



European Synchrotron Radiation Facility
71, avenue des Martyrs
CS 40220
38043 Grenoble Cedex 9

Robust and Optimal Sensor Fusion - Matlab Computation

DEHAEZE *Thomas*

dehaeze.thomas@gmail.com

October 26, 2020

Table of Contents

1	Sensor Description	2
1.1	Sensor Dynamics	2
1.2	Sensor Model Uncertainty	3
1.3	Sensor Noise	4
1.4	Save Model	4
2	Introduction to Sensor Fusion	6
2.1	Sensor Fusion Architecture	6
2.2	Super Sensor Noise	6
2.3	Super Sensor Dynamical Uncertainty	6
3	Optimal Super Sensor Noise: \mathcal{H}_2 Synthesis	8
3.1	\mathcal{H}_2 Synthesis	8
3.2	Super Sensor Noise	9
3.3	Discrepancy between sensor dynamics and model	9
4	Robust Sensor Fusion: \mathcal{H}_∞ Synthesis	12
4.1	Weighting Function used to bound the super sensor uncertainty	12
4.2	\mathcal{H}_∞ Synthesis	13
4.3	Super sensor uncertainty	14
4.4	Super sensor noise	14
4.5	Conclusion	15
5	Optimal and Robust Sensor Fusion: Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ Synthesis	17
5.1	Mixed $\mathcal{H}_2 / \mathcal{H}_\infty$ Synthesis	17
5.2	Obtained Super Sensor's noise	18
5.3	Obtained Super Sensor's Uncertainty	20
5.4	Conclusion	20
6	Matlab Functions	21
6.1	createWeight	21
6.2	plotMagUncertainty	21
6.3	plotPhaseUncertainty	22

This document is arranged as follows:

- Section 1: the sensors are described (dynamics, uncertainty, noise)
- Section 2: the sensor fusion architecture is described and the super sensor noise and dynamical uncertainty are derived
- Section 3: the \mathcal{H}_2 synthesis is used to design complementary filters such that the RMS value of the super sensor's noise is minimized
- Section 4: the \mathcal{H}_∞ synthesis is used to design complementary filters such that the super sensor's uncertainty is bonded to acceptable values
- Section 5: the mixed $\mathcal{H}_2/\mathcal{H}_\infty$ synthesis is used to both limit the super sensor's uncertainty and to lower the RMS value of the super sensor's noise
- Section 6: Matlab functions used for the analysis are described

1 Sensor Description

In Figure 1 is shown a schematic of a sensor model that is used in the following study. In this example, the measured quantity x is the velocity of an object.

Table 1 – Description of signals in Figure 1

Notation	Meaning	Unit
x	Physical measured quantity	$[m/s]$
\tilde{n}_i	White noise with unitary PSD	
n_i	Shaped noise	$[m/s]$
v_i	Sensor output measurement	$[V]$
\hat{x}_i	Estimate of x from the sensor	$[m/s]$
$\Phi_n(\omega)$	Power Spectral Density of n	$[\frac{(m/s)^2}{Hz}]$
$\phi_n(\omega)$	Amplitude Spectral Density of n	$[\frac{m/s}{\sqrt{Hz}}]$
σ_n	Root Mean Square Value of n	$[m/s \text{ rms}]$

Table 2 – Description of Systems in Figure 1

Notation	Meaning	Unit
\hat{G}_i	Nominal Sensor Dynamics	$[\frac{V}{m/s}]$
W_i	Weight representing the size of the uncertainty at each frequency	
Δ_i	Any complex perturbation such that $\ \Delta_i\ _\infty < 1$	
N_i	Weight representing the sensor noise	$[m/s]$

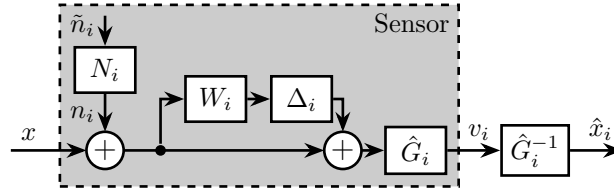


Figure 1 – Sensor Model

1.1 Sensor Dynamics

Let's consider two sensors measuring the velocity of an object.

The first sensor is an accelerometer. Its nominal dynamics $\hat{G}_1(s)$ is defined below.

```

1 m_acc = 0.01; % Inertial Mass [kg]
2 c_acc = 5; % Damping [N/(m/s)]
3 k_acc = 1e5; % Stiffness [N/m]
4 g_acc = 1e5; % Gain [V/m]
5
6 G1 = g_acc*m_acc*s/(m_acc*s^2 + c_acc*s + k_acc); % Accelerometer Plant [V/(m/s)]

```

The second sensor is a displacement sensor, its nominal dynamics $\hat{G}_2(s)$ is defined below.

```

1 w_pos = 2*pi*2e3; % Measurement Bandwidth [rad/s]
2 g_pos = 1e4; % Gain [V/m]
3
4 G2 = g_pos/s/(1 + s/w_pos); % Position Sensor Plant [V/(m/s)]

```

These nominal dynamics are also taken as the model of the sensor dynamics. The true sensor dynamics has some uncertainty associated to it and described in section 1.2.

Both sensor dynamics in $[\frac{V}{m/s}]$ are shown in Figure 2.

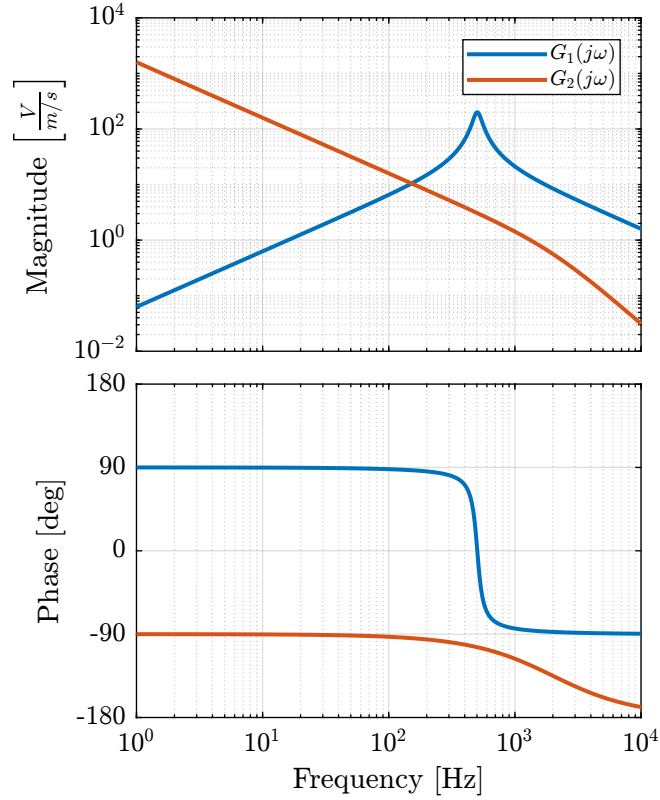


Figure 2 – Sensor nominal dynamics from the velocity of the object to the output voltage

1.2 Sensor Model Uncertainty

The uncertainty on the sensor dynamics is described by multiplicative uncertainty (Figure 1). The true sensor dynamics $G_i(s)$ is then described by (1).

$$G_i(s) = \hat{G}_i(s) (1 + W_i(s)\Delta_i(s)); \quad |\Delta_i(j\omega)| < 1 \forall \omega \quad (1)$$

The weights $W_i(s)$ representing the dynamical uncertainty are defined below and their magnitude is shown in Figure 3.

```

1 W1 = createWeight('n', 2, 'w0', 2*pi*3, 'G0', 2, 'G1', 0.1, 'Gc', 1) * ...
2   createWeight('n', 2, 'w0', 2*pi*1e3, 'G0', 1, 'G1', 4/0.1, 'Gc', 1/0.1);
3
4 W2 = createWeight('n', 2, 'w0', 2*pi*1e2, 'G0', 0.05, 'G1', 4, 'Gc', 1);

```

The bode plot of the sensors nominal dynamics as well as their defined dynamical spread are shown in Figure 4.

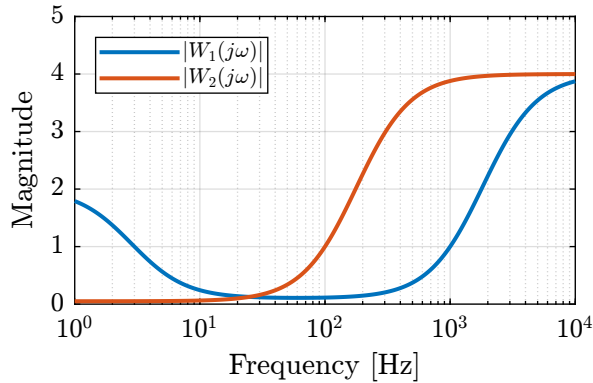


Figure 3 – Magnitude of the multiplicative uncertainty weights $|W_i(j\omega)|$

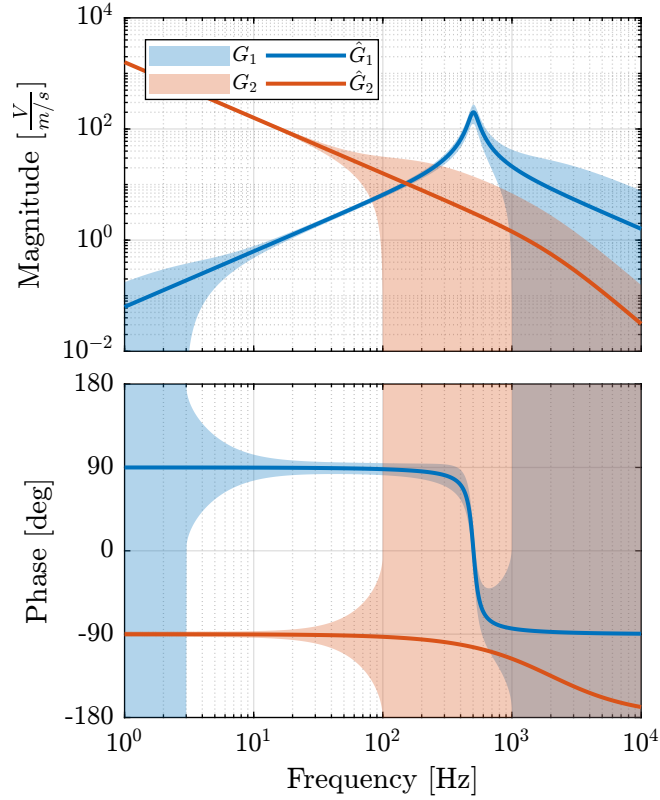


Figure 4 – Nominal Sensor Dynamics \hat{G}_i (solid lines) as well as the spread of the dynamical uncertainty (background color)

1.3 Sensor Noise

The noise of the sensors n_i are modelled by shaping a white noise with unitary PSD \tilde{n}_i (2) with a LTI transfer function $N_i(s)$ (Figure 1).

$$\Phi_{\tilde{n}_i}(\omega) = 1 \quad (2)$$

The Power Spectral Density of the sensor noise $\Phi_{n_i}(\omega)$ is then computed using (3) and expressed in $[\frac{(m/s)^2}{Hz}]$.

$$\Phi_{n_i}(\omega) = |N_i(j\omega)|^2 \Phi_{\tilde{n}_i}(\omega) \quad (3)$$

The weights N_1 and N_2 representing the amplitude spectral density of the sensor noises are defined below and shown in Figure 5.

```

1  omegac = 0.15*2*pi; G0 = 1e-1; Ginf = 1e-6;
2  N1 = (Ginf*s/omegac + G0)/(s/omegac + 1)/(1 + s/2/pi/1e4);
3
4  omegac = 1000*2*pi; G0 = 1e-6; Ginf = 1e-3;
5  N2 = (Ginf*s/omegac + G0)/(s/omegac + 1)/(1 + s/2/pi/1e4);

```

1.4 Save Model

All the dynamical systems representing the sensors are saved for further use.

```

1  save('./mat/model.mat', 'freqs', 'G1', 'G2', 'N2', 'N1', 'W2', 'W1');

```

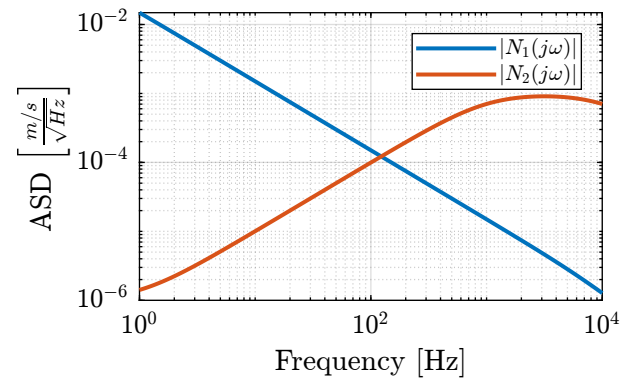


Figure 5 – Amplitude spectral density of the sensors $\sqrt{\Phi_{n_i}(\omega)} = |N_i(j\omega)|$

2 Introduction to Sensor Fusion

2.1 Sensor Fusion Architecture

The two sensors presented in Section 1 are now merged together using complementary filters $H_1(s)$ and $H_2(s)$ to form a super sensor (Figure 6).

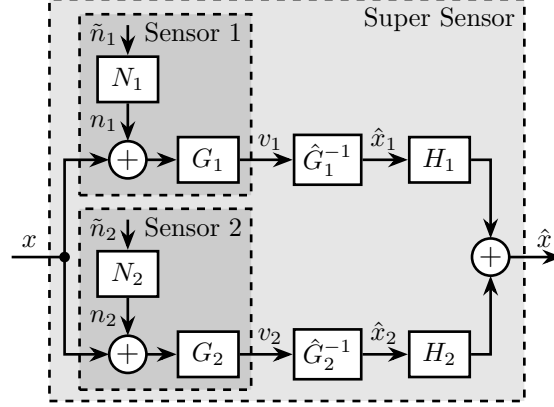


Figure 6 – Sensor Fusion Architecture

The complementary property of $H_1(s)$ and $H_2(s)$ means that the sum of their transfer function is equal to 1 (4).

$$H_1(s) + H_2(s) = 1 \quad (4)$$

The super sensor estimate \hat{x} is given by (5).

$$\hat{x} = \left(H_1 \hat{G}_1^{-1} G_1 + H_2 \hat{G}_2^{-1} G_2 \right) x + \left(H_1 \hat{G}_1^{-1} G_1 N_1 \right) \tilde{n}_1 + \left(H_2 \hat{G}_2^{-1} G_2 N_2 \right) \tilde{n}_2 \quad (5)$$

2.2 Super Sensor Noise

If we first suppose that the models of the sensors \hat{G}_i are very close to the true sensor dynamics G_i (6), we have that the super sensor estimate \hat{x} is equals to the measured quantity x plus the noise of the two sensors filtered out by the complementary filters (7).

$$\hat{G}_i^{-1}(s)G_i(s) \approx 1 \quad (6)$$

$$\hat{x} = x + \underbrace{(H_1 N_1) \tilde{n}_1 + (H_2 N_2) \tilde{n}_2}_n \quad (7)$$

As the noise of both sensors are considered to be uncorrelated, the PSD of the super sensor noise is computed as follow:

$$\Phi_n(\omega) = |H_1(j\omega)N_1(j\omega)|^2 + |H_2(j\omega)N_2(j\omega)|^2 \quad (8)$$

And the Root Mean Square (RMS) value of the super sensor noise σ_n is given by Equation (9).

$$\sigma_n = \sqrt{\int_0^\infty \Phi_n(\omega) d\omega} \quad (9)$$

2.3 Super Sensor Dynamical Uncertainty

If we consider some dynamical uncertainty (the true system dynamics G_i not being perfectly equal to our model \hat{G}_i) that we model by the use of multiplicative uncertainty (Figure 7), the super sensor dynamics is then equals to:

$$\begin{aligned} \frac{\hat{x}}{x} &= \left(H_1 \hat{G}_1^{-1} \hat{G}_1 (1 + W_1 \Delta_1) + H_2 \hat{G}_2^{-1} \hat{G}_2 (1 + W_2 \Delta_2) \right) \\ &= \left(H_1 (1 + W_1 \Delta_1) + H_2 (1 + W_2 \Delta_2) \right) \\ &= (1 + H_1 W_1 \Delta_1 + H_2 W_2 \Delta_2), \quad \|\Delta_i\|_\infty < 1 \end{aligned} \quad (10)$$

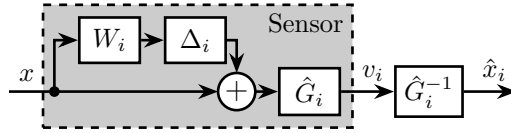


Figure 7 – Sensor Model including Dynamical Uncertainty

The uncertainty set of the transfer function from \hat{x} to x at frequency ω is bounded in the complex plane by a circle centered on 1 and with a radius equal to $|W_1(j\omega)H_1(j\omega)| + |W_2(j\omega)H_2(j\omega)|$ as shown in Figure 8.

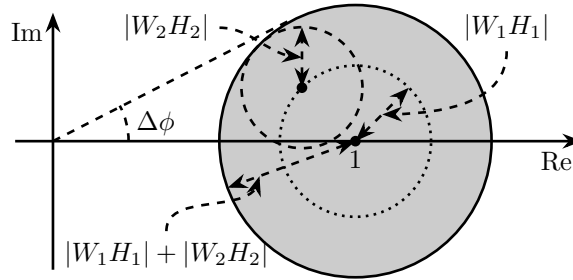


Figure 8 – Super Sensor model uncertainty displayed in the complex plane

3 Optimal Super Sensor Noise: \mathcal{H}_2 Synthesis

In this section, the complementary filters $H_1(s)$ and $H_2(s)$ are designed in order to minimize the RMS value of super sensor noise σ_n .

The RMS value of the super sensor noise is (neglecting the model uncertainty):

$$\begin{aligned} \sigma_n &= \sqrt{\int_0^\infty |H_1(j\omega)N_1(j\omega)|^2 + |H_2(j\omega)N_2(j\omega)|^2 d\omega} \\ &= \left\| \begin{matrix} H_1 N_1 \\ H_2 N_2 \end{matrix} \right\|_2 \end{aligned} \tag{11}$$

The goal is to design $H_1(s)$ and $H_2(s)$ such that $H_1(s) + H_2(s) = 1$ (complementary property) and such that $\left\| \begin{matrix} H_1 N_1 \\ H_2 N_2 \end{matrix} \right\|_2$ is minimized (minimized RMS value of the super sensor noise). This is done using the \mathcal{H}_2 synthesis in Section 3.1.

3.1 \mathcal{H}_2 Synthesis

Consider the generalized plant $P_{\mathcal{H}_2}$ shown in Figure 9 and described by Equation (12).

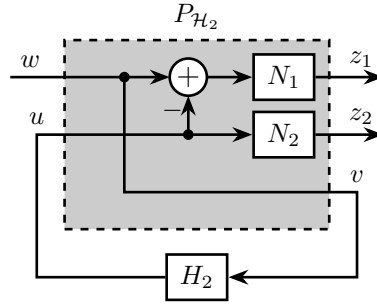


Figure 9 – Architecture used for \mathcal{H}_∞ synthesis of complementary filters

$$\begin{pmatrix} z_1 \\ z_2 \\ v \end{pmatrix} = \underbrace{\begin{bmatrix} N_1 & -N_1 \\ 0 & N_2 \\ 1 & 0 \end{bmatrix}}_{P_{\mathcal{H}_2}} \begin{pmatrix} w \\ u \end{pmatrix} \tag{12}$$

Applying the \mathcal{H}_2 synthesis on $P_{\mathcal{H}_2}$ will generate a filter $H_2(s)$ such that the \mathcal{H}_2 norm from w to (z_1, z_2) which is actually equals to σ_n by defining $H_1(s) = 1 - H_2(s)$:

$$\left\| \begin{matrix} z_1/w \\ z_2/w \end{matrix} \right\|_2 = \left\| \begin{matrix} N_1(1 - H_2) \\ N_2 H_2 \end{matrix} \right\|_2 = \sigma_n \quad \text{with} \quad H_1(s) = 1 - H_2(s) \tag{13}$$

We then have that the \mathcal{H}_2 synthesis applied on $P_{\mathcal{H}_2}$ generates two complementary filters $H_1(s)$ and $H_2(s)$ such that the RMS value of super sensor noise is minimized.

The generalized plant $P_{\mathcal{H}_2}$ is defined below

```
1 PH2 = [N1 -N1;
2         0  N2;
3         1  0];
```

The \mathcal{H}_2 synthesis using the `h2syn` command

```
1 [H2, -, gamma] = h2syn(PH2, 1, 1);
```

Finally, $H_1(s)$ is defined as follows

```
1 H1 = 1 - H2;
```

The obtained complementary filters are shown in Figure 10.

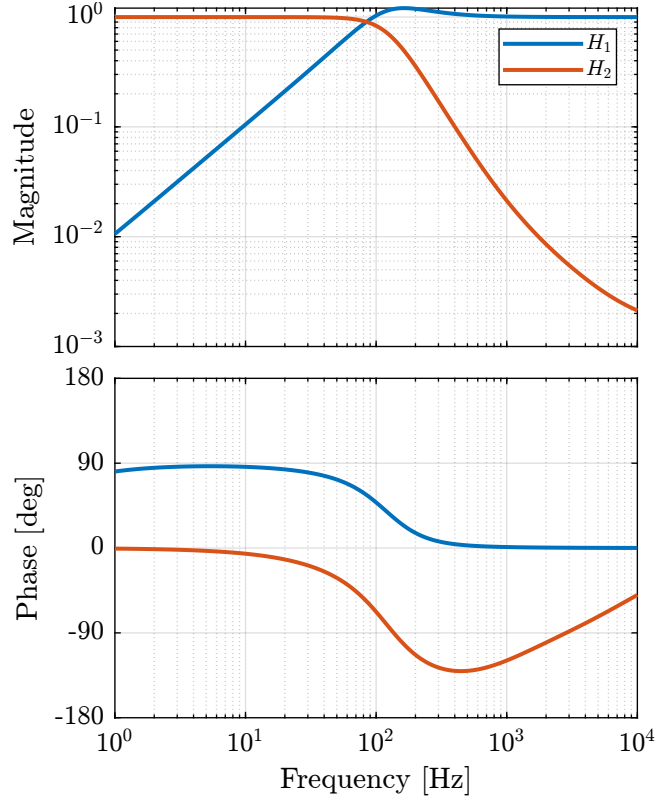


Figure 10 – Obtained complementary filters using the \mathcal{H}_2 Synthesis

3.2 Super Sensor Noise

The Power Spectral Density of the individual sensors' noise Φ_{n_1} , Φ_{n_2} and of the super sensor noise $\Phi_{n_{\mathcal{H}_2}}$ are computed below.

```

1 PSD_S1 = abs(squeeze(freqresp(N1, freqs, 'Hz'))).^2;
2 PSD_S2 = abs(squeeze(freqresp(N2, freqs, 'Hz'))).^2;
3 PSD_H2 = abs(squeeze(freqresp(N1*H1, freqs, 'Hz'))).^2 + ...
4           abs(squeeze(freqresp(N2*H2, freqs, 'Hz'))).^2;

```

The obtained ASD are shown in Figure 11.

The RMS value of the individual sensors and of the super sensor are listed in Table 3.

Table 3 – RMS value of the individual sensor noise and of the super sensor using the \mathcal{H}_2 Synthesis

	RMS value [m/s]
σ_{n_1}	0.015
σ_{n_2}	0.080
$\sigma_{n_{\mathcal{H}_2}}$	0.003

A time domain simulation is now performed. The measured velocity x is set to be a sweep sine with an amplitude of 0.1 [m/s]. The velocity estimates from the two sensors and from the super sensors are shown in Figure 12. The resulting noises are displayed in Figure 13.

3.3 Discrepancy between sensor dynamics and model

If we consider sensor dynamical uncertainty as explained in Section 1.2, we can compute what would be the super sensor dynamical uncertainty when using the complementary filters obtained using the \mathcal{H}_2 Synthesis.

The super sensor dynamical uncertainty is shown in Figure 14.

It is shown that the phase uncertainty is not bounded between 100Hz and 200Hz. As a result the super sensor signal can not be used for feedback applications about 100Hz.

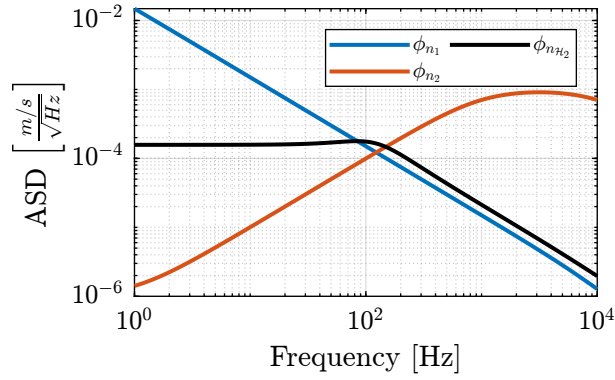


Figure 11 – Power Spectral Density of the estimated \hat{x} using the two sensors alone and using the optimally fused signal

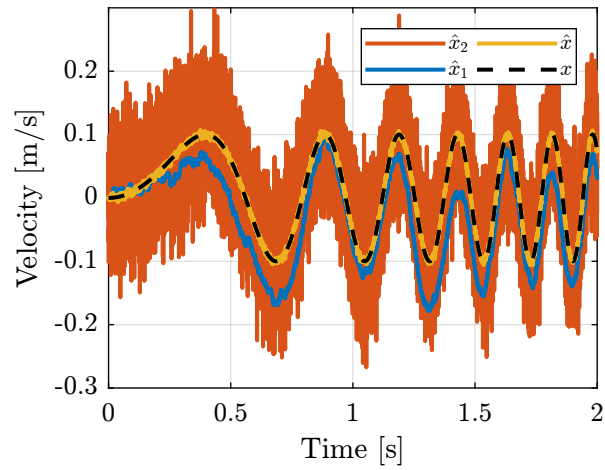


Figure 12 – Noise of individual sensors and noise of the super sensor

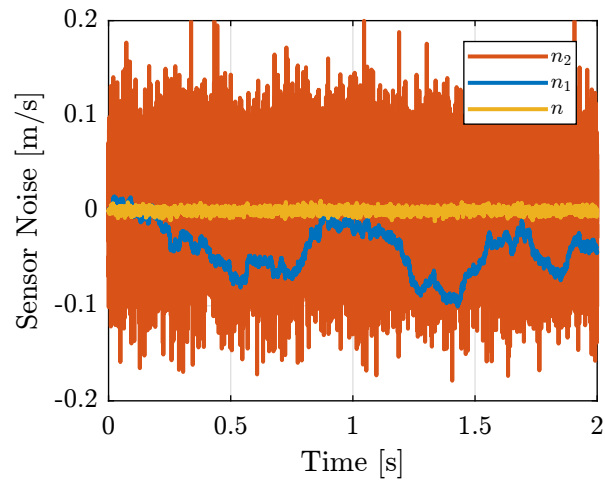


Figure 13 – Noise of the two sensors n_1, n_2 and noise of the super sensor n

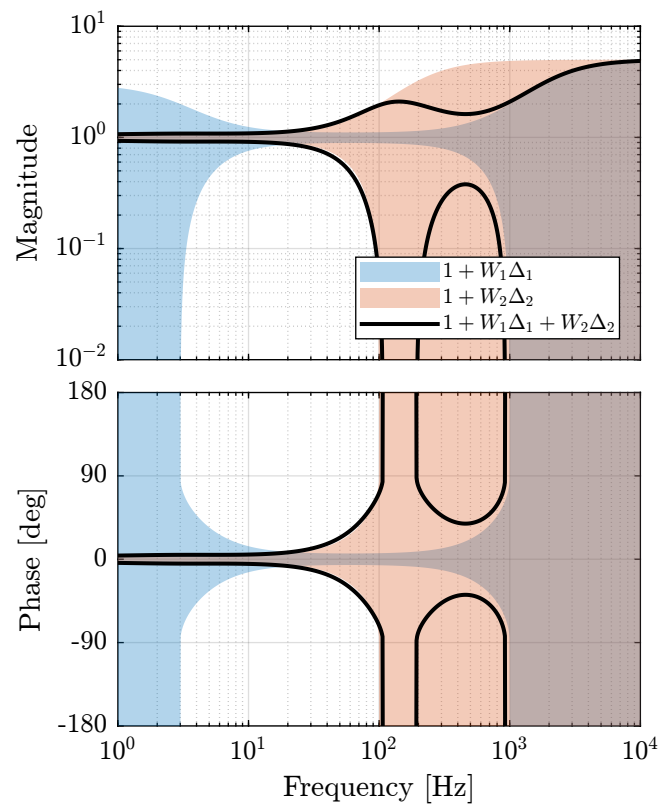


Figure 14 – Super sensor dynamical uncertainty when using the \mathcal{H}_2 Synthesis

4 Robust Sensor Fusion: \mathcal{H}_∞ Synthesis

We initially considered perfectly known sensor dynamics so that it can be perfectly inverted.

We now take into account the fact that the sensor dynamics is only partially known. To do so, we model the uncertainty that we have on the sensor dynamics by multiplicative input uncertainty as shown in Figure 15.

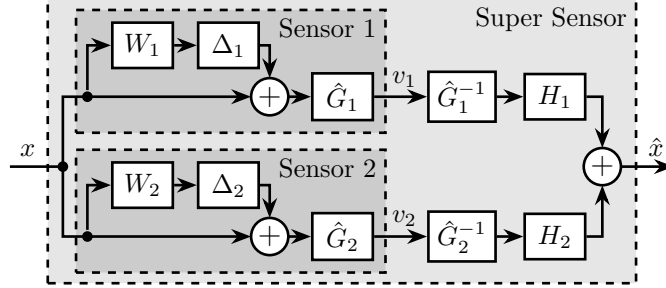


Figure 15 – Sensor fusion architecture with sensor dynamics uncertainty

As explained in Section 1.2, at each frequency ω , the dynamical uncertainty of the super sensor can be represented in the complex plane by a circle with a radius equals to $|H_1(j\omega)W_1(j\omega)| + |H_2(j\omega)W_2(j\omega)|$ and centered on 1. In order to specify a wanted upper bound on the dynamical uncertainty, a weight $W_u(s)$ is used where $1/|W_u(j\omega)|$ represents the maximum allowed radius of the uncertainty circle corresponding to the super sensor dynamics at a frequency ω (14).

$$|H_1(j\omega)W_1(j\omega)| + |H_2(j\omega)W_2(j\omega)| < \frac{1}{|W_u(j\omega)|}, \quad \forall \omega \quad (14)$$

$$\Leftrightarrow |H_1(j\omega)W_1(j\omega)W_u(j\omega)| + |H_2(j\omega)W_2(j\omega)W_u(j\omega)| < 1, \quad \forall \omega \quad (15)$$

$|W_u(j\omega)|$ is also linked to the gain uncertainty ΔG (16) and phase uncertainty $\Delta\phi$ (17) of the super sensor.

$$\Delta G(\omega) \leq \frac{1}{|W_u(j\omega)|}, \quad \forall \omega \quad (16)$$

$$\Delta\phi(\omega) \leq \arcsin\left(\frac{1}{|W_u(j\omega)|}\right), \quad \forall \omega \quad (17)$$

The choice of W_u is presented in Section 4.1.

Condition (15) can almost be represented by (18) (within a factor $\sqrt{2}$).

$$\left\| \frac{H_1(s)W_1(s)W_u(s)}{H_2(s)W_2(s)W_u(s)} \right\|_\infty < 1 \quad (18)$$

The objective is to design $H_1(s)$ and $H_2(s)$ such that $H_1(s) + H_2(s) = 1$ (complementary property) and such that (18) is verified (bounded dynamical uncertainty).

This is done using the \mathcal{H}_∞ synthesis in Section 4.2.

4.1 Weighting Function used to bound the super sensor uncertainty

$W_u(s)$ is defined such that the super sensor phase uncertainty is less than 10 degrees below 100Hz (19) and is less than 180 degrees below 400Hz (20).

$$\frac{1}{|W_u(j\omega)|} < \sin\left(10\frac{\pi}{180}\right), \quad \omega < 100 \text{ Hz} \quad (19)$$

$$\frac{1}{|W_u(j2\pi 400)|} < 1 \quad (20)$$

The uncertainty bounds of the two individual sensor as well as the wanted maximum uncertainty bounds of the super sensor are shown in Figure 16.

```

1 Dphi = 10; % [deg]
2
3 Wu = createWeight('n', 2, 'w0', 2*pi*4e2, 'G0', 1/sin(Dphi*pi/180), 'G1', 1/4, 'Gc', 1);

```

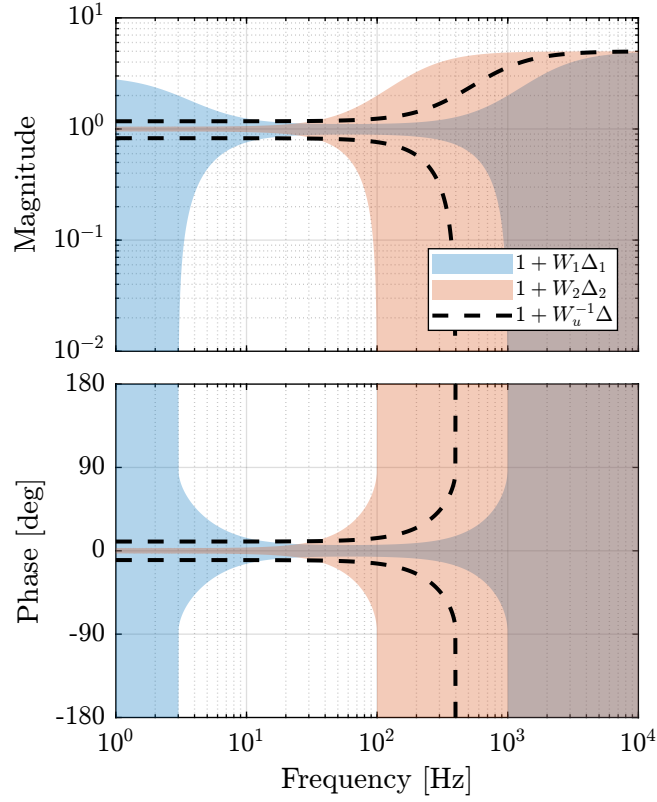


Figure 16 – Uncertainty region of the two sensors as well as the wanted maximum uncertainty of the super sensor (dashed lines)

4.2 \mathcal{H}_∞ Synthesis

The generalized plant $P_{\mathcal{H}_\infty}$ used for the \mathcal{H}_∞ Synthesis of the complementary filters is shown in Figure 17 and is described by Equation (21).

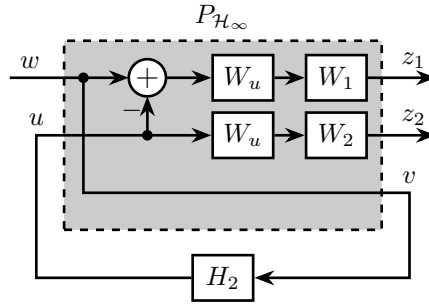


Figure 17 – Architecture used for \mathcal{H}_∞ synthesis of complementary filters

$$\begin{pmatrix} z_1 \\ z_2 \\ v \end{pmatrix} = \underbrace{\begin{bmatrix} W_u W_1 & -W_u W_1 \\ 0 & W_u W_2 \\ 1 & 0 \end{bmatrix}}_{P_{\mathcal{H}_\infty}} \begin{pmatrix} w \\ u \end{pmatrix} \quad (21)$$

The generalized plant is defined below.

```

1 P = [Wu*W1 -Wu*W1;
2       0      Wu*W2;
3       1      0];

```

And the \mathcal{H}_∞ synthesis is performed using the `hinfscn` command.

```

1 H2 = hinfscn(P, 1, 1, 'TOLGAM', 0.001, 'DISPLAY', 'on');

```

Test bounds: 0.7071 <= gamma <= 1.291

gamma	X>=0	Y>=0	rho(XY)<1	p/f
9.554e-01	0.0e+00	0.0e+00	3.529e-16	p
8.219e-01	0.0e+00	0.0e+00	5.204e-16	p
7.624e-01	3.8e-17	0.0e+00	1.955e-15	p
7.342e-01	0.0e+00	0.0e+00	5.612e-16	p
7.205e-01	0.0e+00	0.0e+00	7.184e-16	p
7.138e-01	0.0e+00	0.0e+00	0.000e+00	p
7.104e-01	4.1e-16	0.0e+00	6.749e-15	p
7.088e-01	0.0e+00	0.0e+00	2.794e-15	p
7.079e-01	0.0e+00	0.0e+00	6.503e-16	p
7.075e-01	0.0e+00	0.0e+00	4.302e-15	p

Best performance (actual): 0.7071

The \mathcal{H}_∞ is successful as the \mathcal{H}_∞ norm of the “closed loop” transfer function from (w) to (z_1, z_2) is less than one. $H_1(s)$ is then defined as the complementary of $H_2(s)$.

1 $H_1 = 1 - H_2$;

The obtained complementary filters as well as the wanted upper bounds are shown in Figure 18.

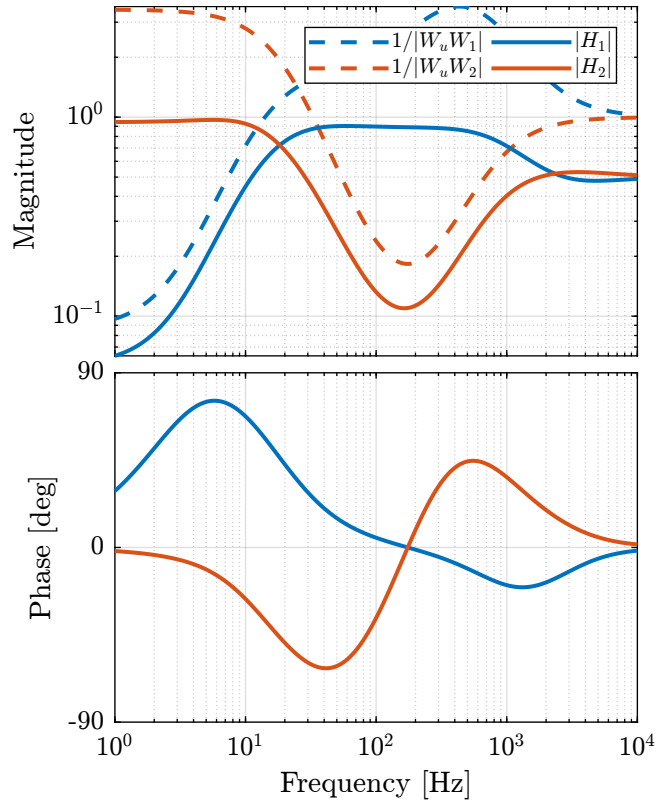


Figure 18 – Obtained complementary filters using the \mathcal{H}_∞ Synthesis

4.3 Super sensor uncertainty

The super sensor dynamical uncertainty is displayed in Figure 19. It is confirmed that the super sensor dynamical uncertainty is less than the maximum allowed uncertainty defined by the norm of $W_u(s)$.

The \mathcal{H}_∞ synthesis thus allows to design filters such that the super sensor has specified bounded uncertainty.

4.4 Super sensor noise

We now compute the obtain Power Spectral Density of the super sensor’s noise. The Amplitude Spectral Densities are shown in Figure 20.

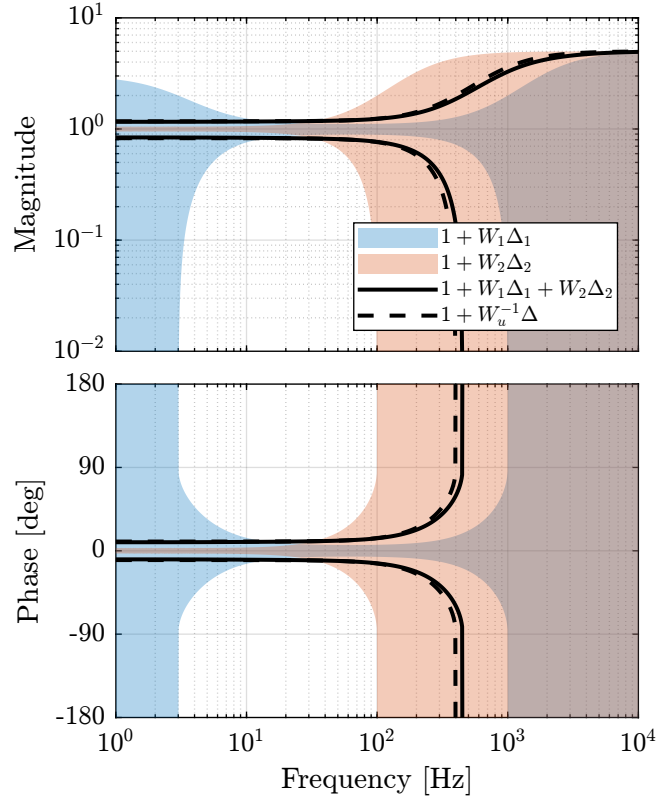


Figure 19 – Super sensor dynamical uncertainty (solid curve) when using the \mathcal{H}_∞ Synthesis

```

1 PSD_S2 = abs(squeeze(freqresp(N2, freqs, 'Hz'))).^2;
2 PSD_S1 = abs(squeeze(freqresp(N1, freqs, 'Hz'))).^2;
3 PSD_Hinf = abs(squeeze(freqresp(N1*H1, freqs, 'Hz'))).^2 + ...
4             abs(squeeze(freqresp(N2*H2, freqs, 'Hz'))).^2;

```

The obtained RMS of the super sensor noise in the \mathcal{H}_2 and \mathcal{H}_∞ case are shown in Table 4. As expected, the super sensor obtained from the \mathcal{H}_∞ synthesis is much noisier than the super sensor obtained from the \mathcal{H}_2 synthesis.

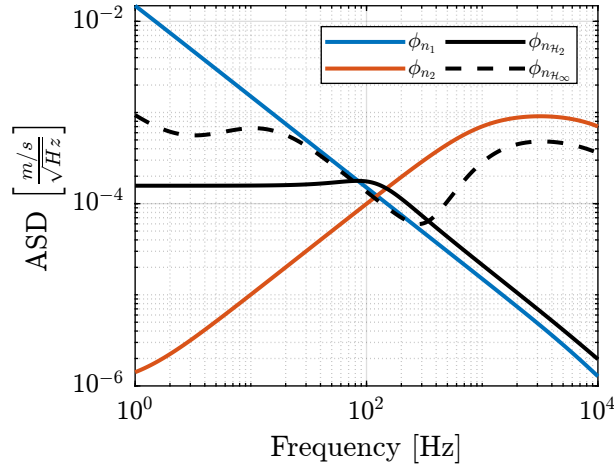


Figure 20 – Power Spectral Density of the estimated \hat{x} using the two sensors alone and using the \mathcal{H}_∞ synthesis

4.5 Conclusion

Using the \mathcal{H}_∞ synthesis, the dynamical uncertainty of the super sensor can be bounded to acceptable values. However, the RMS of the super sensor noise is not optimized as it was the case with the \mathcal{H}_2 synthesis

Table 4 – *Comparison of the obtained RMS noise of the super sensor*

	RMS [m/s]
Optimal: \mathcal{H}_2	0.0027
Robust: \mathcal{H}_∞	0.041

5 Optimal and Robust Sensor Fusion: Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ Synthesis

The (optima) \mathcal{H}_2 synthesis and the (robust) \mathcal{H}_∞ synthesis are now combined to form an Optimal and Robust synthesis of complementary filters for sensor fusion.

The sensor fusion architecture is shown in Figure 21 (\hat{G}_i are omitted for space reasons).

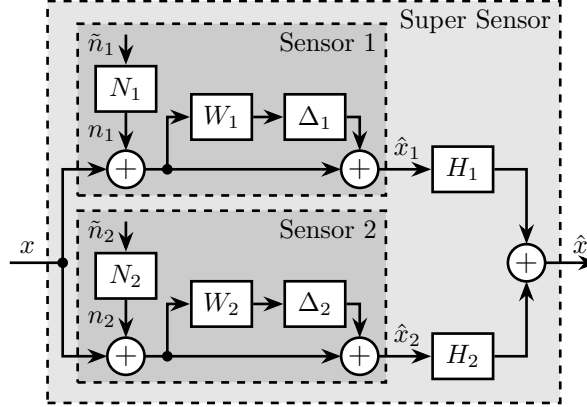


Figure 21 – Sensor fusion architecture with sensor dynamics uncertainty

The goal is to design complementary filters such that:

- the maximum uncertainty of the super sensor is bounded to acceptable values (defined by $W_u(s)$)
- the RMS value of the super sensor noise is minimized

To do so, we can use the Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ Synthesis presented in Section 5.1.

5.1 Mixed $\mathcal{H}_2 / \mathcal{H}_\infty$ Synthesis

The synthesis architecture that is used here is shown in Figure 22.

The filter $H_2(s)$ is synthesized such that it:

- keeps the \mathcal{H}_∞ norm of the transfer function from w to $z_{\mathcal{H}_\infty}$ below some specified value
- minimizes the \mathcal{H}_2 norm of the transfer function from w to $z_{\mathcal{H}_2}$

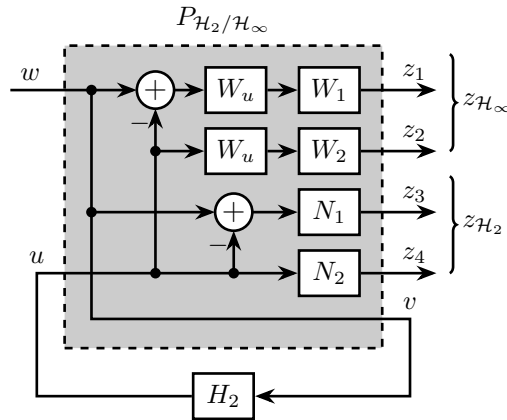


Figure 22 – Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ Synthesis

Let's see that with $H_1(s) = 1 - H_2(s)$

$$\left\| \frac{z_\infty}{w} \right\|_\infty = \left\| \frac{H_1(s)W_1(s)W_u(s)}{H_2(s)W_2(s)W_u(s)} \right\|_\infty \quad (22)$$

$$\left\| \frac{z_2}{w} \right\|_2 = \left\| \frac{H_1(s)N_1(s)}{H_2(s)N_2(s)} \right\|_2 = \sigma_n \quad (23)$$

The generalized plant $P_{\mathcal{H}_2/\mathcal{H}_\infty}$ is defined below

```

1 W1u = ss(W2*Wu); W2u = ss(W1*Wu); % Weight on the uncertainty
2 W1n = ss(N2); W2n = ss(N1); % Weight on the noise
3
4 P = [Wu*W1 -Wu*W1;
5      0      Wu*W2;
6      N1     -N1;
7      0      N2;
8      1      0];

```

And the mixed $\mathcal{H}_2/\mathcal{H}_\infty$ synthesis is performed.

```

1 [H2, -] = h2hinfsyn(ss(P), 1, 1, 2, [0, 1], 'HINFMAX', 1, 'H2MAX', Inf, 'DKMAX', 100, 'TOL', 1e-3, 'DISPLAY', 'on');
1 H1 = 1 - H2;

```

The obtained complementary filters are shown in Figure 23.

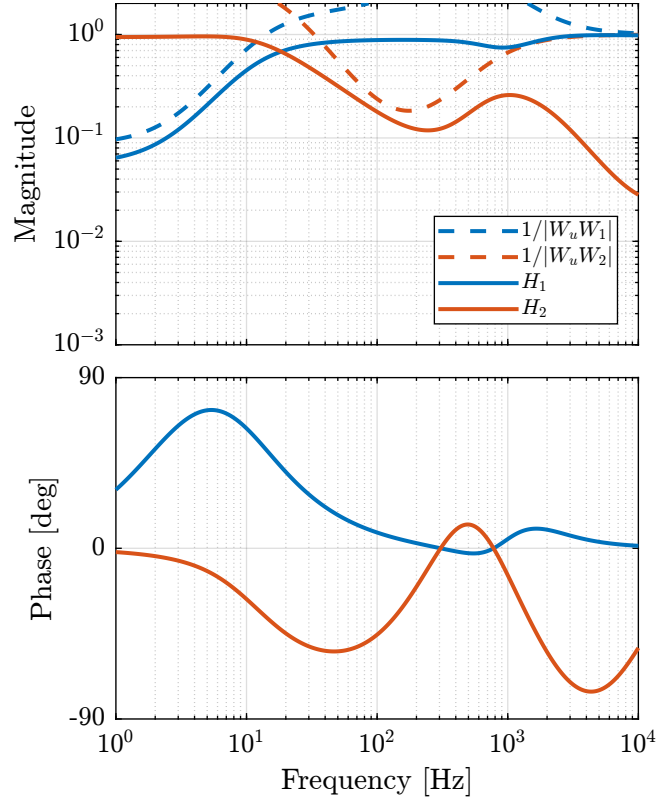


Figure 23 – Obtained complementary filters after mixed $\mathcal{H}_2/\mathcal{H}_\infty$ synthesis

5.2 Obtained Super Sensor's noise

The Amplitude Spectral Density of the super sensor's noise is shown in Figure 24.

A time domain simulation is shown in Figure 25.

The RMS values of the super sensor noise for the presented three synthesis are listed in Table 5.

```

1 PSD_S2 = abs(squeeze(freqresp(N2, freqs, 'Hz'))).^2;
2 PSD_S1 = abs(squeeze(freqresp(N1, freqs, 'Hz'))).^2;
3 PSD_H2Hinf = abs(squeeze(freqresp(N1*H1, freqs, 'Hz'))).^2 + ...
4             abs(squeeze(freqresp(N2*H2, freqs, 'Hz'))).^2;

```

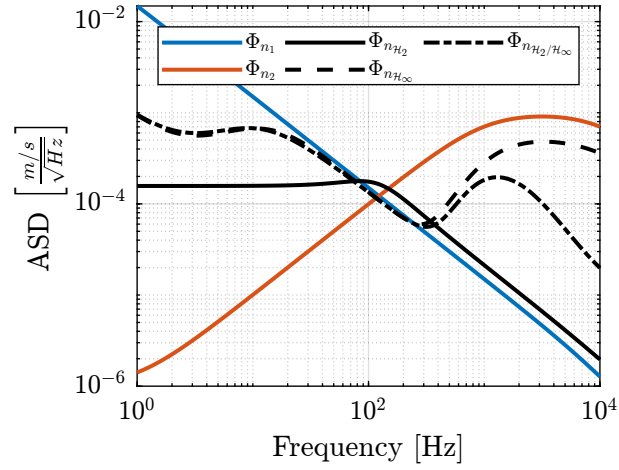


Figure 24 – Power Spectral Density of the Super Sensor obtained with the mixed $\mathcal{H}_2/\mathcal{H}_\infty$ synthesis

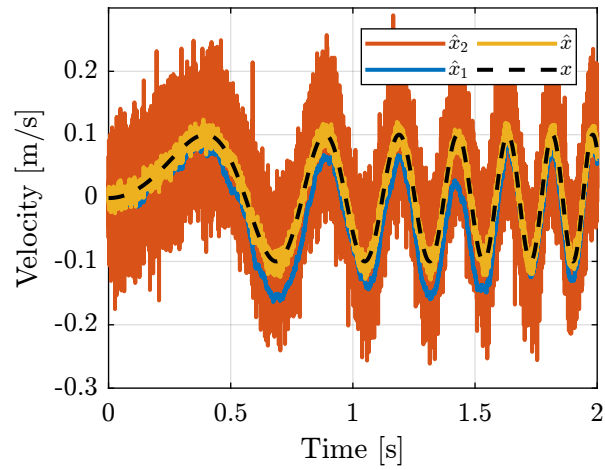


Figure 25 – Noise of individual sensors and noise of the super sensor

Table 5 – Comparison of the obtained RMS noise of the super sensor

	RMS [m/s]
Optimal: \mathcal{H}_2	0.0027
Robust: \mathcal{H}_∞	0.041
Mixed: $\mathcal{H}_2/\mathcal{H}_\infty$	0.0098

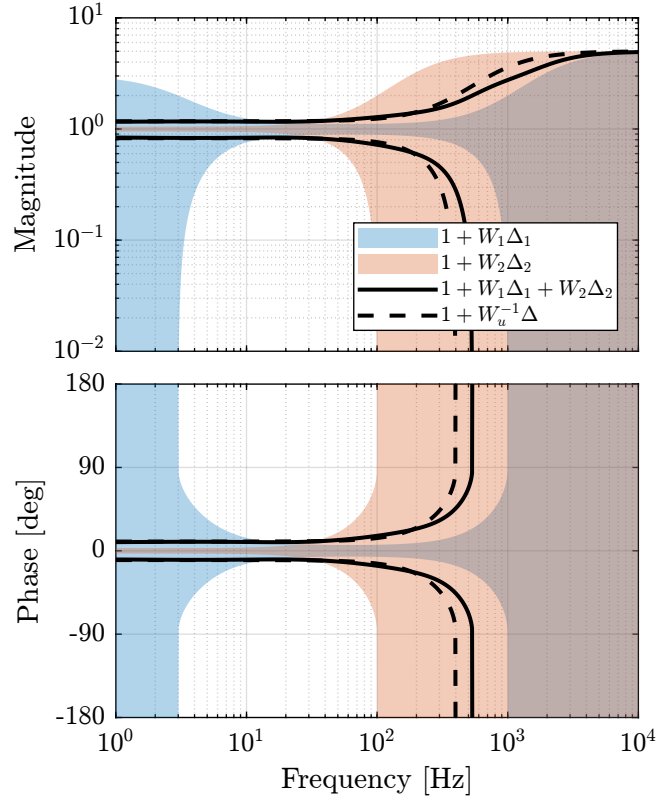


Figure 26 – Super sensor dynamical uncertainty (solid curve) when using the mixed $\mathcal{H}_2/\mathcal{H}_\infty$ Synthesis

5.3 Obtained Super Sensor's Uncertainty

The uncertainty on the super sensor's dynamics is shown in Figure 26.

5.4 Conclusion

The mixed $\mathcal{H}_2/\mathcal{H}_\infty$ synthesis of the complementary filters allows to:

- limit the dynamical uncertainty of the super sensor
- minimize the RMS value of the estimation

6 Matlab Functions

6.1 createWeight

This Matlab function is accessible [here](#).

```
1 function [W] = createWeight(args)
2 % createWeight -
3 %
4 % Syntax: [in_data] = createWeight(in_data)
5 %
6 % Inputs:
7 % - n - Weight Order
8 % - G0 - Low frequency Gain
9 % - G1 - High frequency Gain
10 % - Gc - Gain of W at frequency w0
11 % - w0 - Frequency at which |W(j w0)| = Gc
12 %
13 % Outputs:
14 % - W - Generated Weight
15
16 arguments
17 args.n (1,1) double {mustBeInteger, mustBePositive} = 1
18 args.G0 (1,1) double {mustBeNumeric, mustBePositive} = 0.1
19 args.G1 (1,1) double {mustBeNumeric, mustBePositive} = 10
20 args.Gc (1,1) double {mustBeNumeric, mustBePositive} = 1
21 args.w0 (1,1) double {mustBeNumeric, mustBePositive} = 1
22 end
23
24 mustBeBetween(args.G0, args.Gc, args.G1);
25
26 s = tf('s');
27
28 W = ((1/args.w0)*sqrt((1-(args.G0/args.Gc)^(2/args.n))/(1-(args.Gc/args.G1)^(2/args.n)))*s +
    ↪ (args.G0/args.Gc)^(1/args.n))/((1/args.G1)^(1/args.n)*(1/args.w0)*sqrt((1-(args.G0/args.Gc)^(2/args.n))/(1-(args.Gc/args.G1)^(2/args.n)))*s
    ↪ + (1/args.Gc)^(1/args.n))^(args.n);
29
30 end
31
32 % Custom validation function
33 function mustBeBetween(a,b,c)
34 if ~(a > b && b > c) || (c > b && b > a)
35 eid = 'createWeight:inputError';
36 msg = 'Gc should be between G0 and G1.';
37 throwAsCaller(MException(eid,msg))
38 end
39 end
```

6.2 plotMagUncertainty

This Matlab function is accessible [here](#).

```
1 function [p] = plotMagUncertainty(W, freqs, args)
2 % plotMagUncertainty -
3 %
4 % Syntax: [p] = plotMagUncertainty(W, freqs, args)
5 %
6 % Inputs:
7 % - W - Multiplicative Uncertainty Weight
8 % - freqs - Frequency Vector [Hz]
9 % - args - Optional Arguments:
10 % - G
11 % - color_i
12 % - opacity
13 %
14 % Outputs:
15 % - p - Plot Handle
16
17 arguments
18 W
19 freqs double {mustBeNumeric, mustBeNonnegative}
20 args.G = tf(1)
21 args.color_i (1,1) double {mustBeInteger, mustBePositive} = 1
22 args.opacity (1,1) double {mustBeNumeric, mustBeNonnegative} = 0.3
23 args.DisplayName char = ''
24 end
25
26 % Get defaults colors
27 colors = get(groot, 'defaultAxesColorOrder');
28
```

```

29 p = patch([freqs flip(freqs)], ...
30 [abs(squeeze(freqresp(args.G, freqs, 'Hz'))).*(1 + abs(squeeze(freqresp(W, freqs, 'Hz')))); ...
31 flip(abs(squeeze(freqresp(args.G, freqs, 'Hz'))).*max(1 - abs(squeeze(freqresp(W, freqs, 'Hz'))), 1e-6))], 'w', ...
32 'DisplayName', args.DisplayName);
33
34 p.FaceColor = colors(args.color_i, :);
35 p.EdgeColor = 'none';
36 p.FaceAlpha = args.opacity;
37
38 end

```

6.3 plotPhaseUncertainty

This Matlab function is accessible [here](#).

```

1 function [p] = plotPhaseUncertainty(W, freqs, args)
2 % plotPhaseUncertainty -
3 %
4 % Syntax: [p] = plotPhaseUncertainty(W, freqs, args)
5 %
6 % Inputs:
7 % - W - Multiplicative Uncertainty Weight
8 % - freqs - Frequency Vector [Hz]
9 % - args - Optional Arguments:
10 % - G
11 % - color_i
12 % - opacity
13 %
14 % Outputs:
15 % - p - Plot Handle
16
17 arguments
18 W
19 freqs double {mustBeNumeric, mustBeNonnegative}
20 args.G = tf(1)
21 args.color_i (1,1) double {mustBeInteger, mustBePositive} = 1
22 args.opacity (1,1) double {mustBeNumeric, mustBePositive} = 0.3
23 args.DisplayName char = ''
24 end
25
26 % Get defaults colors
27 colors = get(groot, 'defaultAxesColorOrder');
28
29 % Compute Phase Uncertainty
30 Dphi = 180/pi*asin(abs(squeeze(freqresp(W, freqs, 'Hz'))));
31 Dphi(abs(squeeze(freqresp(W, freqs, 'Hz')) > 1) = 360;
32
33 % Compute Plant Phase
34 G_ang = 180/pi*angle(squeeze(freqresp(args.G, freqs, 'Hz')));
35
36 p = patch([freqs flip(freqs)], [G_ang+Dphi; flip(G_ang-Dphi)], 'w', ...
37 'DisplayName', args.DisplayName);
38
39 p.FaceColor = colors(args.color_i, :);
40 p.EdgeColor = 'none';
41 p.FaceAlpha = args.opacity;
42
43 end

```

ref

References

- [1] T. Dehaeze, M. Vermat, and C. Christophe, “Complementary filters shaping using h-infinity synthesis,” in *7th International Conference on Control, Mechatronics and Automation (ICCMA)*, 2019, pp. 459–464.