DC-Motor Velocity Control Problem

Francesco Russo

Index

1. Introduction

The goal is to get acquainted with the use of the **performance weights** $w_S(s)$, $w_U(s)$ and $w_T(s)$ in the definition of the closed loop specifications.

1.1 Theoretical outlines

Before focusing on the realization of this homework, is better to give some theoretical nod.
In general, we are interested to design a controller able to perform control actions, it a way that the **closed-loop system** meets the target specifications. We can proceed through two different strategies:

- **Open-loop shaping**
- **Closed-loop shaping**

1.1.1 Open-loop shaping

Given the specifications, we translate the specifications from the closed-loop to the open-loop specifications and we define a desired behaviour for the **loop-function** $\boldsymbol{L(s)}$. Thus, we design a ***controller*** in such way that the **loop-function** has a given desired shape in order to guarantee some **dynamic properties** and **steady-state** specifications. We call this procedure **open-loop shaping** since we are looking the open-loop system.

NOTE: this technique need, eventually, to re-iterate.

1.1.2 Closed-loop shaping

For the same reason seen above, when we directly focus on the **closed-loop** system, we talk about of **closed-loop shaping**.
This procedure is the common one, since closed-loop specifications are usually given, thus, we can directly use without translation to the open-loop ones. For this purpose, we focus on the:

- **Sensitivity function:** $S(s)$;
- **Complementary sensitivity function:** $T(s)$;
- **Control sensitivity function**: $S_u(s)$.

To define the above functions, let's consider the ***classic feedback control scheme***:

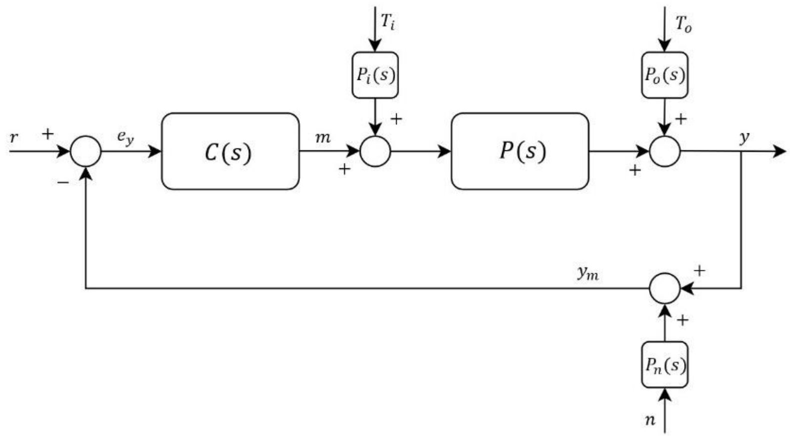

Figure 1.1: *Feedback Control Scheme*

The goal is to find a controller such that it solves some specifications. These specifications are:

- **Asymptotic stability** of the closed-loop system.
- **Reference tracking** (the output follows as too as possible the reference which can be simul as desired behaviour)
- **Disturbance attenuation**
- **Noise rejection**

The system output is:

$$y(s) = P_o(s)d_o(s) + P(s)P_i(s)d_i(s) + P(s)C(s)r(s) - P(s)C(s)n(s) - P(s)C(s)y(s)$$

and this implies that:

$$[1 + P(s)C(s)]y(s) = P(s)C(s)r(s) - P(s)C(s)n(s) + P_o(s)d_o(s) + P(s)P_i(s)d_i(s)$$

Defining the **sensitivity function** and the **complementary sensitivity function**, respectively, the following functions:

$$S(s) \triangleq (1 + P(s)C(s))^{-1}$$

$$T(s) \triangleq (1 + P(s)C(s))^{-1} \cdot P(s)C(s)$$

the output of the system, becomes:

$$y(s) = S(s)[\, P_o(s)d_o(s) + P(s)P_i(s)d_i(s)] + T(s)[r(s) - n(s)]$$

Let's consider the **tracking error** which is defined as $e_y = r - y$. Substituting in this relation the output of the system, we obtain:

$$\mathrm{e_y(s) = S(s)(r(s) - P_o(s)d_o(s) - P(s)P_i(s)d_i(}s\mathrm{)) + T(s)n(s)}$$

Through these last two relationships, we are able to understand that the magnitude of the sensitivity function, $|S(j\omega)|$, should be small in order:

- to obtain **good tracking behaviour**;
- to reject the **output disturbance** $(\mathrm{P_o(s)d_o(s)})$;
- to reject **input disturbance** $(\mathrm{P_i(s)d_i(s)})$.

Moreover, we add that $|T(j\omega)|$ should be small in order to reject measurement noise $n(s)$ in $y(s)$ or $e_y(s)$.

BE CAREFUL: we cannot make $|T(j\omega)|$ and $|S(j\omega)|$ small simultaneously at the same frequency.

Another quantity, that we consider, is the **control sensitivity function** which is defined as follow:

$$\mathrm{S_u(s)} \triangleq \left(1 + \mathrm{C(s)P(s)}\right)^{-1}\mathrm{C(s)}$$

In the following figures are depicted the typical shape of the **sensitivity function** and **complementary sensitivity function**.

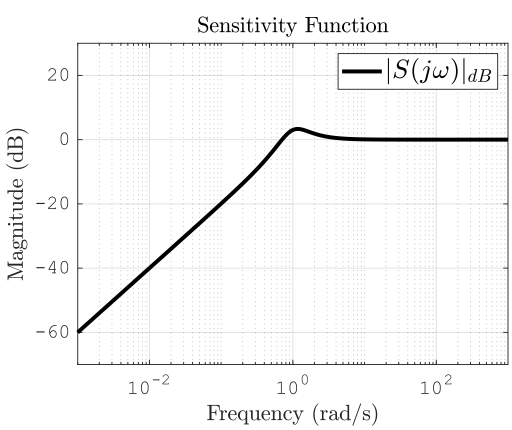

***Figure 1.2:** Shape of sensitivity function*

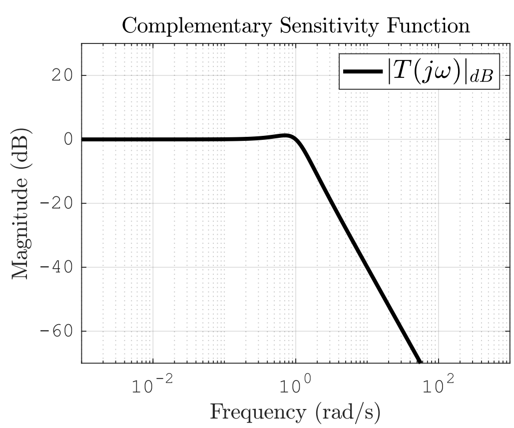

***Figure 1.3:** Shape of complementary sensitivity function*

We see that $S(s)$ behaves as **High-Pass Filter** while $T(s)$ behaves as **Low-Pass Filter**, this means that $S(s)$ is be able to attenuate signals with low frequency range, instead $T(s)$ is be able to attenuate signals with high frequency range.

What is the main advantage in considering $S(s)$?

The main advantage of considering $S(s)$, is that, because we ideally want $S(s)$ small, it is sufficient to consider its magnitude $|S(j\omega)|$. Typical specifications in terms of $S(s)$ include:

- bound of M_S^{max};
- bound on $\alpha(\omega)$ in order to have attenuation with respect to **reference**, to the **tracking error**, and with respect to the **input** and **output disturbance**.

These specifications, mathematically, may be captured by an upped bound $\frac{1}{|w_S(j\omega)|}$, on the magnitude of $S(s)$ where $w_S(s)$ is the **sensitivity weight**. With this in mind, the goal becomes:
Find a controller $C(s)$, among all stabilizing controllers, such that:

$$|\mathrm{S}(\mathrm{j}\omega)| \leq \frac{1}{|\mathrm{w_s}(\mathrm{j}\omega)|}, \forall\omega$$

and this implies that:

$$|\mathrm{S}(\mathrm{j}\omega)\mathrm{w_s}(\mathrm{j}\omega)| \leq 1, \forall\omega$$

which is equivalent to say that:

$$\left\|\mathrm{w_s}(\mathrm{s})\mathrm{S}(\mathrm{s})\right\|_\infty \leq 1$$

Thus, the goal can be re-written as follow:

Find $C(s)$, **among all stabilizing controllers**, such that $||\mathrm{w_S}(\mathrm{s})\mathrm{S}(\mathrm{s})||_\infty \leq 1 = \gamma$

A **typical shape** of $w_S(s)$ is:

$$\mathrm{w_S}(\mathrm{s}) = \frac{\frac{\mathrm{s}}{\mathrm{M_S^{max}}} + \mathrm{B_{3S}^{min}}}{\mathrm{s} + \mathrm{B_{3S}^{min}}\mathrm{A_S}}$$

Similar considerations may be made for $T(s)$. A **typical shape** of $w_T(s)$ is:

$$w_T(s) = \frac{s + \frac{B_{3T}^{max}}{M_T}}{A_T\, s + B_{3T}^{max}}$$

We have seen weights for sensitivity and complementary sensitivity function. Now, to take account of controller effort, we need a weight in the **control sensitivity function**. A **typical shape** of $w_U(s)$ is:

$$w_U(s) = Constant$$

Actually, in order to limit the actuator bandwidth, let's define the weighted control sensitivity function, as follow:

$$w_U(s) = \frac{k(s + B_{3L})}{A_S s + B_{3L}}$$

Now, defining:

$$N(s) = \begin{bmatrix} w_S(s)S(s) \\ w_T(s)T(s) \\ w_u(s)S_u(s) \end{bmatrix} \longleftarrow \textbf{stacked weighted sensitivity}$$

The goal becomes: **find a controller $C(s)$, among all stabilizing controllers, that minimize $||N(s)||_\infty$**.

The following diagrams show the typical shapes for the **weighted functions** defined above.

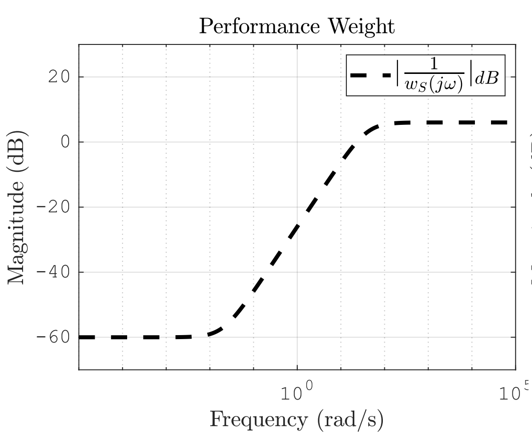

Figure 1.4: *Weighted sensitivity function*

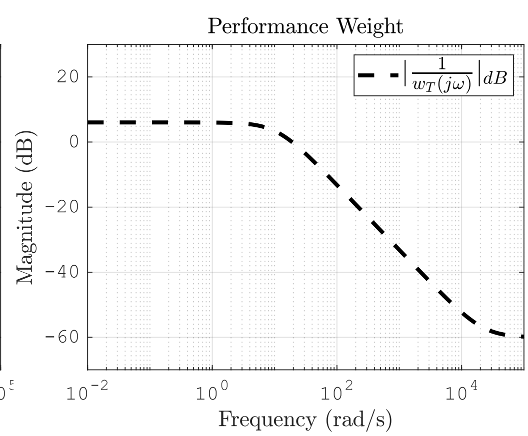

Figure 1.5: *Weighted complementary sensitivity function*

2 DC-Motor Velocity Control Problem

In this section, we want to solve the **velocity control problem** of a **DC-Motor**. In particular, we will focus on the $\mathcal{H}_\infty$ **controller design** which will have to be able to drive the rotation of the rotor of **DC-Motor** at steady state, at a given angular velocity. Moreover, we will consider three case studies and, in each case, we will compare the results, obtained in the case under discussion, with the previous results and at the end we will choose the best controller among them and we will compare its results with the results obtained thought PI controller.

The three case studies are:

- ***Controller design*** which take into to account only the weighted sensitivity $\boldsymbol{w_S}$;
- ***Controller design*** which take into to account both the weighted sensitivity $\boldsymbol{w_s}$ and the weighted control sensitivity $\boldsymbol{w_U}$;
- ***Controller design*** which take into to account all the weighted sensitivities: $\boldsymbol{w_S}$, $\boldsymbol{w_U}$, $\boldsymbol{w_T}$.

Firstly, let's consider the linear model of the DC-Motor shown in *Fig. 2.1*.

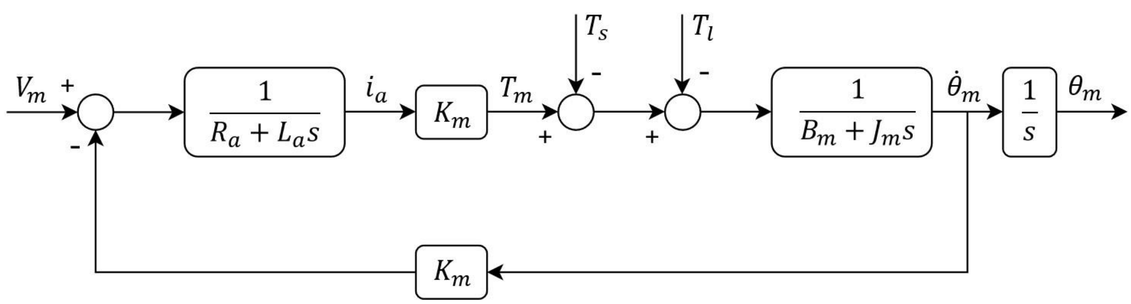

Figure 2.1: *DC-Motor linear model*

where V_m stands for **armature voltage**, T_l stands for **torque load**, R_a stands for **armature resistance**, L_a stands for **armature inductance**, K_m stands for **motor torque constant**, B_m stands for **friction coefficient**, J_m stands for **mechanical inertia**, $\dot{\theta}_m$ stands for **angular velocity** and θ_m stands for **angular position**.

2.1 Motor's parameters

For the controller design, the following motor's parameters[1] are chosen:

Parameter	Value
$\boldsymbol{R_a}$	$0.01485\ \Omega$
$\boldsymbol{L_a}$	$0.01485\ H$
$\boldsymbol{K_m}$	$0.329\ \frac{N \cdot m}{A}$
$\boldsymbol{B_m}$	$0.1\ N \cdot m \cdot s$
$\boldsymbol{J_m}$	$0.5\ Kg \cdot m^2$

Table 1: *Motor's Parameters*

2.2 Transfer Function representation

To describe the DC-Motor model, we use the transfer function representation.
The transfer function between the **output**, in this case $\dot{\theta}_m$ since we want to control the angular velocity, and the **input** V_m is:

$$\mathrm{P(s)} = \frac{\dot{\theta}_\mathrm{m}}{\mathrm{V_m}} = \frac{\mathrm{K_m}}{(\mathrm{R_a} + \mathrm{L_a s})(\mathrm{B_m} + \mathrm{J_m s}) + \mathrm{K_m^2}}$$

Instead, the transfer function between $\dot{\theta}_m$ and the load disturbance T_l is:

$$P_d(\mathrm{s}) = \frac{\dot{\theta}_\mathrm{m}}{\mathrm{T_l}} = -\frac{(\mathrm{R_a} + \mathrm{L_a s})}{(\mathrm{R_a} + \mathrm{L_a s})(\mathrm{B_m} + \mathrm{J_m s}) + \mathrm{K_m^2}}$$

Substituting the DC-Motor's parameters on $\mathrm{P(s)}$ and on $P_d(s)$, we obtain:

$$P(s) = \frac{470}{(s^2 + 10.81s + 156.8)}$$

$$P_d(s) = -\frac{2(s + 10.61)}{(s^2 + 10.81s + 156.8)}$$

[1] https://www.researchgate.net/figure/Parameter-values-for-Series-DC-motor_tbl2_316471897

Remark: What does it happen if we calculate the open loop response of the DC-Motor when we apply a step input with amplitude 1500? From the final value theorem we obtain that, at steady-state, the output is equal to:

$$y(t) = \lim_{s\to 0} s \cdot y(s) = \lim_{s\to 0} s \frac{1500}{s} P(s) = P(0) = 4496$$

As we see in *Fig.2.2*, the output is not able to follow the reference, hence we need to use the feedback control scheme.

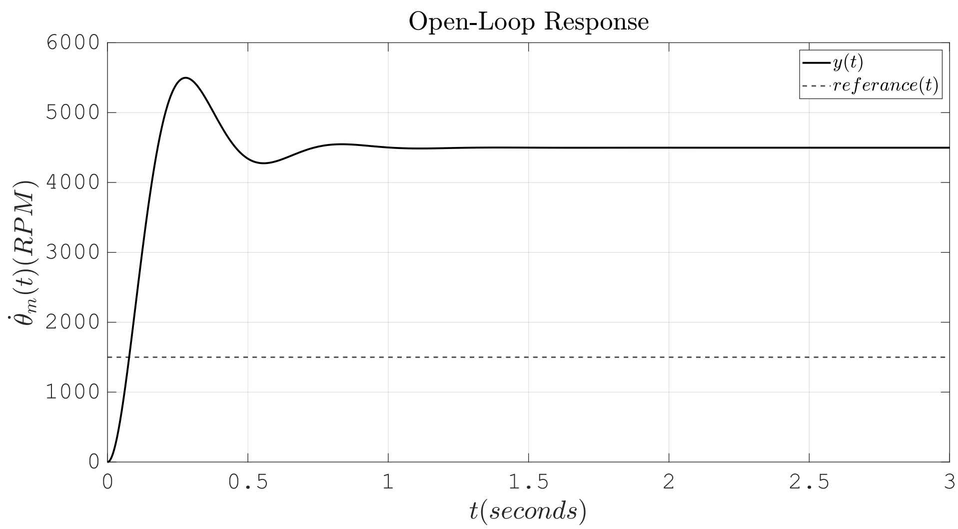

Figure 2.2: *Step Response of the open-loop system*

2.3 $\mathcal{H}_\infty$ controller design which minimize only $||w_S(s)S(s)||_\infty$

We are looking a controller, among all stabilizing controllers, such that $||w_S(s)S(s)||_\infty \leq \gamma$. To construct the weighted sensitivity function, we have to choose the bandwidth B_{3S}, the peak M_s, which represents the upper-bound, and the lower-bound A.

Let's assume the following choices:

- $B_{3S} = 20\frac{\text{rad}}{\text{s}}$, in order to reject the disturbances with frequency content up to $20\frac{rad}{s}$
- $M_S = 2$
- $A_S = 0.001$

Thus, our weighted function is equal to:

$$w_S(s) = \frac{s + 40}{2\,s + 0.04}$$

By this choice and thanks to MATLAB.*Command* `mixsyn`, we find a controller that minimize the $\mathcal{H}_\infty$ norm of interest; in particular we find that:

$$||w_S(s)S(s)||_\infty \leq 0.5036.$$

In *Fig. 2.3* we see the shape of the sensitivity functions and the open-loop function.

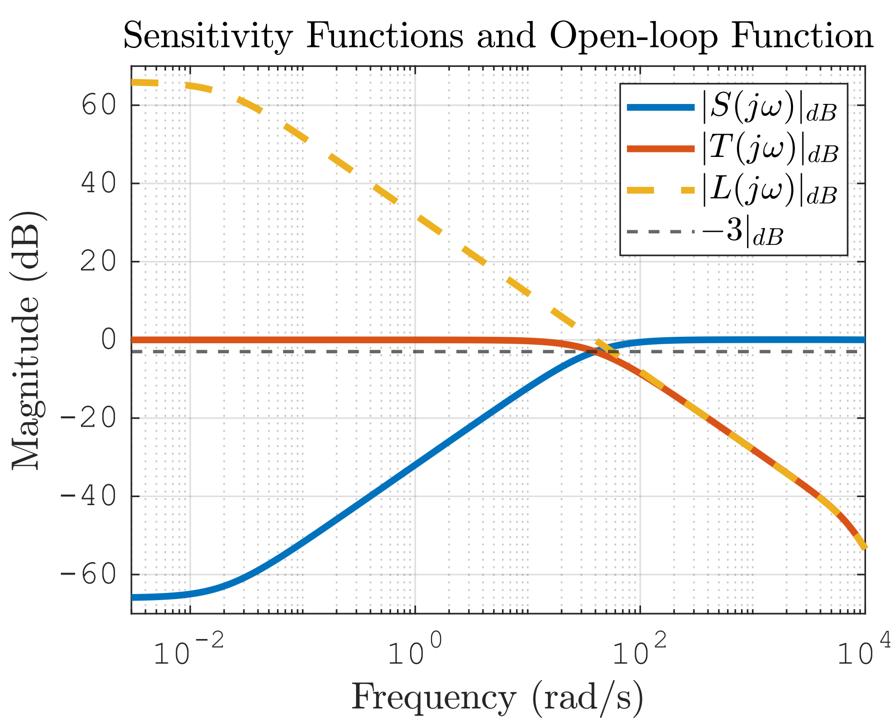

Figure 2.3: *Plot of the sensitivity functions and open-loop function when we use the $\mathcal{H}_\infty$ controller that minimize $||w_S(s)S(s)||_\infty$*

In the *Fig. 2.4* the plot of the **sensitivity function** and the **performance weight** are depicted. We see the red line is below of the black piecewise line and this means that the value of the $\mathcal{H}_\infty$ norm is less than 1 and so, the sensitivity function satisfies the specifications relatively to the bound on M_S^{max} and to the bound on $\alpha(\omega)$, in order to attenuate the magnitude of the disturbance at low frequency range.

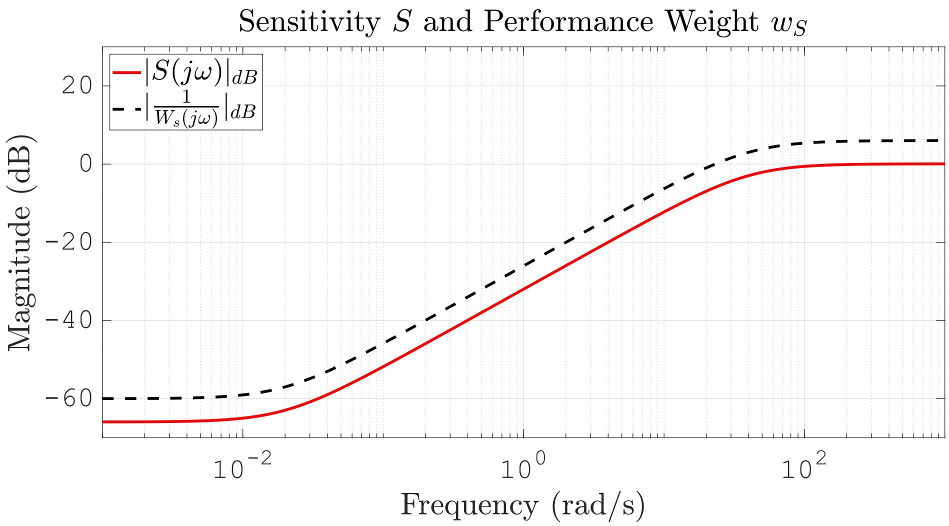

Figure 2.4*: Sensitivity function and weighted sensitivity function*

Compared to the open-loop case, is this feedback control scheme capable of causing the output to follow the input? The answer is clearly yes. To see this, let's use **Simulink**[2] in order to plot the output response, the disturbance response and the control action, which are shown in *Fig.2.6*.
The Simulink block-scheme of this homework is presented in *Fig. 2.5*.

As input, we have considered a:

- ***step signal***, for the **input reference**, defined as follow:

$$\delta_{-1}(t) = \begin{cases} 0, & t < 0 \\ 1500, & t \geq 1 \end{cases}$$

- ***sinusoidal signal***, for the **disturbance**, defined as follow:

$$T_l = 400\sin\,(0.1\,t)$$

- ***step signal***, for the **disturbance**, defined as follow:

$$\delta_{-1}(t) = \begin{cases} 0, & t < 0 \\ 400, & t \geq 1 \end{cases}$$

[2] **Simulink** is a MATLAB - based graphical programming environment for modelling, simulating and analysing multidomain dynamical systems. It offers tight integration with the rest of the MATLAB environment and can either drive MATLAB or be scripted from it. Simulink is widely used in **automatic control** and **digital signal processing** and so on.

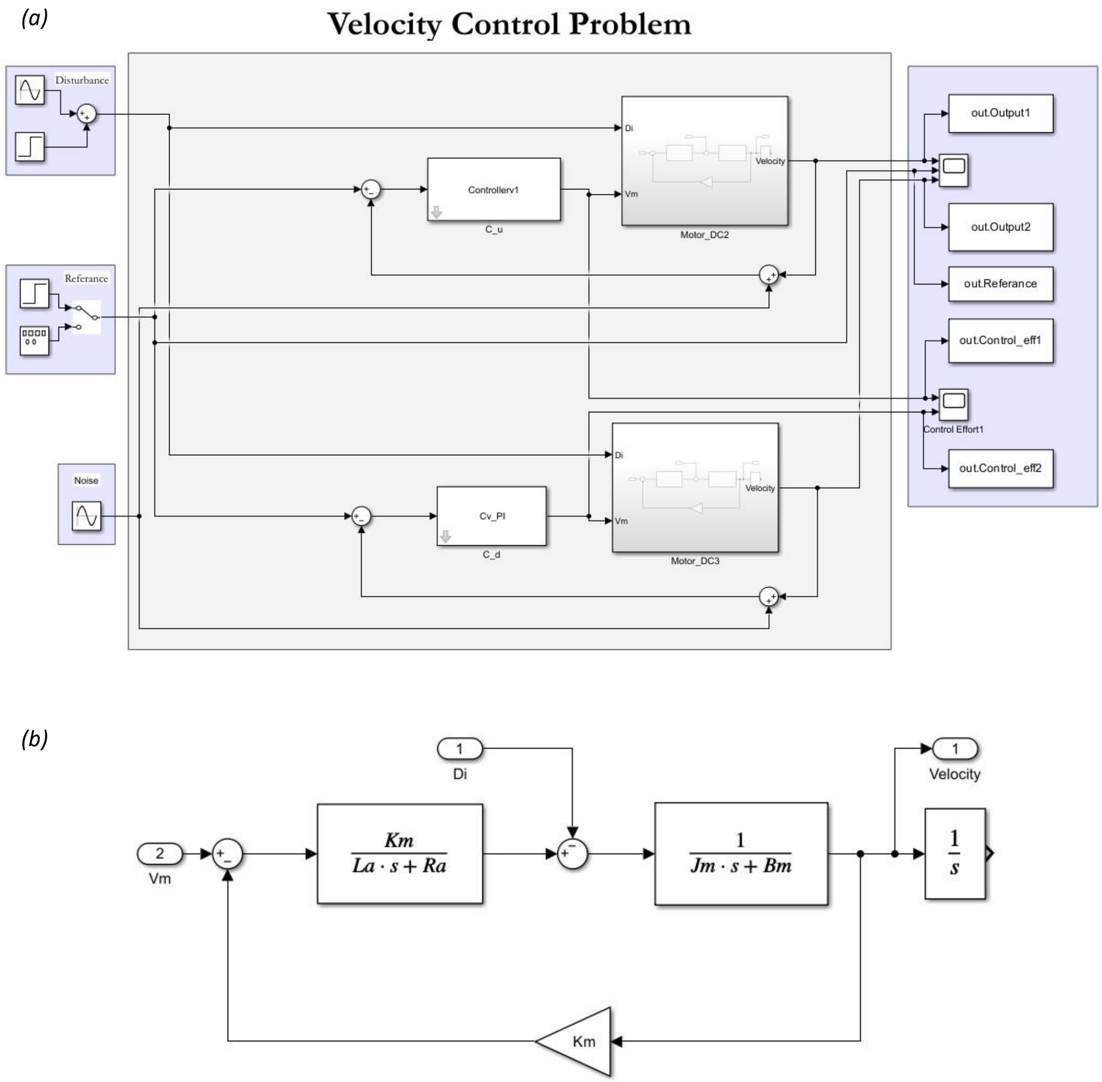

Figure 2.5: *(a) block-scheme of the control system. (b) block-scheme of the DC-motor model*

We note from, $Fig.2.6$ (c)-(d), that the effect of the disturbance T_l **is attenuated** quite well. Regarding the **control effort**, $Fig.2.6$ (b), however, we note that the system, in order to attenuates the disturbance, introduces a very accentuated control effort.

To minimize the **control energy** we include the term $w_u(s)S_u(s)$, in the $\mathcal{H}_\infty$ norm of the weighted mixed sensitivity, because this term allows to "penalize" the control action $u(t)$ so that the control effort decreases.

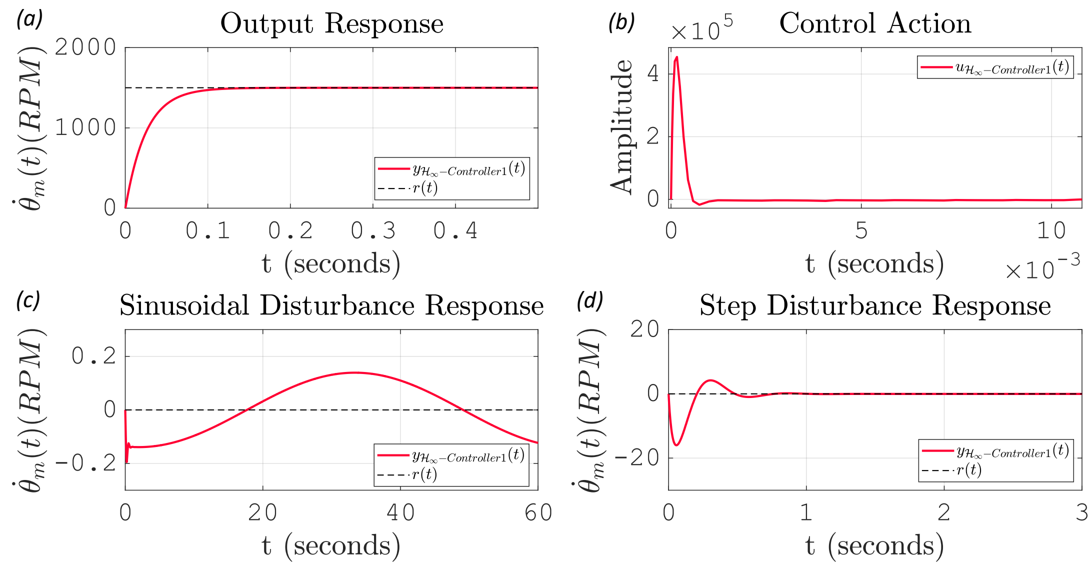

Figure 2.6: *(a) Output response. (b) Control action. (c) Sinusoidal disturbance response. (d) Step disturbance response*

2.4 $\mathcal{H}_\infty$ controller design which minimize $\left\|\begin{matrix} w_S(s)S(s) \\ w_U(s)S_u(s)\end{matrix}\right\|_\infty$

We are looking a controller, among all stabilizing controllers, such that:

$$||N(s)||_\infty = \left[\left\|\begin{matrix} w_S(s)S(s) \\ w_U(s)S_u(s)\end{matrix}\right\|_\infty\right] \leq \gamma$$

It is clear that $w_S(s)$ not change, therefore, we will have to choose $w_u(s)$. We define the weighted control sensitivity function as follow:

$$w_U(s) = \frac{k(s + B_{3L})}{A_S s + B_{3L}}$$

We decide to use the following choices:

- $\mathrm{B_{3L}} = 400\,\frac{\mathrm{rad}}{\mathrm{s}}$
- $\mathrm{k} = 0.1$
- $A_S = 0.001$

Thus, our weighted control sensitivity function is equal to:

$$w_U(s) = \frac{0.1s + 40}{0.01\,s + 400}$$

By these choices, we find a controller that minimize the $\mathcal{H}_\infty$ norm; in particular we find that:

$$\left[\left\| \begin{matrix} w_S(s)S(s) \\ w_U(s)S_u(s) \end{matrix} \right\|_\infty\right] \leq 0.7945.$$

In *Fig*.2.7 is shown the comparison of the sensitivity function obtained by first controller and the sensitivity function obtained by the second controller, while in *Fig*.2.8 is shown the comparison of the control sensitivity function obtained by the first controller and the control sensitivity function obtained by the second controller.

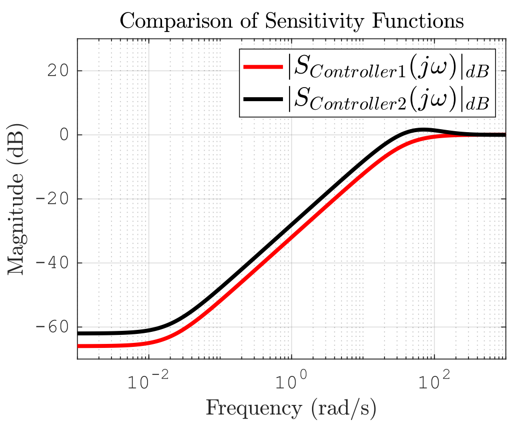

Figure 2.7: *Comparison of the sensitivity functions*

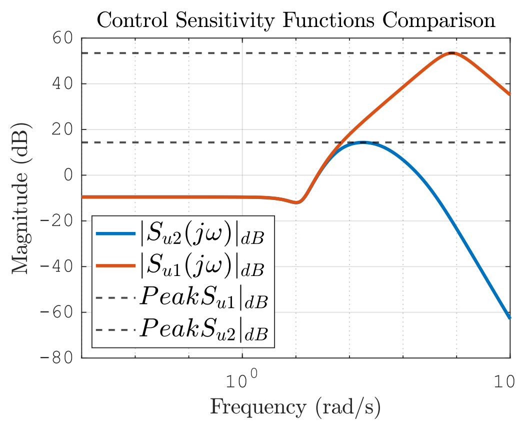

Figure 2.8: *Comparison of the control sensitivity functions*

From *Fig*.2.7 we see that the new controller is not able **to reject** the disturbance better than the previous controller. However, through the introduction of this new controller, we obtain a big advantage that is shown in *Fig*. 2.8, that the peak of the control sensitivity function, obtained by the new controller, is much smaller than the peak of the control sensitivity function, obtained by the previous controller.
How does this translate in terms of control effort? Let's look the *Fig*.2.9(b).

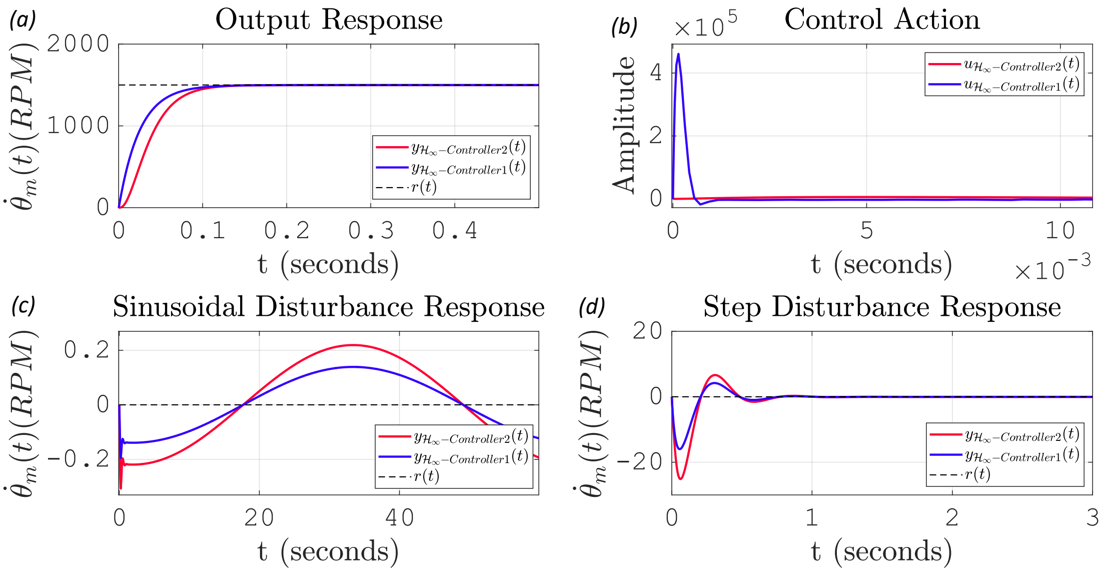

***Figure 2.9:** (a) Output response. (b) Control action. (c) Sinusoidal disturbance response. (d) Step disturbance response*

We see that the control effort, obtained with the addition of the weighted control sensitivity function, is smaller than the previous control effort, see $Fig.2.9$(b), obtained by the previous controller.
Moreover, as we can see from $Fig.2.9$ (c)-(d) and as mentioned before, the new controller rejects less both sinusoidal and step disturbance and, clearly, having a smaller bandwidth B_{3S} than before, the system is slightly slower than the previous one, see Fig.2.9 (a).

Up to now, we have not considered the case in which there could be the presence of a measurement noise at high-frequency content. If we introduce them, for example if we consider as a **measurement noise**, a ***sinusoidal signal*** defined as follow:

$$n = 500 \sin(1000t)$$

we obtain the following output measurements when we consider the first controller that we have found:

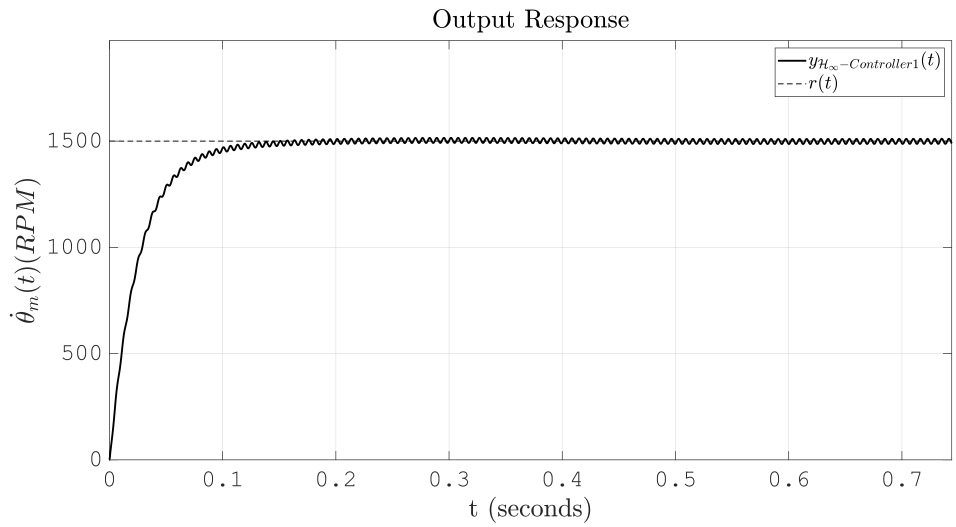

Figure 2.10: *Output response affected by measurement noise*

To solve this problem, we include also the term $w_T(s)T(s)$ in the H_∞ norm of the weighted mixed sensitivity.

2.5 $\mathcal{H}_\infty$ controller design which minimize $\left\|\begin{matrix} w_S(s)S(s) \\ w_U(s)S_u(s) \\ w_T(s)T(s) \end{matrix}\right\|_\infty$

We are looking a controller, among all stabilizing controllers, such that:

$$\|N(s)\|_\infty = \left[\left\|\begin{matrix} w_S(s)S(s) \\ w_U(s)S_u(s) \\ w_T(s)T(s) \end{matrix}\right\|_\infty\right] \le \gamma$$

We have to choose the weighted complementary sensitivity function and so, we have to choose the bandwidth $\mathrm{B_{3T}}$, the peak $\mathrm{M_T}$ which represents the upper-bound, and the lower-bound $\mathrm{A_T}$.

We use the following choices:

- $\mathrm{B_{3S}} = 22\,\frac{\mathrm{rad}}{\mathrm{s}}$
- $\mathrm{M_S} = 2$
- $A = 0.001$

Thus, our weighted function is equal to:

$$w_T(s) = \frac{s + 11}{0.001\, s + 22}$$

By these choices, we find a controller that minimize the $\mathcal{H}_\infty$ norm; in particular we find that:

$$\left[\left\| \begin{matrix} w_S(s)S(s) \\ w_U(s)S_u(s) \\ w_T(s)T(s) \end{matrix} \right\|_\infty\right] \leq 1.1532$$

In the *Fig.* 2.11 is shown the plot of the **complementary sensitivity function** and the **performance weight**. We see the red line is below of the black piecewise line and this means that the complementary sensitivity function satisfies the specifications relatively to the bound on M_T^{max} and to the bound on $\beta(\omega)$, in order to attenuate the magnitude of the measurement noises at high frequency range.

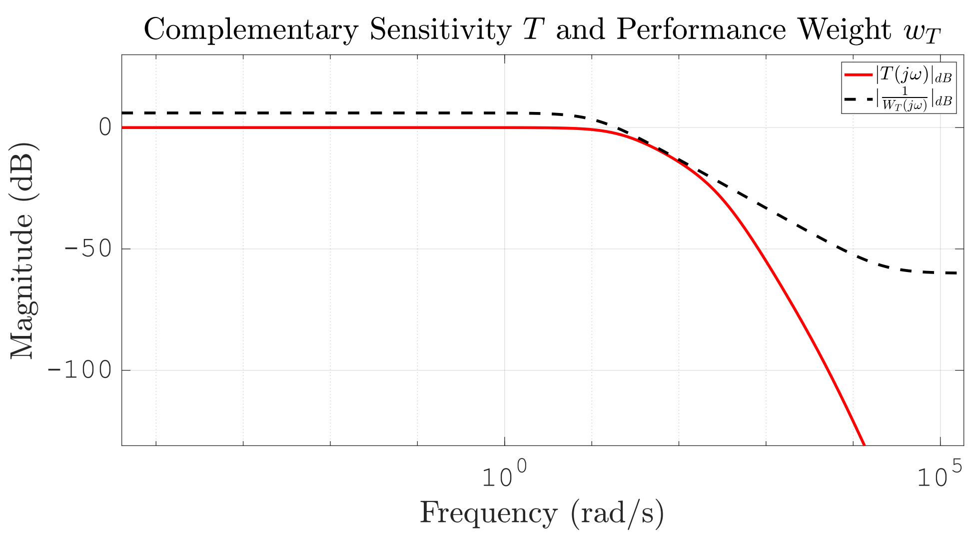

***Figure 2.11**: Complementary sensitivity function and weighted complementary sensitivity function*

In *Fig.*2.12 is shown the comparison of the complementary sensitivity function, obtained by the first controller, and the complementary sensitivity function, obtained by the third controller. We see that the red line is steeper than the black line and this means that the third controller is able to attenuate the measurement noises more than the first controller and this due to the addition of the weighted complementary sensitivity function.

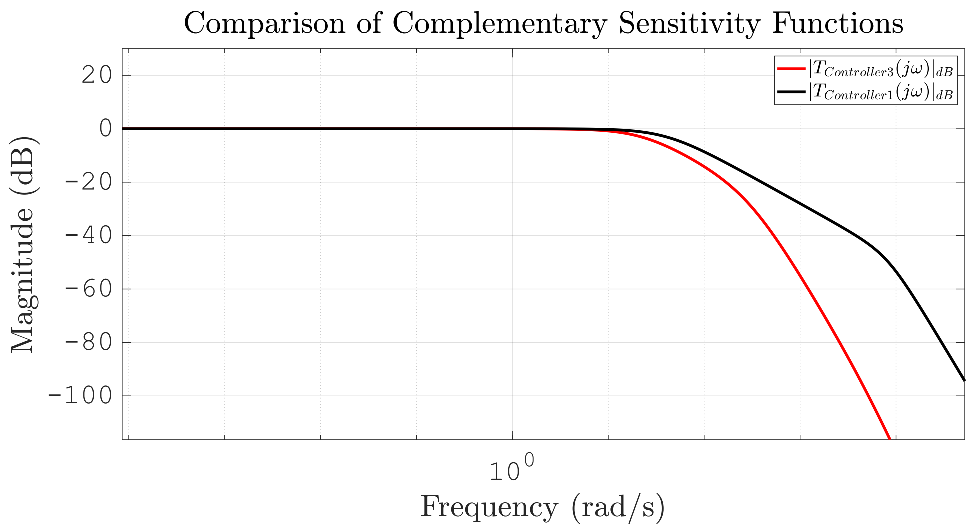

***Figure 2.12**: Comparison of the complementary sensitivity functions*

In *Fig.2.13* are shown the results in terms of output response, control action and disturbance response between, the performance obtained by the third controller and the performance obtained by the first controller.

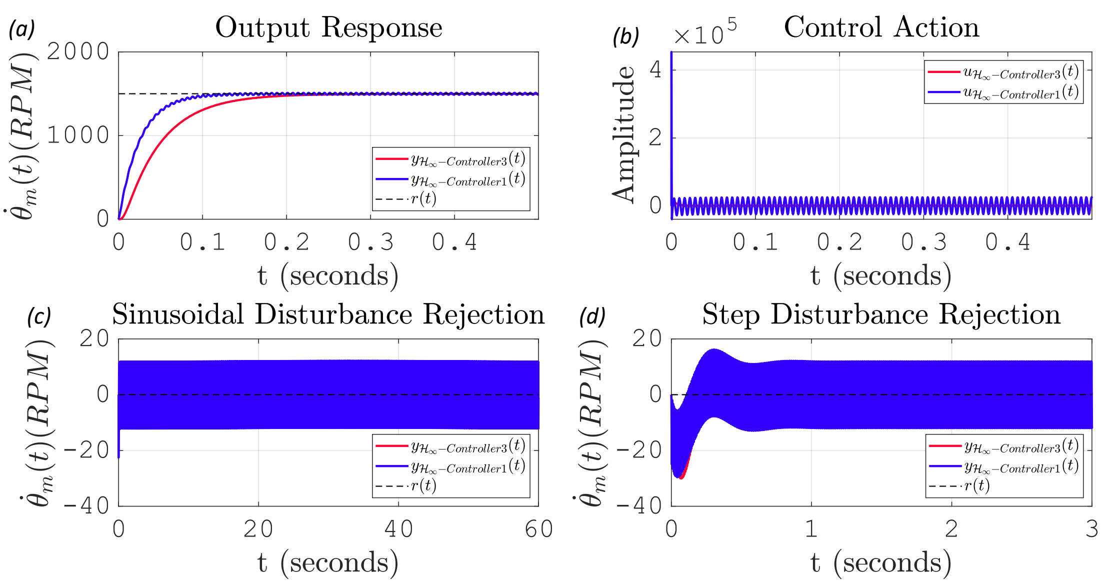

***Figure 2.13:** (a) Output response. (b) Control action. (c) Sinusoidal disturbance response. (d) Step disturbance response*

Instead, what can we say about the comparison between the third and second controller? Let's see the *Fig.*2.14.

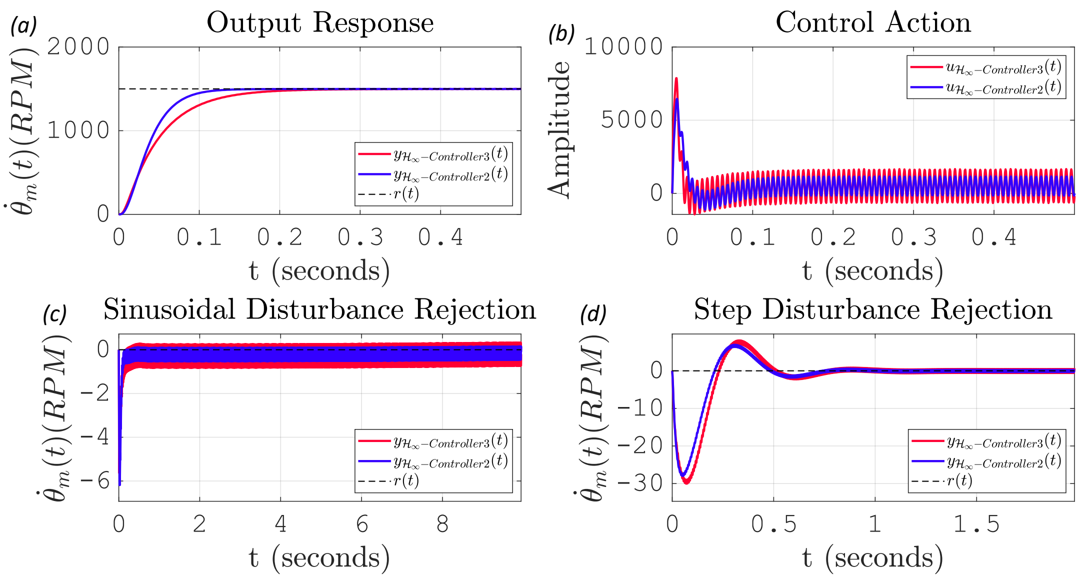

Figure 2.14: *(a) Output response. (b) Control action. (c) Sinusoidal disturbance response. (d) Step disturbance response*

From the *Fig.*2.14 (b), we note that the addition of the weighted complementary sensitivity function has incremented the control effort, but luckily not by much. Moreover, from *Fig.*2.14 (a)-(c)-(d), we see that the performance of the two system are very similar, although the second controller is slightly better.

2.6 *PI* controller

At this point we compare the performances, related to the **step response**, the **rejection of the disturbance** and the **control effort**, between the best controller, which is the second, and the PI controller, that we remember is defined as follow:

$$C(s) = K_P + K_I \frac{1}{s}$$

We obtain the parameters K_P and K_I through the `pidtune` function of MATLAB, which returns the follow values: $K_P = 0$ and $K_I = 1.24$.

In *Fig.*2.15, is shown the comparison of the sensitivity function obtained by second controller and the sensitivity function obtained by the PI controller.

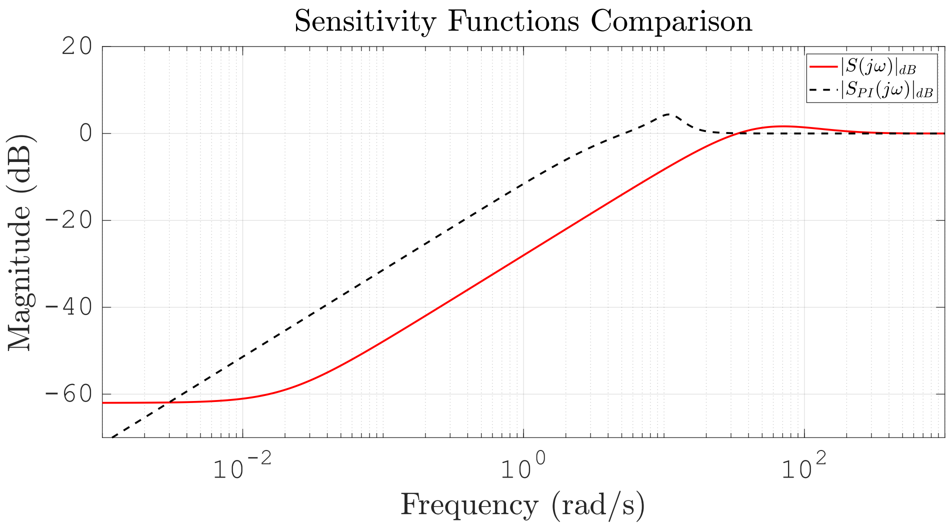

Figure 2.15: *Comparison of the sensitivity functions*

We see that the sensitivity function of the $\mathcal{H}_\infty$controller is able **to reject** the disturbance better than the PI controller since is steeper.

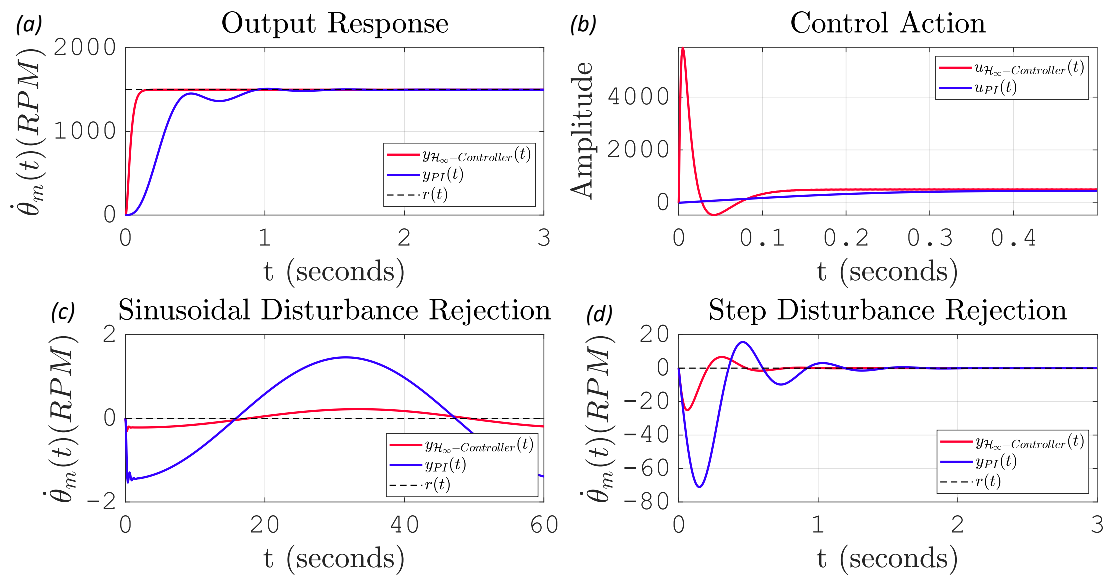

Figure 2.16: *(a) Output response. (b) Control action. (c) Sinusoidal disturbance response. (d) Step disturbance response*

In *Fig.*2.16 are shown the comparison in terms of output response, control action and disturbance response, and in particular, we note that the $\mathcal{H}_\infty$ controller reaches steady state faster than the PI controller. Moreover, the PI controller has a lower capability to reject disturbances, both sinusoidal and step, that the $\mathcal{H}_\infty$ controller.
However, the main advantage of the PI controller is that, it has a smaller control effort than the $\mathcal{H}_\infty$ controller.

2.7 Conclusion

On the bases of the three cases that we have considered and on the comparison between them, which were designed with the minimization of the $\mathcal{H}_\infty$ norm, and the PI controller we can say that, although the latter provides a slower step response, compared with the $\mathcal{H}_\infty$ controllers, its performance is fairly good with respect to the disturbance rejection and measurement noise. Regarding the three $\mathcal{H}_\infty$ controllers, which we have considered, we can conclude by saying that the second controller analysed is the best one, not only because is able to reject the disturbances and the measurements noise, but because it gives us a very low control effort with respect to the control effort given by the third and the first controller (see Fig.2.9 (b) and Fig. 2.14 (b)).

2.8 Consideration on the disturbance rejection

If we consider, now, the *Fig.*2.16 (a) and we expand the temporal window, we obtain the *Fig.*2.17.

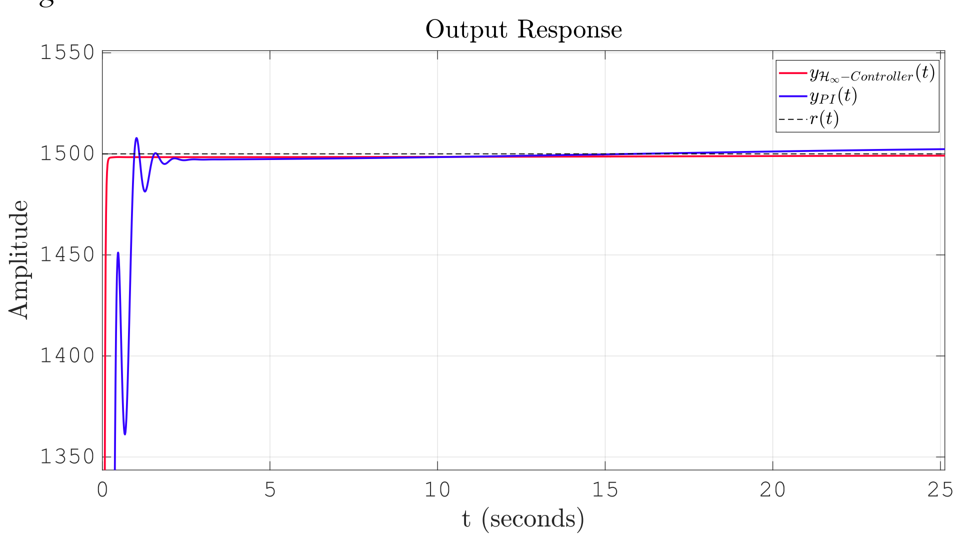

Figure 2.17: *Output response*

Why does this happen? Because the $\mathcal{H}_\infty$ controller does not minimize the norm:

$$\left\| w_S(s) P_d(s) S(s) \right\|_\infty$$

In fact:

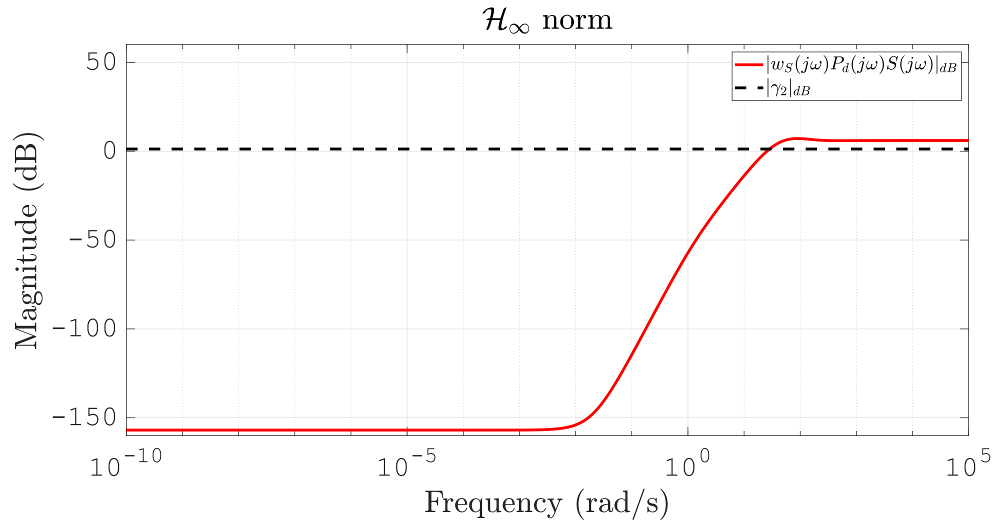

Figure 2.18: *Magnitude of* $\left\| w_S(s) P_d(s) S(s) \right\|_\infty$

3. DC-Motor Velocity Control Problem through Extended Plant

The aim of this section to solve the problem of the disturbance rejection, since our previous controller was not effective with respect to the sinusoidal disturbances.

To solve this, we have to build the **extented plant**. We can represente the general control scheme as follow:

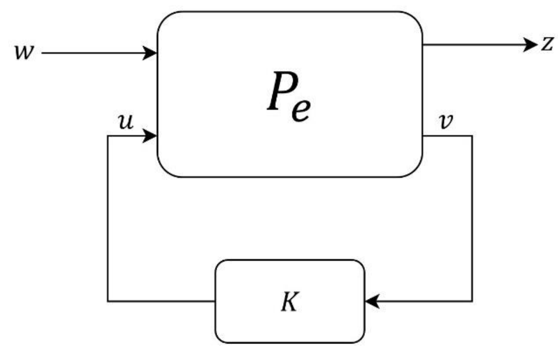

Figure 2.1: *General control scheme*

where:

- ***w*** are the **exogeneous variables** where, with the term exogeneous, we refer to external inputs. From this, we understand that the exogeneous variables are the **reference**, the **disturbance**, **measurements noise**.
- ***z*** defines the **performance variables**. We measure how well the control scheme is performing. It is clear that this depend on our interests. For example we could consider:
  - The output: y
  - The tracking error: $r - y$
  - The control input: u
- P_e is the extended plant
- K is the controller.

With this general control scheme, *Fig.*2.1, we find a controller able to generate a suitable control signal, such that, thanks to its input v, minimizes the $\mathcal{H}_\infty$ norm of interest.
For our control problem, let's make the following choices:

- $w = \begin{bmatrix} r \\ T_l \end{bmatrix}$
- $z = \begin{bmatrix} e_y \\ u \end{bmatrix}$

With these choices, the general control scheme becomes:

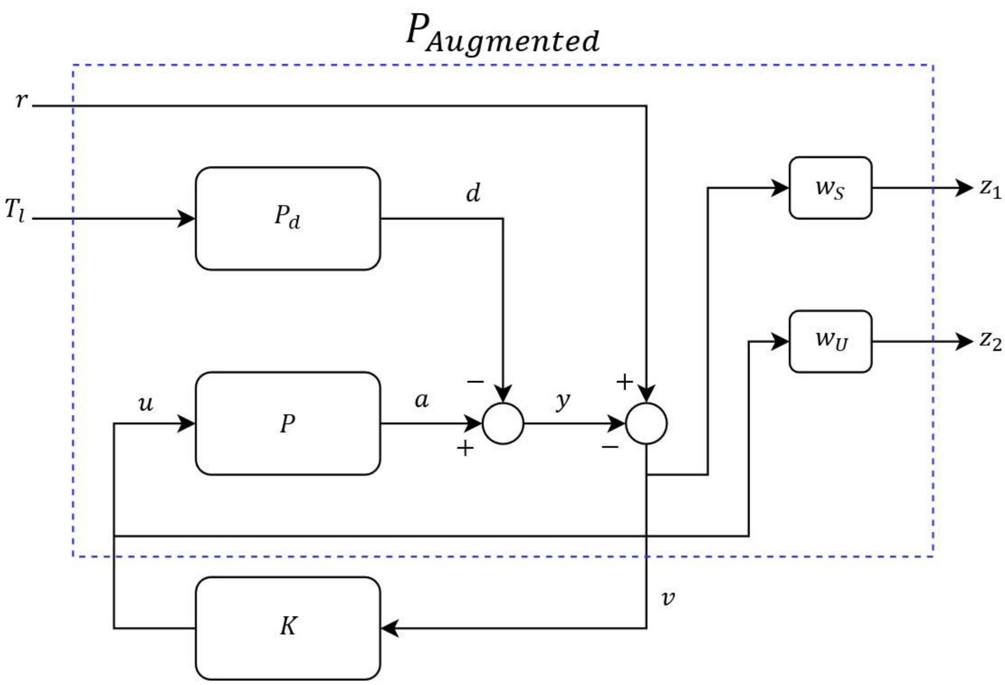

Figure 2.2*: Extended plant of the DC-Motor*

Where, we remember that:

- $\mathrm{P(s)} = \frac{\dot{\theta}_m}{V_m} = \frac{K_m}{(R_a+L_a s)(B_m+J_m s)+K_m^2} = \frac{470}{(s^2+10.81s+156.8)}$
- $P_d(\mathrm{s}) = \frac{\dot{\theta}_m}{T_l} = -\frac{(R_a+L_a s)}{(R_a+L_a s)(B_m+J_m s)+K_m^2} = -\frac{2(s+10.61)}{(s^2+10.81s+156.8)}$
- $w_S(s) = \frac{s+40}{2\,s+0.04}$
- $w_U(s) = \frac{0.1s+40}{0.01\,s+400}$

By these choices and thanks to MATLAB.*Command* `hinfsyn`, we find a controller that minimize the $\mathcal{H}_\infty$ norm; in particular we find that:

$$\left\| \begin{matrix} w_S(s)S(s) \\ w_s(s)P_d(s)S(s) \\ w_U(s)S_u(s) \end{matrix} \right\|_\infty \leq 0.7974$$

In the *Fig. 2.3* are shown the plot of the **sensitivity function** and the **performance weight $\boldsymbol{w_S(s)}$**, and the plot of the **control sensitivity function** and the **performance weight $\boldsymbol{w_U(s)}$**. We see that, in both plots, the red line is below of the black piecewise line and this means that the sensitivity function and the control sensitivity function satisfy their specifications.

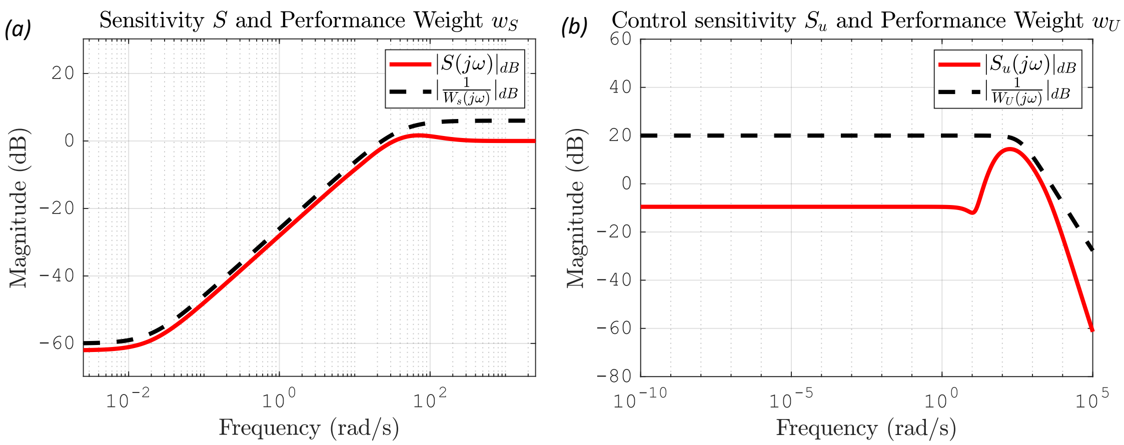

***Figure 2.3**:(a) Sensitivity function and weighted sensitivity function. (b) Control sensitivity function and weighted control sensitivity function*

From *Fig.* 2.4, we see that all the curves remain below the γ value and this means that the controller correctly minimizes the stacked weighted sensitivity $\mathcal{H}_\infty$ norm.

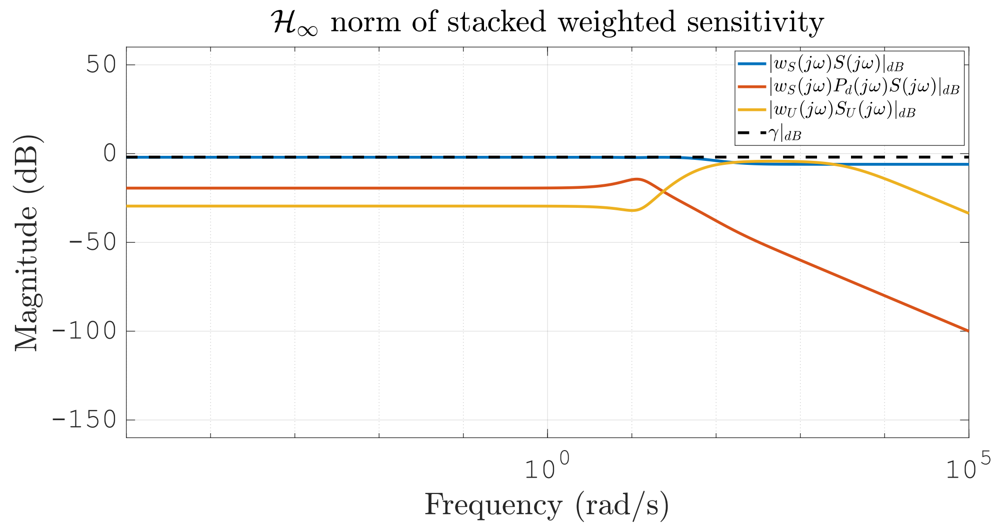

***Figure 2.4**: $\mathcal{H}_\infty$ stacked weighted sensitivity norm plot*

What can we say about the disturbance rejection? Effectively, is this last controller reject better the disturbance than the previous? To answer this question, let's see the *Fig.*2.5 which shown the output response and the disturbance response.

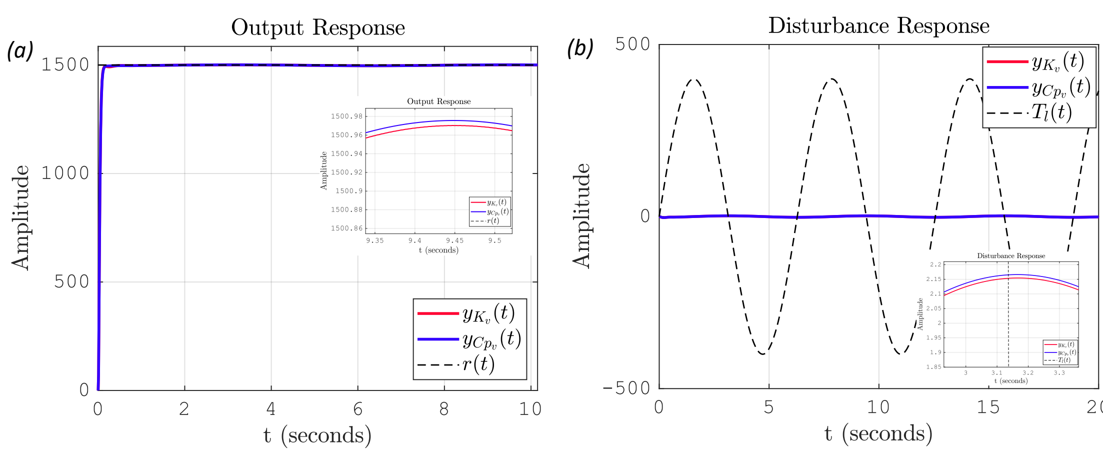

***Figure 2.5**:(a) Output response. (b) Disturbance response*

Both the controllers reject the disturbance in a very similar way, although the last controller attenuates it a little more.