupling between the sway and yaw motion. We will not expect satisfactory performance with a PID controller.

The control structure is shown in figure 17.

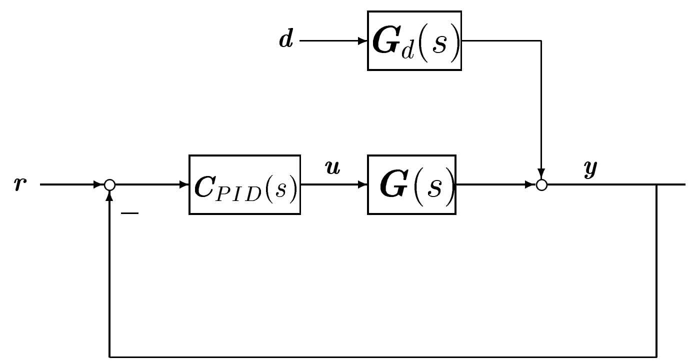


Figure 17: *PID controller*

We use

$$\boldsymbol{C}_{PID}(s) = \boldsymbol{T}^\dagger \begin{pmatrix} c_x(s) & & \\ & c_y(s) & \\ & & c_\psi(s) \end{pmatrix} \tag{64}$$

with

$$c_j(s) = K_{p,j} \frac{1 + T_{i,j}s}{T_{i,j}s} \frac{1 + T_{d,j}s}{1 + \alpha_j T_{d,j}s}$$

Numerical values are given in table 4.

Table 4: *Numerical values in PID controller*

j	$K_{p,j}$	$T_{i,j}$	$T_{d,j}$	α_j
x	$-4.74 \cdot 10^{-2}$	50	33.33	0.1
y	$-2.53 \cdot 10^{-1}$	50	33.33	0.1
ψ	-2	50	33.33	0.1

5.1.2 Simulations

Two different simulations were performed:

- Smooth step in reference signals
- Step in water current

Results of these simulations are shown in figures 18 and 19.

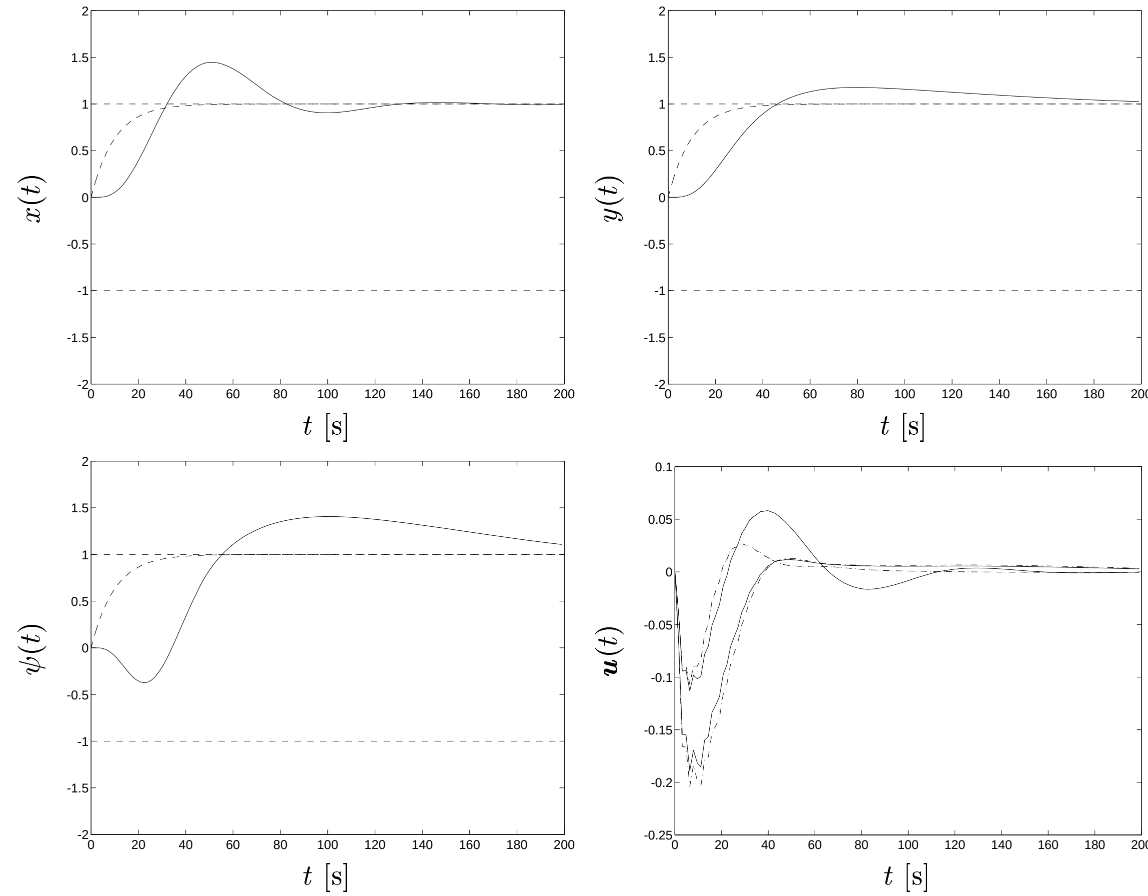


Figure 18: *PID controller: Step in reference signals*

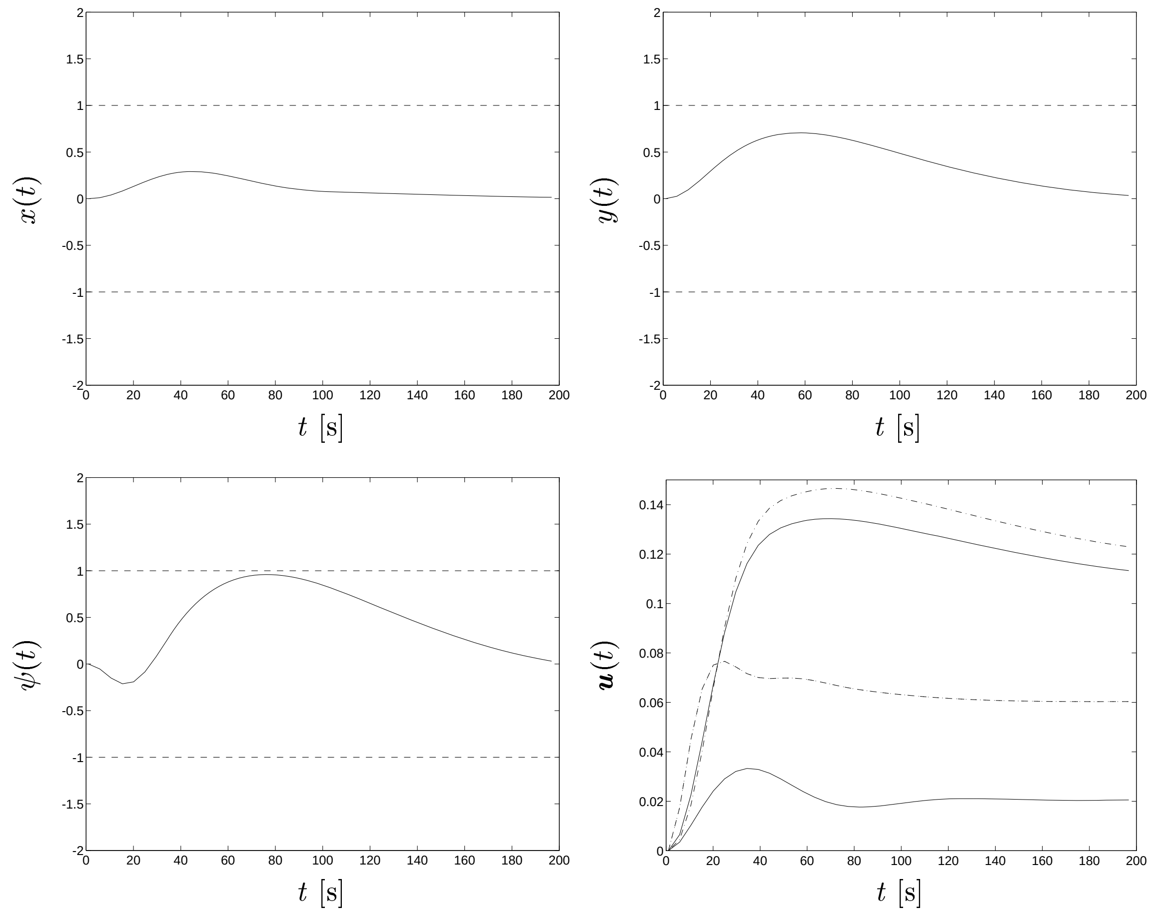


Figure 19: *PID controller: Step in disturbance*

5.2 H_2 controller

5.2.1 Design

The control structure is shown in figure 20.

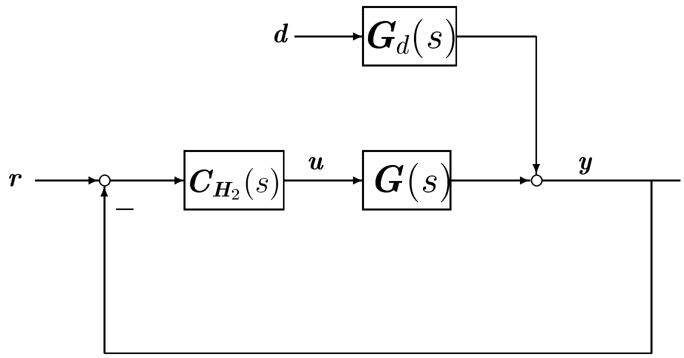


Figure 20: *H_2 controller*

Using the routine `h2syn` in Matlab, an H_2-optimal controller was designed. A balanced realization of this controller has 15 states.

We used weights as shown in figure 21.

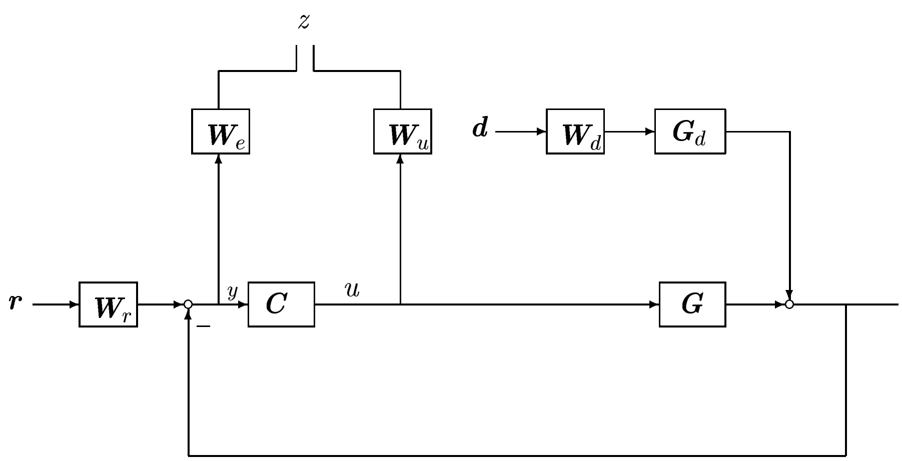


Figure 21: *Weights in H_2 design*

The values of these weights were selected identical to the weights of the μ analysis.

However, instead of weighting input uncertainty, we imposed a weight on control action:

$$W_u(s) = \frac{s}{1 + 10^{-3}s} I_{5\times 5} \tag{65}$$

A plot of this weight is shown in figure 22.

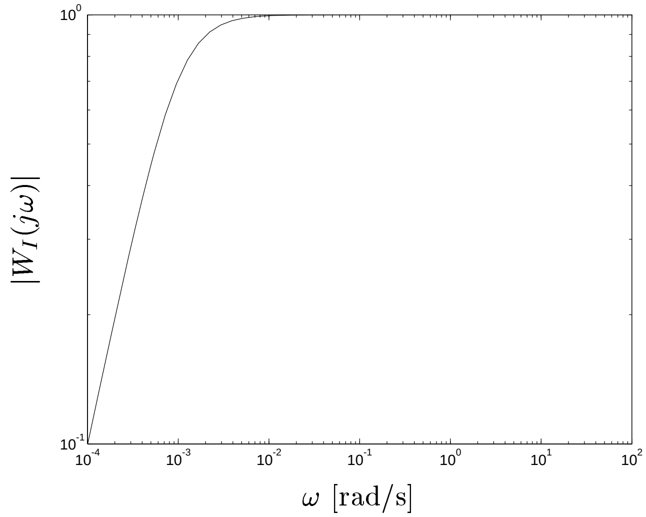


Figure 22: *Weight on control action*

5.2.2 Simulations

Two different simulations were performed:

- Smooth step in reference signals
- Step in water current

Results of these simulations are shown in figures 23 and 24.

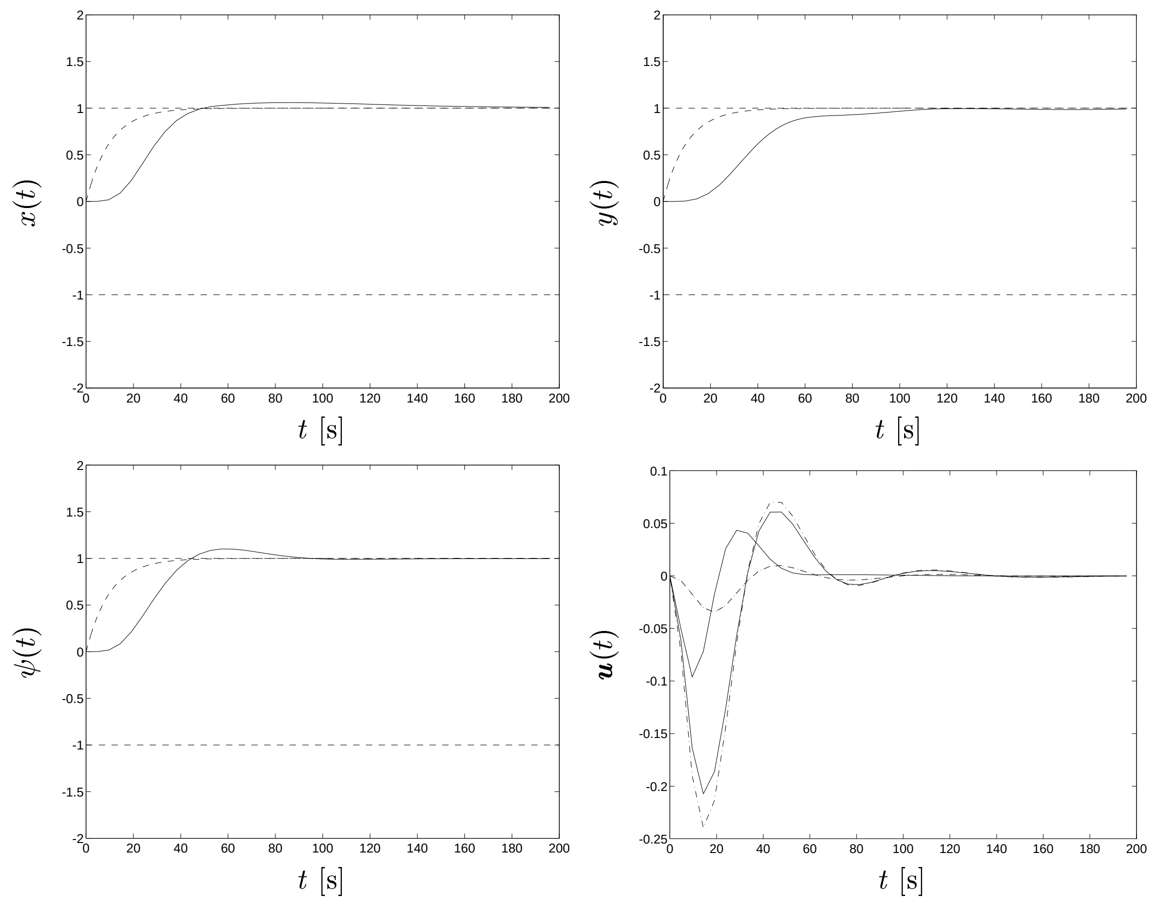


Figure 23: $\boldsymbol{H}_2$ *controller: Step in reference signals*

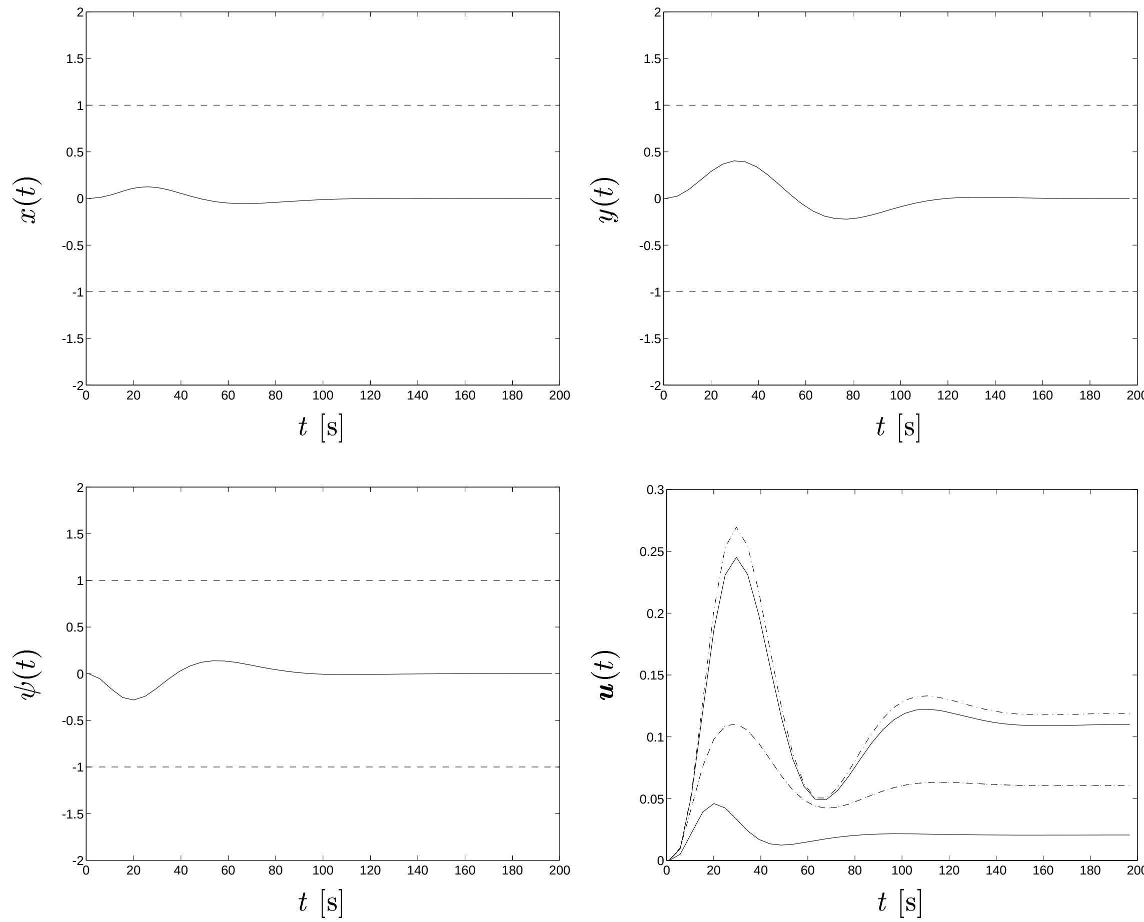


Figure 24: $\boldsymbol{H}_2$ *controller: Step in disturbance*

5.3 Results from μ analysis

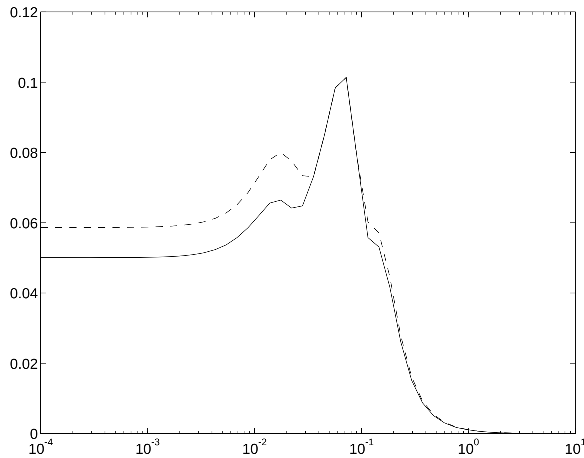
5.3.1 PID

The following results were obtained (table 5):

Table 5: *μ for PID controller*

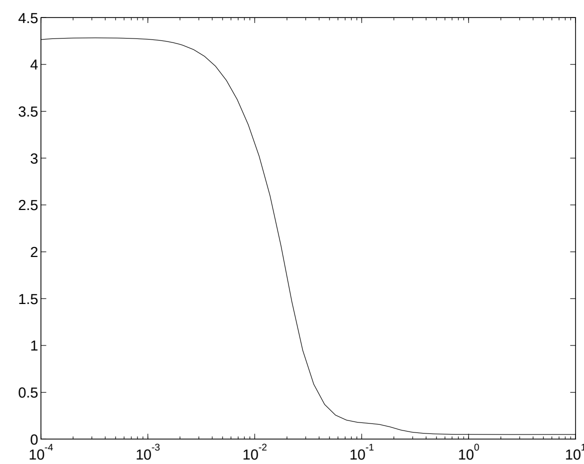
RS	NP	RP
0.10	4.28	> 1

Figure 25 to 27 shows the result of the μ analysis as a function of frequency.



ω [rad/s]

Figure 25: *$\mu(M)$ (solid) and $\overline{\sigma}(M)$ (dashed) — robust stability of PID*

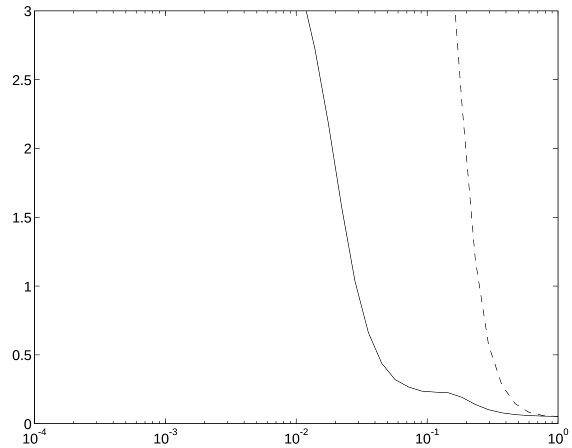


ω [rad/s]

Figure 26: *$\overline{\sigma}(N_{22})$ — nominal performance of PID*

5.3.2 H_2

The following results were obtained (table 6):



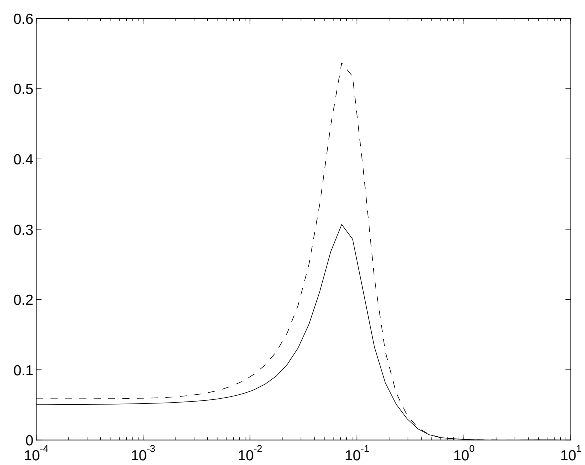
ω [rad/s]

Figure 27: *$\mu(N)$ (solid) and $\overline{\sigma}(N)$ (dashed) — robust performance of PID*

Table 6: *μ for $\boldsymbol{H}_2$ controller*

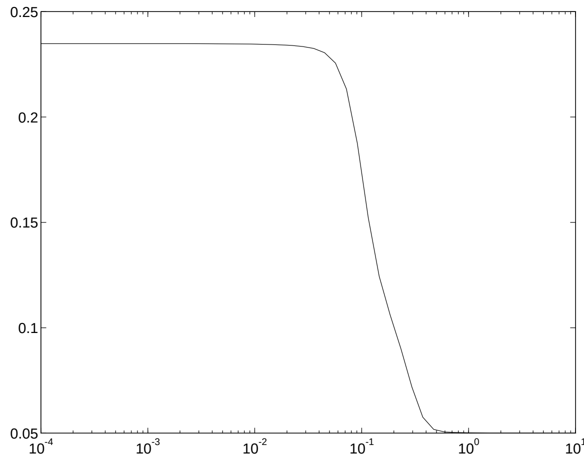
RS	NP	RP
0.31	0.23	> 1

Figure 28 to 30 shows the result of the μ analysis as a function of frequency.



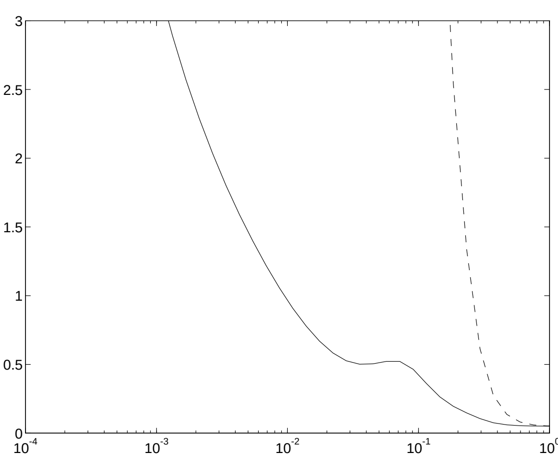
ω [rad/s]

Figure 28: *$\mu(M)$ (solid) and $\overline{\sigma}(M)$ (dashed) — robust stability of $\boldsymbol{H}_2$*



ω [rad/s]

Figure 29: $\overline{\sigma}(N_{22})$ — *nominal performance of* $\boldsymbol{H}_2$



ω [rad/s]

Figure 30: $\mu(N)$ *(solid) and* $\overline{\sigma}(N)$ *(dashed) — robust performance of* $\boldsymbol{H}_2$

6 $\boldsymbol{H}_\infty$ controller design

6.1 Design

The control structure is shown in figure 31.

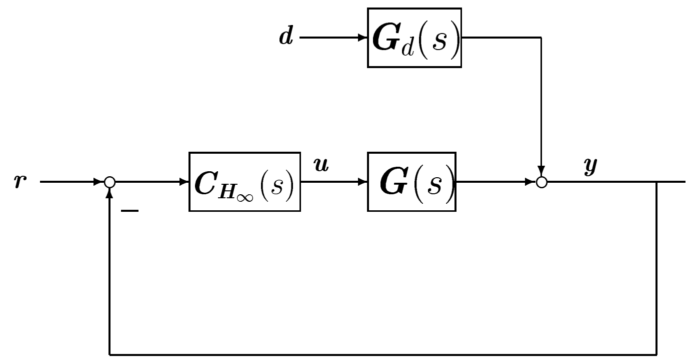


Figure 31: $\boldsymbol{H}_\infty$ *controller*

Identical weights to the $\boldsymbol{H}_2$ case were used, and the output from the algorithm is shown in table 7. A balanced realization of the resulting controller has 17 states.

Table 7: *Results from* $\boldsymbol{H}_\infty$ *controller design*

```
Test bounds:      0.2000 <  gamma  <=     10.0000
```

gamma	hamx_eig	xinf_eig	hamy_eig	yinf_eig	nrho_xy	p/f
10.000	1.0e-05	-1.5e-16	1.0e-05	-5.2e-18	0.0005	p
5.100	1.0e-05	-1.3e-12	1.0e-05	-6.8e-16	0.0019	p
2.650	1.0e-05	-7.1e-13	1.0e-05	0.0e+00	0.0072	p
1.425	1.0e-05	-1.2e-15	1.0e-05	-1.7e-13	0.0250	p
0.812	1.0e-05	-2.0e-13	1.0e-05	-6.0e-15	0.0785	p
0.506	1.0e-05	-4.6e-15	1.0e-05	-2.0e-13	0.2122	p
0.353	1.0e-05	-4.1e-13	1.0e-05	-4.7e-17	0.4779	p
0.277	1.0e-05	-4.1e-14	1.0e-05	-4.2e-15	0.8901	p
0.238	1.0e-05	1.3e-16	1.0e-05	-7.3e-17	1.3889#	f

```
 Gamma value achieved:     0.2766
```

6.2 Simulations

Two different simulations were performed:

- Smooth step in reference signals
- Step in water current

Results of these simulations are shown in figures 32 and 33.

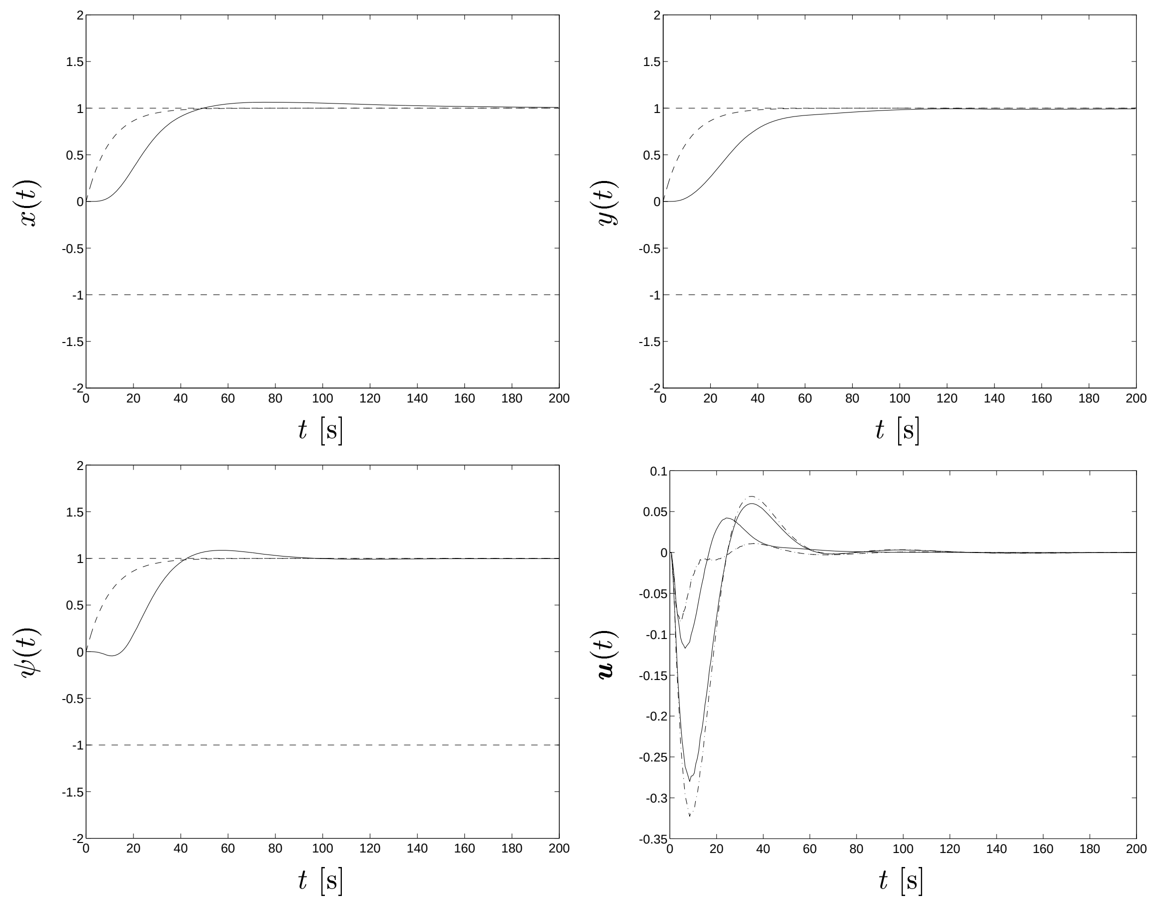


Figure 32: $\boldsymbol{H}_\infty$ *controller: Step in reference signals*

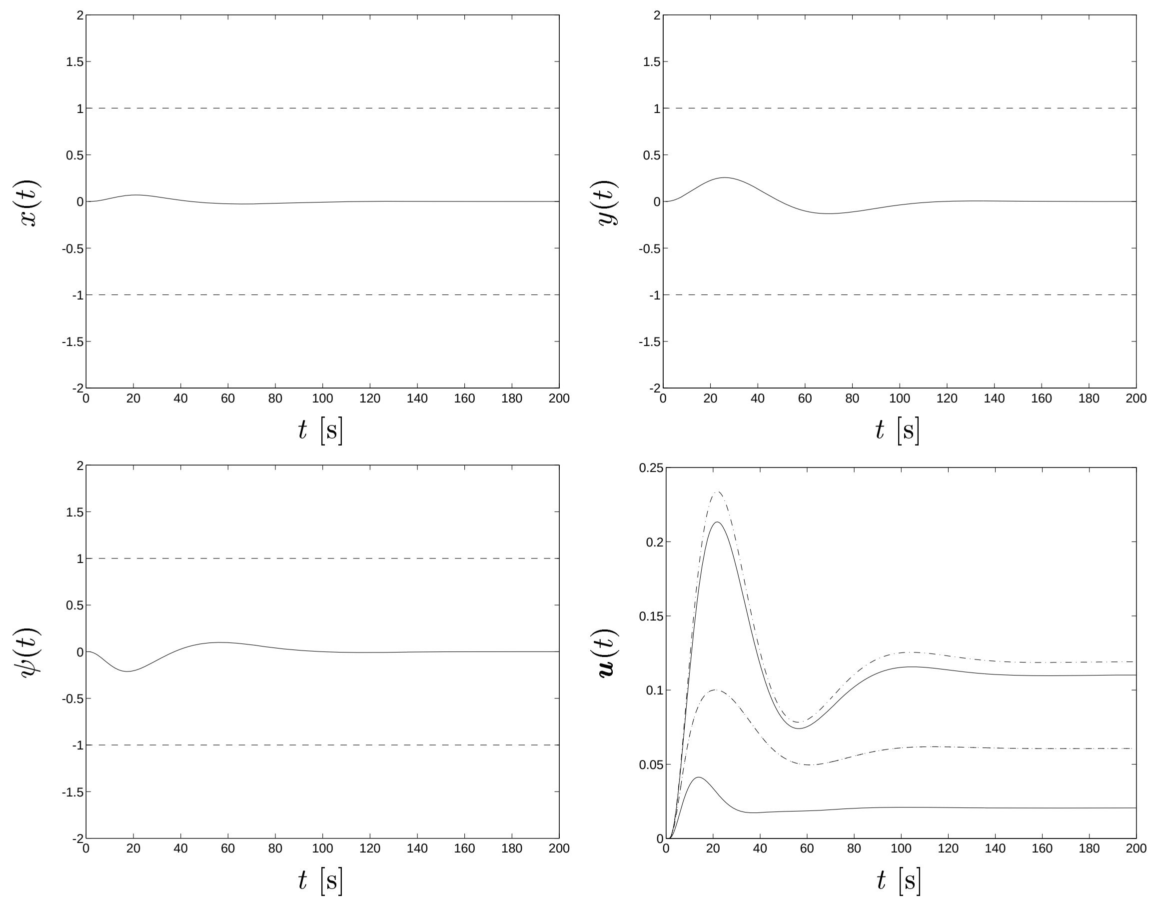


Figure 33: $\boldsymbol{H}_\infty$ *controller: Step in disturbance*

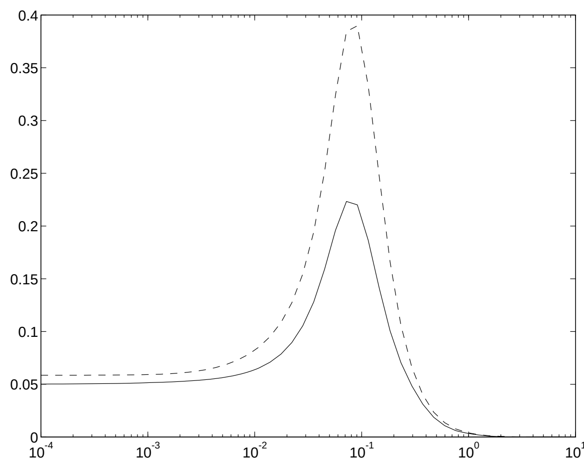
6.3 μ for $\boldsymbol{H}_\infty$ controller

The weights of chapter 4 were used, and the following results obtained (table 8):

Table 8: *μ for $\boldsymbol{H}_\infty$ controller*

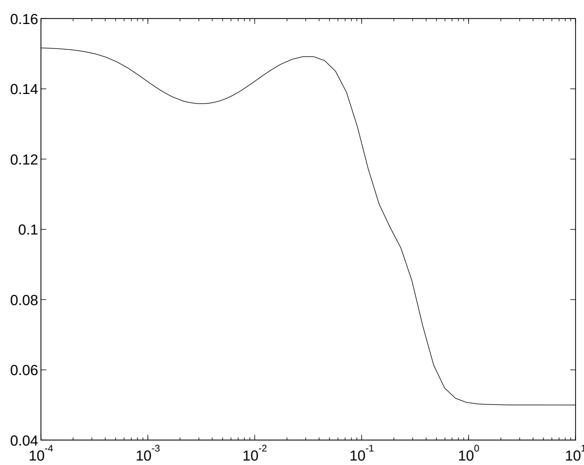
RS	NP	RP
0.22	0.15	> 1

Figure 34 to 36 shows the result of the μ analysis as a function of frequency.



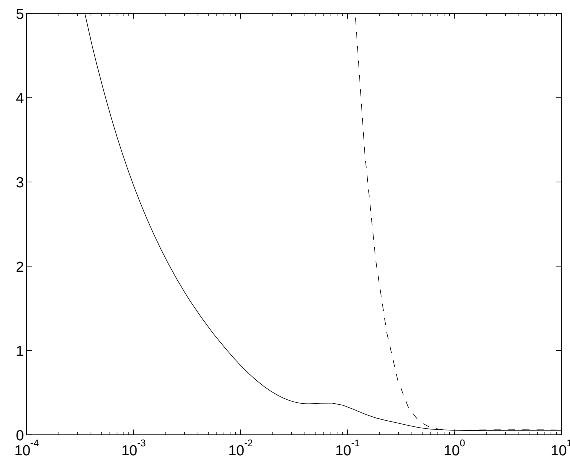
ω [rad/s]

Figure 34: *$\mu(M)$ (solid) and $\overline{\sigma}(M)$ (dashed) — robust stability of $\boldsymbol{H}_\infty$*



ω [rad/s]

Figure 35: *$\overline{\sigma}(N_{22})$ — nominal performance of $\boldsymbol{H}_\infty$*



ω [rad/s]

Figure 36: $\mu(N)$ *(solid) and* $\overline{\sigma}(N)$ *(dashed) — robust performance of* $\boldsymbol{H}_\infty$

7 Discussion and conclusions

More time could have been used to improve the control designs. Some points seem clear anyhow:

- Robust stability is easily obtained, for all three design methods. It is somewhat surprising that this is the case for the PID design as well. However, studying the RGA of $\boldsymbol{G}(s)\boldsymbol{T}^{\dagger}$ we see that there really is little interaction.
- Nominal performance is not obtained for the PID design. We see from time simulations that the response is especially bad with respect to step in the disturbance (water current).
- Robust performance is not obtained in any of the three designs. At first this may seem very depressing, but we should not be surprised. From the plots of $\mu_{\hat{\Delta}}(N)$ as a function of frequency we see that the problems arise at low frequencies. So: We are not guaranteed to obtain the desired steady-state accuracy with a 10% uncertainty in the force coefficients of each thruster.

  A way to go, could be to include uncertainty in the $\boldsymbol{H}_\infty$ control design and eventually develop a μ optimal controller.

A more interesting discussion is the matter of weight selections:

- Performance weight (bandwidth requirement) is easily selected using the singular value plot of $\boldsymbol{G}$ and $\boldsymbol{G}_d$. Desired steady-state accuracy is also selected, in our case somewhat arbitrarily.
- Weight on input uncertainty is selected from common sense. A 10% uncertainty in the force coefficients of the thruster is not unreasonable (at steady state). The uncertainty in these coefficients could be selected as a constant, but in this case it increases to 100% at high frequencies. In this case we saw that the uncertainty at low frequencies cause problems with regards to robust performance.
- In this case the reference input was given little weight compared to the disturbance input (water current). It should be discussed whether this is a reasonable approach. From time simulations of $\boldsymbol{H}_2$ and $\boldsymbol{H}_\infty$ we see that the response to disturbances is very satisfactory. The performance in set point changes is not good at all. One may argue that the control objective is *regulation*, i.e. maintaining a desired set point — not changes.
- The $\boldsymbol{H}_2$ and $\boldsymbol{H}_\infty$ gives comparable results, but $\boldsymbol{H}_\infty$ seems to be slightly better. It has lower values for RS and NP, and time simulations show better rejection of step in disturbance.It is particularly interesting to note this, as we used the same weights in both cases.

  The two approaches to design may seem identical to the user of a Matlab toolbox, but they will give different designs, due to the fact that $\boldsymbol{H}_2$ involves averaging over all frequencies, while $\boldsymbol{H}_\infty$ considers the worst frequency.

Comment to time simulations:

- Saturation in control variables is not experienced in any of the designs.

A few comments on tasks that were not performed:

- Nominal stability of the control system should have been considered. Simulation should not be trusted to indicated unstable poles, especially slow ones.
- An analysis of robust performance with respect to input uncertainty (sensitivity) could have been interesting. We would then have established maximum uncertainty possible as a function of desired steady-state accuracy.
- The design should have taken into account that high frequency thruster usage (due to wave induced motion) should be avoided. In this case we used a very simple method, putting little weight on wave disturbances. More weight should also be put on the control action at high frequencies.
- Analysis of plant and disturbances could have been more detailed. In particular, the closed loop disturbance gain matrix (CLDG) could have been considered.

Concluding remarks:

- Nominal performance and robust stability were easily established with $\boldsymbol{H}_2$ and $\boldsymbol{H}_\infty$ designs. Simulations showed good disturbance rejection.
- Robust performance could not be established. This, however could be obtained with an explicit description of uncertainty in the design phase. Next phase would then be a μ optimal controller.

General comments to the project work:

- Interesting
- Time consuming. Much time was spent on the first few chapters.
- Matlab can sometimes be very trying. Much time was spent figuring out what went wrong and why. The documentation of the μ control toolbox is limited.
- There is definitely a need for design examples in the book.

References

[1] T. I. Fossen, *Guidance and Control of Ocean Vehicles.* John Wiley and Sons Ltd., 1994.

[2] J. G. Balchen, N. A. Jenssen, E. Mathisen, and S. Sælid, "A dynamic positioning system based on Kalman filtering and optimal control," *Modeling, Identification and Control*, vol. 1, pp. 135–163, 1980.

[3] O. M. Faltinsen, *Sea Loads on Ships and Offshore Structures.* Cambridge: Cambridge University Press, 1990.

A Program code

A.1 project.m

```
%
% project.m
%
% Project work 43917 Multivariable frequency analysis
%
% $Id$
%
% This is the main file with menu selections etc.
%


print_to_file = 1;

project_title = 'Project work 43917';
project_items = str2mat( 'Update model', ...
                         'Analysis' );
project_cmds  = str2mat( 'model', ...
                         'analysis');

choices('project_window', project_title, project_items, project_cmds);
```

A.2 model.m

```
%
% model.m
%
% Project work 43917 Multivariable frequency analysis
%
% $Id$
%
% This file defines the vessel model
%

% vessel scaling parameters

L       = 76.2;         % length                  [m]
m       = 4e6;          % mass                    [kg]
g       = 9.81;         % acceleration of gravity [m/s^2]


% from Fossen (1994), in Bis system

T_pp       = [  1.0000       0       0       0       0;
                     0  1.0000  1.0000  1.0000  1.0000;
                     0 -0.3937 -0.3937  0.4514  0.3215; ];

K_pp       = diag([ 0.0260, 0.0140, 0.0140, 0.0015, 0.0053]);

M_pp       = [  1.1274       0       0;
                     0  1.8902 -0.0744;
                     0 -0.0744  0.1278;  ];

A_tilde_pp = [ -0.0367       0       0;
                     0 -0.0933 -0.0073;
                     0  0.1882 -0.3310  ];


D_pp       = -M_pp*A_tilde_pp*1;


% scaling to physical units

T   = diag([ 1 1 L]) * T_pp;
K   = m*g * K_pp;               % muliply by force
M   = m * diag([1 1 L]) * M_pp * diag([1 1 L]);
D   = m/sqrt(L/g) * diag([1 1 L]) * D_pp * diag([1 1 L]);


% low frequency model

epsilon = 1e-10;  % small numerical work-around

A_L = [  -epsilon*eye(3,3),  eye(3,3);
                 zeros(3,3),  -inv(M)*D  ];

B_L = [  zeros(3,3);
         -inv(M)    ];
```

```
% high frequency model

K_wx      = 3;                  % max displacement in x [m]
K_wy      = 2;                  % max displacement in y [m]
K_wpsi    = 4/180*pi;           % max displacement in psi [rad]
omega_0   = 2*pi*0.1;           % dominating wave frequency
zeta      = 0.5;

A_H_tilde = [      0,           1;
              -omega_0^2, -2*zeta*omega_0 ];

A_H       = [ A_H_tilde,  zeros(2,2), zeros(2,2);
              zeros(2,2),  A_H_tilde, zeros(2,2);
              zeros(2,2), zeros(2,2),  A_H_tilde ];

E_H       = [    0     0     0;
               K_wx    0     0;
           0     0     0;
 0   K_wy    0;
 0     0     0;
 0     0   K_wpsi  ];

% thruster model

T_x       = 5;
T_y       = 5;
T_psi     = 5;

A_thr     = diag([-1/T_x, -1/T_y, -1/T_psi]);


% linearized current model

beta_C0   = 45/180*pi;
psi_L0    = 0;

E_C_tilde = [ cos(beta_C0-psi_L0);
              sin(beta_C0-psi_L0);
                     0           ];

E_C       = [ zeros(3,1);
              inv(M)*D*E_C_tilde ];
```

```
% measurement model

C_L       = [ eye(3), zeros(3,3) ];

C_H       = [ 0 1  0 0  0 0;
              0 0  0 1  0 0;
              0 0  0 0  0 1  ];

% total vessel model

A         = [     A_L,     zeros(6,6),     B_L;
              zeros(6,6),     A_H,       zeros(6,3);
      zeros(3,6),  zeros(3,6),    A_thr      ];

B         = [ zeros(6,5);
              zeros(6,5);
              -A_thr*T*K  ];

C         = [ C_L, C_H, zeros(3,3); ];

E         = [    E_C,     zeros(6,3);
              zeros(6,1),     E_H;
      zeros(3,1), zeros(3,3)  ];

% scaling of plant and disturbances

D_d       = diag([ 0.5 1 1 1 ]);
D_u       = diag([ 1 1 1 1 1 ]);
D_e       = diag([ 1/5 1/5 1/(3/180*pi) ]);


% build system matrices (scaled)

%G_xL   = pck(A_L,eye(6),eye(6),zeros(6,6));
%G_tauL = pck(A_thr,-A_thr*T*K,eye(3),zeros(3,5));
%G_s    = mmult(D_e,C_L,G_xL,B_L,G_tauL,D_u);

G_s   = sysbal(pck(A,B*D_u,D_e*C,zeros(3,5)));
Gd_s  = sysbal(pck(A,E*D_d,D_e*C,zeros(3,4)));
```

A.3 analysis.m

```
%
% analysis.m
%
% Project work 43917 Multivariable frequency analysis
%
% $Id$
%
% This file contains frequency analysis of plant,
% disturbance model etc.
%


% frequencies of interest
omega = logspace(-2,1,50);


analysis_title = 'Analysis';
analysis_items = str2mat( 'Poles/zeros etc', 'RGA', 'SVD' );

analysis_cmds  = str2mat( 'analysis_pz', ...
                          'analysis_RGA', ...
                          'analysis_SVD' );

choices('analysis_window', analysis_title, analysis_items, analysis_cmds);



%
% analysis_pz.m
% $Id$
%

for i=1:3,
  for j=1:5,
    disp(['G', num2str(i), num2str(j), ':']);
    [Aij,Bij,Cij,Dij]     = unpck(sel(G_s,i,j));
    [Aijm,Bijm,Cijm,Dijm] = minreal(Aij,Bij,Cij,Dij,1e-5);
    disp('Poles:');
    eig(Aijm)
    disp('Zeros:');
    tzero(Aijm,Bijm,Cijm,Dijm)
    disp('Gain:');
    dc = dcgain(Aijm,Bijm,Cijm,Dijm)
    [mag,phase] = bode(Aijm,Bijm,Cijm,Dijm,1,omega);
    if dc<0 % dc gain will probably be infinite for some elements
      phase = phase-360;
    end;
    clf;
    semilogx(omega,20*log10(mag)),grid;
    axis_scaling_amp;
    label = [ 'G', num2str(i), num2str(j), 'omega_abs' ];
    xlabel('frequency');
    ylabel(label);
    if print_to_file,
      print('-deps', label);
      unix(['ps2frag ', label, '.eps']);
    end;
    disp('Press a key'); pause;
    clf;
    semilogx(omega,phase), grid;
    axis_scaling_phase;
    label = [ 'G', num2str(i), num2str(j), 'omega_phase' ];
    xlabel('frequency');
    ylabel(label);
    if print_to_file
      print('-deps', label);
      unix(['ps2frag ', label, '.eps']);
    end;
    disp('Press a key'); pause;
  end;
end;


for i=1:3,
  for j=1:4,
    disp(['Gd', num2str(i), num2str(j), ':']);
    [Aij,Bij,Cij,Dij]     = unpck(sel(Gd_s,i,j));
    [Aijm,Bijm,Cijm,Dijm] = minreal(Aij,Bij,Cij,Dij,1e-5);
    disp('Poles:');
    eig(Aijm)
    disp('Zeros:');
    tzero(Aijm,Bijm,Cijm,Dijm)
    disp('Gain:');
    dc = dcgain(Aijm,Bijm,Cijm,Dijm)
    [mag,phase] = bode(Aijm,Bijm,Cijm,Dijm,1,omega);
    if dc<0 % dc gain will probably be infinite for some elements
      phase = phase-360;
```

```
    end;
    clf;
    semilogx(omega,20*log10(mag)),grid;
    axis_scaling_amp;
    label = [ 'Gd', num2str(i), num2str(j), 'omega_abs' ];
    xlabel('frequency');
    ylabel(label);
    if print_to_file,
      print('-deps', label);
      unix(['ps2frag ', label, '.eps']);
    end;
    disp('Press a key'); pause;
    clf;
    semilogx(omega,phase), grid;
    axis_scaling_phase;
    label = [ 'Gd', num2str(i), num2str(j), 'omega_phase' ];
    xlabel('frequency');
    ylabel(label);
    if print_to_file
      print('-deps', label);
      unix(['ps2frag ', label, '.eps']);
    end;
    disp('Press a key'); pause;
  end;
end;
```

```
%
% analysis_RGA.m
% $Id$
%

% use pinv(T) as thruster allocation


G_omega      = frsp(mmult(G_s,pinv(T)),  omega);    % plant

G_omega_pinv = vinv(G_omega);

RGA          = veval('.*',G_omega,transp(G_omega_pinv));

subplot(3,3,1);
for i=1:3,
  for j=1:3,
    subplot(3,3,(i-1)*3+j);
    vplot('liv,m',sel(RGA,i,j)),grid;
    axis([1e-3 1e1 -2 2]);
    ylabel(['lambda',num2str(i),num2str(j)]);
  end;
end;

print('-deps','RGA');
unix('ps2frag RGA.eps');
```

```
%
% analysis_SVD.m
% $Id$
%

G_omega  = frsp(G_s,  omega);    % plant
Gd_omega = frsp(Gd_s, omega);    % disturbances

[U,  S,  V]  = vsvd(G_omega);
[Ud, Sd, Vd] = vsvd(Gd_omega);

clf;
vplot('liv,d',dB(S)),grid;
ylabel('plant');
xlabel('frequency');
print('-deps','SVD_G_omega');
unix('ps2frag SVD_G_omega.eps');

clf;
vplot('liv,d',dB(Sd)),grid;
ylabel('disturbance');
xlabel('frequency');
print('-deps','SVD_Gd_omega');
unix('ps2frag SVD_Gd_omega.eps');
```

A.4 Control design

```
%
% controller_pid.m
% $Id$
%


alpha_x = 0.1;
Kp_x    = -15*3.16e-3;
Ti_x    = 1/0.02;
Td_x    = 1/0.03;

alpha_y = 0.1;
Kp_y    = -8*3.16e-2;
Ti_y    = 1/0.02;
Td_y    = 1/0.03;

alpha_psi = 0.1;
Kp_psi  = -2;
Ti_psi  = 1/0.02;
Td_psi  = 1/0.03;

c_x   = nd2sys(conv([Ti_x 1],[Td_x 1]), ...
               conv([Ti_x 0],[alpha_x*Td_x 1]), Kp_x);

c_y   = nd2sys(conv([Ti_y 1],[Td_y 1]), ...
               conv([Ti_y 0],[alpha_y*Td_y 1]), Kp_y);

c_psi = nd2sys(conv([Ti_psi 1],[Td_psi 1]), ...
               conv([Ti_psi 0],[alpha_psi*Td_psi 1]), Kp_psi);


%zeta_t      = 0.2;
%zeta_n      = 1;
%c_notch     = nd2sys([1/omega_0^2,2*zeta_t/omega_0,1],...
%                 [1/omega_0^2,2*zeta_n/omega_0,1]);

%C_notch_s   = daug(c_notch,c_notch,c_notch);

%G_001 = var2con(vebe('abs',frsp(mmult(G_s,pinv(T)),0.01)));
%C_s   = mmult(pinv(T), pinv(G_001), daug(c_x, c_y, c_psi));
C_s   = mmult(pinv(T), daug(c_x, c_y, c_psi));
disp('Press a key'); pause;

systemnames   = 'G_s Gd_s C_s';
inputvar      = '[r(3); d(4)]';
outputvar     = '[G_s+Gd_s; C_s]';
input_to_C_s  = '[r-G_s-Gd_s]';
input_to_G_s  = '[C_s]';
input_to_Gd_s = '[d]';

sysoutname   = 'T_s';
cleanupsysic = 'yes';
sysic;

tfinal = 200;
t      = 0:tfinal;
tau    = 10;
r      = vpck([(1-exp(-t/tau))',(1-exp(-t/tau))',(1-exp(-t/tau))'],t);
d      = vpck([zeros(tfinal+1,1),zeros(tfinal+1,3)],t);
y      = vdcmate(trsp(T_s,transp(sbs(r,d)),tfinal));

clf;

vplot(sel(y,1,1),sel(r,1,1)),hold;
plot([0 tfinal],[1 1],'--'); plot([0 tfinal],[-1 -1],'--');
axis([0 tfinal -2 2]),hold off;
xlabel('time'); ylabel('x');
if print_to_file,
  print('-deps','PIDref_x'); unix('ps2frag PIDref_x.eps');
end;
disp('Press a key'); pause;

vplot(sel(y,2,1),sel(r,1,2)),hold;
plot([0 tfinal],[1 1],'--'); plot([0 tfinal],[-1 -1],'--');
axis([0 tfinal -2 2]),hold off;
xlabel('time'); ylabel('y');
if print_to_file,
  print('-deps','PIDref_y'); unix('ps2frag PIDref_y.eps');
end;
disp('Press a key'); pause;

vplot(sel(y,3,1),sel(r,1,3)),hold;
plot([0 tfinal],[1 1],'--'); plot([0 tfinal],[-1 -1],'--');
axis([0 tfinal -2 2]),hold off;
xlabel('time'); ylabel('psi');
```

```
if print_to_file,
  print('-deps','PIDref_psi'); unix('ps2frag PIDref_psi.eps');
end;
disp('Press a key'); pause;


vplot(sel(y,4:8,1)),hold;
xlabel('time'); ylabel('controller');
if print_to_file,
  print('-deps','PIDref_u'); unix('ps2frag PIDref_u.eps');
end;

disp('Press a key'); pause;


r      = vpck([zeros(tfinal,1),zeros(tfinal,1),zeros(tfinal,1)],1:tfinal);
d      = vpck([ones(tfinal,1),zeros(tfinal,3)],1:tfinal);
y      = vdcmate(trsp(T_s,transp(sbs(r,d)),tfinal),30);

clf;

vplot(sel(y,1,1)),hold;
plot([0 tfinal],[1 1],'--'); plot([0 tfinal],[-1 -1],'--');
axis([0 tfinal -2 2]),hold off;
xlabel('time'); ylabel('x');
if print_to_file,
  print('-deps','PIDcurr_x'); unix('ps2frag PIDcurr_x.eps');
end;
disp('Press a key'); pause;

vplot(sel(y,2,1)),hold;
plot([0 tfinal],[1 1],'--'); plot([0 tfinal],[-1 -1],'--');
axis([0 tfinal -2 2]),hold off;
xlabel('time'); ylabel('y');
if print_to_file,
  print('-deps','PIDcurr_y'); unix('ps2frag PIDcurr_y.eps');
end;
disp('Press a key'); pause;

vplot(sel(y,3,1)),hold;
plot([0 tfinal],[1 1],'--'); plot([0 tfinal],[-1 -1],'--');
axis([0 tfinal -2 2]),hold off;
xlabel('time'); ylabel('psi');
if print_to_file,
  print('-deps','PIDcurr_psi'); unix('ps2frag PIDcurr_psi.eps');
end;
disp('Press a key'); pause;

vplot(sel(y,4:8,1)),hold;
xlabel('time'); ylabel('controller');
if print_to_file,
  print('-deps','PIDcurr_u'); unix('ps2frag PIDcurr_u.eps');
end;
```

```
%
% controller_cloop.m
% $Id$
%

GC_s = mmult(G_s,C_s);

omega   = logspace(-2,0,50);

clf;
for i=1:3,
    disp(['CLOOP', num2str(i), num2str(i), ':']);
    [Aii,Bii,Cii,Dii]     = unpck(sel(GC_s,i,i));
    [Aiim,Biim,Ciim,Diim] = minreal(Aii,Bii,Cii,Dii,1e-5);
    disp('Poles:');
    eig(Aiim)
    disp('Zeros:');
    tzero(Aiim,Biim,Ciim,Diim)
    disp('Gain:');
    [mag,phase] = bode(Aiim,Biim,Ciim,Diim,1,omega);
    subplot(211);
    semilogx(omega,20*log10(mag)),grid;
    subplot(212);
    semilogx(omega,phase), grid;
    disp('Press a key'); pause;
end;
```

```
%
% controller_H2
%
% $Id$
```

```
%

% now, specify interconnections, using sysic
%

% first, specify weights
%

% weight on the disturbance
wdC    = nd2sys(1, [1 1e-2], 1e-2);
Wd_s   = daug( wdC, 1e-4, 1e-4, 1e-4 );

% weight on the reference signal
Wr_s   = 0.1*eye(3);

% weight on the control error
we_x    = nd2sys([1 8e-2],[1 1e-5], 1/1.1);
we_y    = we_x; we_psi = we_x;
%we_x   = nd2sys([1 1e-1],[1 1e-5], 1/2);
%we_y   = nd2sys([1 1e-1],[1 1e-5], 1/2);
%we_psi = nd2sys([1 1e-1],[1 1e-5], 1/2);
We_s    = sysbal(daug( we_x, we_y, we_psi ));

% weight on the control
wu     = nd2sys([1 0],[1 1e-3], 1);
Wu_s   = sysbal( daug(wu, wu, wu, wu, wu) );

% now, specify interconnections using sysic

C3_s = pck([],[],[],pinv(T));

systemnames      = 'G_s C3_s Gd_s We_s Wd_s Wu_s Wr_s';
inputvar         = '[r(3); d(4); u_p(3)]';
outputvar        = '[Wu_s; We_s; Wr_s-G_s-Gd_s]';
input_to_G_s     = '[C3_s]';
input_to_C3_s    = '[u_p]';
input_to_Wu_s    = '[C3_s]';
input_to_Wd_s    = '[d]';
input_to_Gd_s    = '[Wd_s]';
input_to_Wr_s    = '[r]';
input_to_We_s    = '[Wr_s-G_s-Gd_s]';

sysoutname       = 'P_s';
cleanupsysic     = 'yes';
```

```
sysic;

[AP,BP,CP,DP ] = unpck(P_s);
DP(1:8,1:7)    = zeros(8,7);       % small work-around
                                   % see mu-toolbox manual (h2syn)
P_s            = pck(AP,BP,CP,DP);

[K_s, N_s] = h2syn(P_s,3,3,-1);

systemnames   = 'G_s Gd_s K_s C3_s';
inputvar      = '[r(3); d(4)]';
outputvar     = '[G_s+Gd_s; C3_s]';
input_to_G_s  = '[C3_s]';
input_to_Gd_s = '[d]';
input_to_K_s  = '[r-G_s-Gd_s]';
input_to_C3_s = '[K_s]';

sysoutname    = '[T_s]';
cleanupsysic  = 'yes';
sysic;



tfinal = 200;
t      = 0:tfinal;
tau    = 10;
r      = vpck([(1-exp(-t/tau))',(1-exp(-t/tau))',(1-exp(-t/tau))'],t);
d      = vpck([zeros(tfinal+1,1),zeros(tfinal+1,3)],t);
y      = vdcmate(trsp(T_s,transp(sbs(r,d)),tfinal),30);

clf;

vplot(sel(y,1,1),sel(r,1,1)),hold;
plot([0 tfinal],[1 1],'--'); plot([0 tfinal],[-1 -1],'--');
axis([0 tfinal -2 2]),hold off;
xlabel('time'); ylabel('x');
if print_to_file,
  print('-deps','H2ref_x'); unix('ps2frag H2ref_x.eps');
end;
disp('Press a key'); pause;

vplot(sel(y,2,1),sel(r,1,2)),hold;
plot([0 tfinal],[1 1],'--'); plot([0 tfinal],[-1 -1],'--');
axis([0 tfinal -2 2]),hold off;
```

```
xlabel('time'); ylabel('y');
if print_to_file,
  print('-deps','H2ref_y'); unix('ps2frag H2ref_y.eps');
end;
disp('Press a key'); pause;

vplot(sel(y,3,1),sel(r,1,3)),hold;
plot([0 tfinal],[1 1],'--'); plot([0 tfinal],[-1 -1],'--');
axis([0 tfinal -2 2]),hold off;
xlabel('time'); ylabel('psi');
if print_to_file,
  print('-deps','H2ref_psi'); unix('ps2frag H2ref_psi.eps');
end;
disp('Press a key'); pause;

vplot(sel(y,4:8,1)),hold;
xlabel('time'); ylabel('controller');
if print_to_file,
  print('-deps','H2ref_u'); unix('ps2frag H2ref_u.eps');
end;

disp('Press a key'); pause;

r      = vpck([zeros(tfinal,1),zeros(tfinal,1),zeros(tfinal,1)],1:tfinal);
d      = vpck([ones(tfinal,1),zeros(tfinal,3)],1:tfinal);
y      = vdcmate(trsp(T_s,transp(sbs(r,d)),tfinal),30);

clf;

vplot(sel(y,1,1)),hold;
plot([0 tfinal],[1 1],'--'); plot([0 tfinal],[-1 -1],'--');
axis([0 tfinal -2 2]),hold off;
xlabel('time'); ylabel('x');
if print_to_file,
  print('-deps','H2curr_x'); unix('ps2frag H2curr_x.eps');
end;
disp('Press a key'); pause;

vplot(sel(y,2,1)),hold;
plot([0 tfinal],[1 1],'--'); plot([0 tfinal],[-1 -1],'--');
axis([0 tfinal -2 2]),hold off;
xlabel('time'); ylabel('y');
if print_to_file,
  print('-deps','H2curr_y'); unix('ps2frag H2curr_y.eps');
end;
disp('Press a key'); pause;

vplot(sel(y,3,1)),hold;
plot([0 tfinal],[1 1],'--'); plot([0 tfinal],[-1 -1],'--');
axis([0 tfinal -2 2]),hold off;
xlabel('time'); ylabel('psi');
if print_to_file,
  print('-deps','H2curr_psi'); unix('ps2frag H2curr_psi.eps');
end;
disp('Press a key'); pause;

vplot(sel(y,4:8,1)),hold;
xlabel('time'); ylabel('controller');
if print_to_file,
  print('-deps','H2curr_u'); unix('ps2frag H2curr_u.eps');
end;

%
% controller_H_inf
%
% $Id$
%

% first, specify weights
%

% weight on the disturbance
wdC    = nd2sys(1, [1 1e-2], 1e-2);
Wd_s   = daug( wdC, 1e-4, 1e-4, 1e-4 );

% weight on the reference signal
Wr_s   = 0.1*eye(3);

% weight on the control error
we_x    = nd2sys([1 8e-2],[1 1e-5], 1/1.1);
we_y    = we_x; we_psi = we_x;
%we_x   = nd2sys([1 1e-1],[1 1e-5], 1/2);
%we_y   = nd2sys([1 1e-1],[1 1e-5], 1/2);
%we_psi = nd2sys([1 1e-1],[1 1e-5], 1/2);
We_s    = sysbal(daug( we_x, we_y, we_psi ));

% weight on the control
wu     = nd2sys([1 0],[1 1e-3], 1);
Wu_s   = sysbal( daug(wu, wu, wu, wu, wu) );
```

```
% now, specify interconnections, using sysic
%

C3_s = pck([],[],[],pinv(T));

systemnames      = 'G_s C3_s Gd_s We_s Wd_s Wu_s Wr_s';
inputvar         = '[r(3); d(4); u_p(3)]';
outputvar        = '[Wu_s; We_s; Wr_s-G_s-Gd_s]';
input_to_G_s     = '[C3_s]';
input_to_C3_s    = '[u_p]';
input_to_Wu_s    = '[C3_s]';
input_to_Wd_s    = '[d]';
input_to_Gd_s    = '[Wd_s]';
input_to_Wr_s    = '[r]';
input_to_We_s    = '[Wr_s-G_s-Gd_s]';

sysoutname       = 'P_s';
cleanupsysic     = 'yes';
sysic;

[K_s, N_s, gfin] = hinfsyn(P_s,3,3,0.2,10,0.05,-1);

disp('Press a key');
pause;

systemnames   = 'G_s Gd_s K_s C3_s';
inputvar      = '[r(3); d(4)]';
outputvar     = '[G_s+Gd_s; C3_s]';
input_to_G_s  = '[C3_s]';
input_to_Gd_s = '[d]';
input_to_K_s  = '[r-G_s-Gd_s]';
input_to_C3_s = '[K_s]';

sysoutname    = '[T_s]';
cleanupsysic  = 'yes';
sysic;

tfinal = 200;
t      = 0:tfinal;
tau    = 10;
r      = vpck([(1-exp(-t/tau))',(1-exp(-t/tau))',(1-exp(-t/tau))'],t);
d      = vpck([zeros(tfinal+1,1),zeros(tfinal+1,3)],t);
y      = vdcmate(trsp(T_s,transp(sbs(r,d)),tfinal));
```

```
clf;

vplot(sel(y,1,1),sel(r,1,1)),hold;
plot([0 tfinal],[1 1],'--'); plot([0 tfinal],[-1 -1],'--');
axis([0 tfinal -2 2]),hold off;
xlabel('time'); ylabel('x');
if print_to_file,
  print('-deps','Hinfref_x'); unix('ps2frag Hinfref_x.eps');
end;
disp('Press a key'); pause;

vplot(sel(y,2,1),sel(r,1,2)),hold;
plot([0 tfinal],[1 1],'--'); plot([0 tfinal],[-1 -1],'--');
axis([0 tfinal -2 2]),hold off;
xlabel('time'); ylabel('y');
if print_to_file,
  print('-deps','Hinfref_y'); unix('ps2frag Hinfref_y.eps');
end;
disp('Press a key'); pause;

vplot(sel(y,3,1),sel(r,1,3)),hold;
plot([0 tfinal],[1 1],'--'); plot([0 tfinal],[-1 -1],'--');
axis([0 tfinal -2 2]),hold off;
xlabel('time'); ylabel('psi');
if print_to_file,
  print('-deps','Hinfref_psi'); unix('ps2frag Hinfref_psi.eps');
end;
disp('Press a key'); pause;

vplot(sel(y,4:8,1)),hold;
xlabel('time'); ylabel('controller');
if print_to_file,
  print('-deps','Hinfref_u'); unix('ps2frag Hinfref_u.eps');
end;

disp('Press a key'); pause;

r      = vpck([zeros(tfinal,1),zeros(tfinal,1),zeros(tfinal,1)],1:tfinal);
d      = vpck([ones(tfinal,1),zeros(tfinal,3)],1:tfinal);
y      = vdcmate(trsp(T_s,transp(sbs(r,d)),tfinal));

clf;
```

```
vplot(sel(y,1,1)),hold;
plot([0 tfinal],[1 1],'--'); plot([0 tfinal],[-1 -1],'--');
axis([0 tfinal -2 2]),hold off;
xlabel('time'); ylabel('x');
if print_to_file,
  print('-deps','Hinfcurr_x'); unix('ps2frag Hinfcurr_x.eps');
end;
disp('Press a key'); pause;

vplot(sel(y,2,1)),hold;
plot([0 tfinal],[1 1],'--'); plot([0 tfinal],[-1 -1],'--');
axis([0 tfinal -2 2]),hold off;
xlabel('time'); ylabel('y');
if print_to_file,
  print('-deps','Hinfcurr_y'); unix('ps2frag Hinfcurr_y.eps');
end;
disp('Press a key'); pause;

vplot(sel(y,3,1)),hold;
plot([0 tfinal],[1 1],'--'); plot([0 tfinal],[-1 -1],'--');
axis([0 tfinal -2 2]),hold off;
xlabel('time'); ylabel('psi');
if print_to_file,
  print('-deps','Hinfcurr_psi'); unix('ps2frag Hinfcurr_psi.eps');
end;
disp('Press a key'); pause;

vplot(sel(y,4:8,1)),hold;
xlabel('time'); ylabel('controller');
if print_to_file,
  print('-deps','Hinfcurr_u'); unix('ps2frag Hinfcurr_u.eps');
end;
```

A.5 μ analysis

```
%
%  mu_PID
%
%  mu-analysis for PID controller
%
%  $Id$
%

% first, specify weights
%

% weight on the disturbance
wdC    = nd2sys(1, [1 1e-2], 1e-2);
Wd_s   = daug( wdC, 1e-4, 1e-4, 1e-4 );

% weight on the control error
we_x    = nd2sys([1 8e-2],[1 1e-5], 1/2);
we_y    = we_x; we_psi = we_x;
We_s   = sysbal(daug( we_x, we_y, we_psi ));

% weight on the reference
Wr_s    = 0.1*eye(3);

% weight on input uncertainty
wI     = nd2sys([1 5e-1],[1 1e1],1);
WI_s   = daug( wI, wI, wI, wI, wI );


systemnames      = 'G_s Gd_s We_s Wd_s WI_s Wr_s C_s';
inputvar         = '[u_Delta(5); r(3); d(4)]';
outputvar        = '[WI_s; We_s]';
input_to_G_s     = '[C_s+u_Delta]';
input_to_C_s     = '[Wr_s-G_s-Gd_s]';
input_to_WI_s    = '[C_s]';
input_to_Wd_s    = '[d]';
input_to_Gd_s    = '[Wd_s]';
input_to_Wr_s    = '[r]';
input_to_We_s    = '[Wr_s-G_s-Gd_s]';

sysoutname       = 'N_s';
cleanupsysic     = 'yes';
sysic;

omega      = logspace(-4,1,50);

N_omega    = frsp(N_s,omega);

M_omega    = sel(N_omega,1:5,1:5);
N22_omega  = sel(N_omega,6:8,6:12);
```

```
% input uncertainty
Delta      = [1 1; 1 1; 1 1; 1 1; 1 1];
% fictitious performance uncertainty
Delta_p    = [7 3];

% robust stability
[mu_RS, dvec, sens, pvec] = mu(M_omega,Delta);
clf; vplot('liv,m',sel(mu_RS,1,1),vnorm(M_omega));
xlabel('frequency');
if print_to_file,
  print('-deps','muPIDRS'); unix('ps2frag muPIDRS.eps');
end;

disp(['mu for RS:  ',num2str(pkvnorm(sel(mu_RS,1,1)))]);
disp('Press a key'); pause;

% nominal performance - full matrix/no need to calculate mu

clf; vplot('liv,m',vnorm(N22_omega));
xlabel('frequency');
if print_to_file,
  print('-deps','muPIDNP'); unix('ps2frag muPIDNP.eps');
end;

disp(['mu for NP:  ',num2str(max(vunpck(vnorm(N22_omega))))]);
disp('Press a key'); pause;

% robust performance

[mu_RP, dvec, sens, pvec] = mu(N_omega,[Delta; Delta_p]);
clf; vplot('liv,m',sel(mu_RP,1,1),vnorm(N_omega));
axis([1e-4 1 0 3])
xlabel('frequency');
if print_to_file,
  print('-deps','muPIDRP'); unix('ps2frag muPIDRP.eps');
end;

disp(['mu for RP:  ',num2str(pkvnorm(sel(mu_RP,1,1)))]);



%
%  mu_H2
%
%  mu-analysis for H2 controller
```

```
%
%  $Id$
%

% first, specify weights
%


% weight on the disturbance
wdC    = nd2sys(1, [1 1e-2], 1e-2);
Wd_s   = daug( wdC, 1e-4, 1e-4, 1e-4 );

% weight on the control error
we_x    = nd2sys([1 8e-2],[1 1e-5], 1/2);
we_y    = we_x; we_psi = we_x;
We_s   = sysbal(daug( we_x, we_y, we_psi ));

% weight on the reference
Wr_s    = 0.1*eye(3);

% weight on input uncertainty
wI     = nd2sys([1 5e-1],[1 1e1],1);
WI_s   = daug( wI, wI, wI, wI, wI );

C3_s   = pck([],[],[],pinv(T));

systemnames      = 'G_s Gd_s We_s Wd_s WI_s Wr_s C3_s K_s';
inputvar         = '[u_Delta(5); r(3); d(4)]';
outputvar        = '[WI_s; We_s]';
input_to_G_s     = '[C3_s+u_Delta]';
input_to_K_s     = '[Wr_s-G_s-Gd_s]';
input_to_WI_s    = '[C3_s]';
input_to_C3_s    = '[K_s]';
input_to_Wd_s    = '[d]';
input_to_Gd_s    = '[Wd_s]';
input_to_Wr_s    = '[r]';
input_to_We_s    = '[Wr_s-G_s-Gd_s]';

sysoutname       = 'N_s';
cleanupsysic     = 'yes';
sysic;

omega      = logspace(-4,1,50);
```

```
N_omega    = frsp(N_s,omega);

M_omega    = sel(N_omega,1:5,1:5);
N22_omega  = sel(N_omega,6:8,6:12);


% input uncertainty
Delta      = [1 1; 1 1; 1 1; 1 1; 1 1];
% fictitious performance uncertainty
Delta_p    = [7 3];

% robust stability
[mu_RS, dvec, sens, pvec] = mu(M_omega,Delta);
clf; vplot('liv,m',sel(mu_RS,1,1),vnorm(M_omega));
xlabel('frequency');
if print_to_file,
  print('-deps','muH2RS'); unix('ps2frag muH2RS.eps');
end;

disp(['mu for RS:  ',num2str(pkvnorm(sel(mu_RS,1,1)))]);
disp('Press a key'); pause;

% nominal performance - full matrix/no need to calculate mu

clf; vplot('liv,m',vnorm(N22_omega));
xlabel('frequency');
if print_to_file,
  print('-deps','muH2NP'); unix('ps2frag muH2NP.eps');
end;

disp(['mu for NP:  ',num2str(max(vunpck(vnorm(N22_omega))))]);
disp('Press a key'); pause;

% robust performance

[mu_RP, dvec, sens, pvec] = mu(N_omega,[Delta; Delta_p]);
clf; vplot('liv,m',sel(mu_RP,1,1),vnorm(N_omega));
xlabel('frequency'); axis([1e-4 1 0 3]);
if print_to_file,
  print('-deps','muH2RP'); unix('ps2frag muH2RP.eps');
end;

disp(['mu for RP:  ',num2str(pkvnorm(sel(mu_RP,1,1)))]);
```

```
%
%  mu_Hinf
%
%  mu-analysis for Hinf controller
%
%  $Id$
%

% first, specify weights
%

% weight on the disturbance
wdC    = nd2sys(1, [1 1e-2], 1e-2);
Wd_s   = daug( wdC, 1e-4, 1e-4, 1e-4 );

% weight on the control error
we_x    = nd2sys([1 8e-2],[1 1e-5], 1/2);
we_y    = we_x; we_psi = we_x;
We_s   = sysbal(daug( we_x, we_y, we_psi ));

% weight on the reference
Wr_s    = 0.1*eye(3);

% weight on input uncertainty
wI     = nd2sys([1 5e-1],[1 1e1],1);
WI_s   = daug( wI, wI, wI, wI, wI );

C3_s   = pck([],[],[],pinv(T));

systemnames      = 'G_s Gd_s We_s WI_s Wr_s Wd_s C3_s K_s';
inputvar         = '[u_Delta(5); r(3); d(4)]';
outputvar        = '[WI_s; We_s]';
input_to_G_s     = '[C3_s+u_Delta]';
input_to_K_s     = '[Wr_s-G_s-Gd_s]';
input_to_WI_s    = '[C3_s]';
input_to_Wr_s    = '[r]';
input_to_C3_s    = '[K_s]';
input_to_Gd_s    = '[Wd_s]';
input_to_Wd_s    = '[d]';
input_to_We_s    = '[Wr_s-G_s-Gd_s]';

sysoutname       = 'N_s';
cleanupsysic     = 'yes';
```

```
sysic;

omega      = logspace(-4,1,50);

N_omega    = frsp(N_s,omega);

M_omega    = sel(N_omega,1:5,1:5);
N22_omega  = sel(N_omega,6:8,6:12);


% input uncertainty
Delta      = [1 1; 1 1; 1 1; 1 1; 1 1];
% fictitious performance uncertainty
Delta_p    = [7 3];

% robust stability
[mu_RS, dvec, sens, pvec] = mu(M_omega,Delta);
clf; vplot('liv,m',sel(mu_RS,1,1),vnorm(M_omega));
xlabel('frequency');
if print_to_file,
  print('-deps','muHinfRS'); unix('ps2frag muHinfRS.eps');
end;

disp(['mu for RS:  ',num2str(pkvnorm(sel(mu_RS,1,1)))]);
disp('Press a key'); pause;

% nominal performance - full matrix/no need to calculate mu

clf; vplot('liv,m',vnorm(N22_omega));
xlabel('frequency');
if print_to_file,
  print('-deps','muHinfNP'); unix('ps2frag muHinfNP.eps');
end;

disp(['mu for NP:  ',num2str(max(vunpck(vnorm(N22_omega))))]);
disp('Press a key'); pause;

% robust performance

[mu_RP, dvec, sens, pvec] = mu(N_omega,[Delta; Delta_p]);
clf; vplot('liv,m',sel(mu_RP,1,1),vnorm(N_omega));
xlabel('frequency');
if print_to_file,
  print('-deps','muHinfRP'); unix('ps2frag muHinfRP.eps');
end;

disp(['mu for RP:  ',num2str(pkvnorm(sel(mu_RP,1,1)))]);
```