



MÁSTER UNIVERSITARIO EN INGENIERÍA INDUSTRIAL

TRABAJO FIN DE MÁSTER DISEÑO E IMPLEMENTACIÓN DE CONTROLES PID DE VELOCIDAD Y SEGUIMIENTO DE PARED PARA UN TURTLEBOT3

Autor: Alejandra Tabasco Ruiz

Director: Jaime Boal Martín-Larrauri

Director: Ramón Rodríguez Pecharromán

Madrid

Julio 2023

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título
**DISEÑO E IMPLEMENTACIÓN DE CONTROLES PID DE VELOCIDAD Y
SEGUIMIENTO DE PARED PARA UN TURTLEBOT3** en la ETS de Ingeniería - ICAI
de la Universidad Pontificia Comillas en el
curso académico 2022/23 es de mi autoría, original e inédito y
no ha sido presentado con anterioridad a otros efectos.
El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido
tomada de otros documentos está debidamente referenciada.



Fdo.: Alejandra Tabasco Ruiz

Fecha: 19/07/2023

Autorizada la entrega del proyecto
EL DIRECTOR DEL PROYECTO

Fdo.: Jaime Boal Martín-Larrauri

Fecha://

Fdo.: Ramón Rodríguez Pecharromán

Fecha://



MÁSTER UNIVERSITARIO EN INGENIERÍA INDUSTRIAL

TRABAJO FIN DE MÁSTER DISEÑO E IMPLEMENTACIÓN DE CONTROLES PID DE VELOCIDAD Y SEGUIMIENTO DE PARED PARA UN TURTLEBOT3

Autor: Alejandra Tabasco Ruiz

Director: Jaime Boal Martín-Larrauri

Director: Ramón Rodríguez Pecharromán

Madrid

Julio 2023

Agradecimientos

Quería agradecer en primer lugar a mis directores, Jaime y Ramón, por su ayuda y disponibilidad durante todo el proyecto. Así como por su paciencia infinita cada vez que me atascaba en algo, por pequeño que fuese.

A mis padres y hermanos por ser siempre una fuente de apoyo y cariño cuando más lo necesito.

Por último, agradecer a todos los amigos que han hecho del paso por ICAI una experiencia inolvidable.

DISEÑO E IMPLEMENTACIÓN DE CONTROLES PID DE VELOCIDAD Y SEGUIMIENTO DE PARED PARA UN TURTLEBOT3

Autor: Tabasco Ruiz, Alejandra.

Directores: Boal Martín-Larrauri, Jaime; Rodríguez Pecharromán, Ramón.

Entidad Colaboradora: ICAI – Universidad Pontificia Comillas.

RESUMEN DEL PROYECTO

El presente proyecto tiene como objetivo general la adquisición de experiencia y dominio en el uso del TurtleBot3 Burger para prácticas de control y robótica móvil. Para ello se han diseñado e implementado unos controles PID de velocidad y seguimiento de pared, estudiando la respuesta de estos en seguimiento a referencia y rechazo a perturbación.

Palabras clave: TurtleBot3, ROS2, MATLAB, Simulink, Controles PID.

1. Introducción y definición del proyecto

Este proyecto surge como continuación de un trabajo fin de grado realizado durante el curso 2021-2022 [1]. Con este proyecto se pretende actualizar la plataforma que se usa actualmente en los laboratorios de control de distintas asignaturas de la universidad. Este trabajo fin de máster tiene como objetivo general la adquisición de experiencia y dominio en el uso del robot para prácticas de control y robótica móvil. Para ello se han diseñado de forma teórica unos controles PID de velocidad y seguimiento de pared, a partir de los modelos de las plantas obtenidos. Posteriormente se ha determinado la forma más adecuada de implementar dichos controles en el robot, teniendo en cuenta que su uso es en prácticas docentes. Por último, se han realizado ensayos con los controles diseñados, estudiando su comportamiento en seguimiento a referencia y rechazo a perturbación.

2. Descripción del sistema

En la Figura 1 se muestra un diagrama de la arquitectura de comunicaciones del sistema. Por un lado, se tiene el ordenador en el que está instalado MATLAB, desde el que se envían las distintas consignas y parámetros necesarios, además de recoger los resultados de los ensayos para su posterior monitorización. Por otro lado, está el TurtleBot3 Burger, un robot móvil basado en ROS [2]. Ambos deben estar conectados a la misma red, para poder enviar y recibir información.

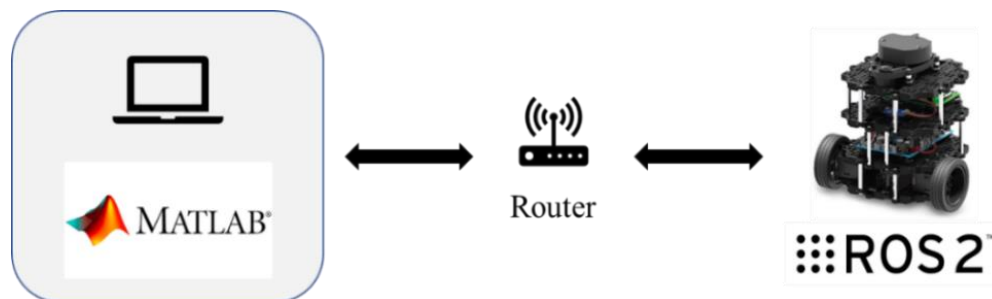


Figura 1. Diagrama de la arquitectura de comunicaciones.

Para la implementación de ambos controles se han combinado dos de las formas de conexión posibles entre el entorno de MATLAB/Simulink y el robot. Por un lado, se ha utilizado el método de generación de nodos de ROS para desplegar en la Raspberry Pi del robot el control PID completo, diseñado en Simulink, haciendo uso de la generación de código C++ (Simulink CODER) y la ROS *Toolbox* de MATLAB [3] [4]. Por otro

lado, se ha usado el método de conexión con ROS para el envío de parámetros y consignas, así como para recoger los resultados de los ensayos para su posterior monitorización [5]. Esto evita esperas a la hora de desplegar de nuevo el control, en caso de que se quiera hacer cambios en los parámetros o consignas.

3. Resultados

Control de velocidad de avance

El modelado de la planta de este control se ha llevado a cabo mediante un proceso de ajuste por mínimos cuadrados. El resultado obtenido se muestra en la Ecuación 1 donde, además de la constante de tiempo y la ganancia estática, se observa que hay un retardo de 120 ms, causado por los retrasos en las comunicaciones de los distintos elementos de la red.

$$P(s) = \frac{V(s)}{U(s)} = \frac{K_m}{T_m \cdot s + 1} \cdot e^{-\rho s} = \frac{0,055}{0,040 \cdot s + 1} \cdot e^{-0,12s}$$

Ecuación 1. Planta del control de velocidad de avance.

Con dicho modelo de la planta se diseñaron, mediante técnicas de respuesta en frecuencia, los controles PID, escogiendo finalmente un control PI con los siguientes parámetros: $K = 8,0$, $T_i = 0,09$ y $b = 1,5$ (ver Figura 2). Se ha escogido este control ya que, gracias a la parte integral, permite conseguir error nulo en seguimiento a referencia y rechazo a perturbación, en este caso causada por el par de fricción debido a la rampa del circuito. No se ha escogido un control con parte diferencial ya que, debido al retardo de la planta, no se logró obtener un incremento de rapidez significativo con respecto al PI.

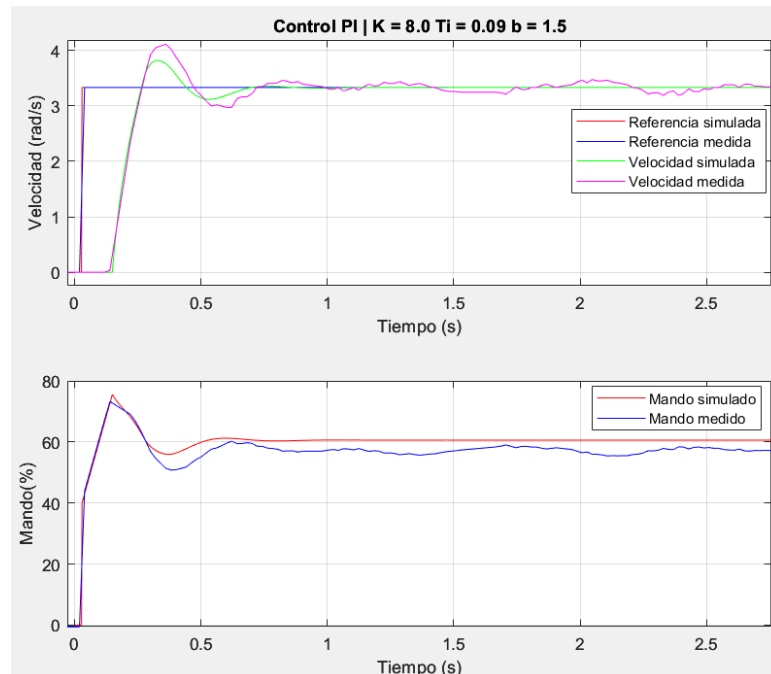


Figura 2. Control PI escogido para el control de velocidad de avance.

Control de seguimiento de pared

Para el control de seguimiento de pared se ha implementado un control en cascada con dos lazos. El lazo externo que tiene como entrada la referencia de distancia deseada y como salida la referencia de ángulo para el lazo interno. Este intenta alcanzar dicha

referencia aplicando la tensión diferencial necesaria a los motores. La planta del lazo interno ($P_1(s)$) se identificó mediante un proceso de ajuste por mínimos cuadrados y la del lazo externo ($P_2(s)$) obteniendo los valores necesarios del punto de operación (la velocidad v_o y la distancia al punto desde donde se mide la distancia x_A), como se muestra en la Ecuación 2. Estas medidas se toman con un sensor VL6180X ToF, ya que el LiDAR, que era el que se iba a utilizar, no medía correctamente debido a los brillos del material de la pared del circuito del laboratorio. Una vez se solucionó el problema se identificaron de nuevo las plantas con el LiDAR (ver Ecuación 3) y se diseñó un control PID con él, para comparar el funcionamiento del robot con cada uno.

$$P_1(s) = \frac{\theta(s)}{U_d(s)} = \frac{K_m}{(T_m \cdot s + 1) \cdot s} = \frac{0,04}{(0,2 \cdot s + 1) \cdot s} \quad P_2(s) = \frac{D_n(s)}{\theta(s)} = \frac{v_o \cdot \left(1 + \frac{x_A}{v_o} \cdot s\right)}{s} = \frac{0,15 \cdot \left(1 + \frac{0,06}{0,15} s\right)}{s}$$

Ecuación 2. Plantas del control de seguimiento de pared con el sensor ToF.

$$P_1(s) = \frac{\theta(s)}{U_d(s)} = \frac{K_m}{(T_m \cdot s + 1) \cdot s} = \frac{0,023}{(0,2 \cdot s + 1) \cdot s} \quad P_2(s) = \frac{D_n(s)}{\theta(s)} = \frac{v_o \cdot \left(1 + \frac{x_A}{v_o} \cdot s\right)}{s} = \frac{0,11 \cdot \left(1 + \frac{0,04}{0,11} s\right)}{s}$$

Ecuación 3. Plantas del control de seguimiento de pared con el LiDAR.

Con el modelo de la Ecuación 2 se diseñaron mediante técnicas de respuesta en frecuencia los controles, escogiendo finalmente un PID con los siguientes parámetros: $K = 31,18$, $T_i = 0,99$, $T_d = 0,15$, $b = 0,9$ y $N = 7$. Se ha escogido este control ya que con él se conseguía error nulo en seguimiento a referencia. Además, gracias a la parte diferencial se obtiene una mayor rapidez que con los controles P y PI. Por otro lado, este control logra una mejor respuesta frente a la perturbación causada por la curva del circuito ya que logra eliminar el error en régimen permanente, gracias a la parte integral, y tiene un pico menor gracias a la parte diferencial, que compensa más rápidamente la perturbación y hace que el robot se acerque lo menos posible a la pared durante la curva. A continuación, se muestra la respuesta de este frente a un escalón en la referencia de distancia (Figura 3) y frente a la perturbación causada por el cambio en el ángulo debido a la llegada de la curva del circuito (Figura 4). Se puede observar la presencia de un retardo de 0,2 segundos en la gráfica de distancia, sin embargo, se ha decidido no incluirlo en el modelo ya que no contribuía a que la simulación tuviera mayor parecido con el ensayo.

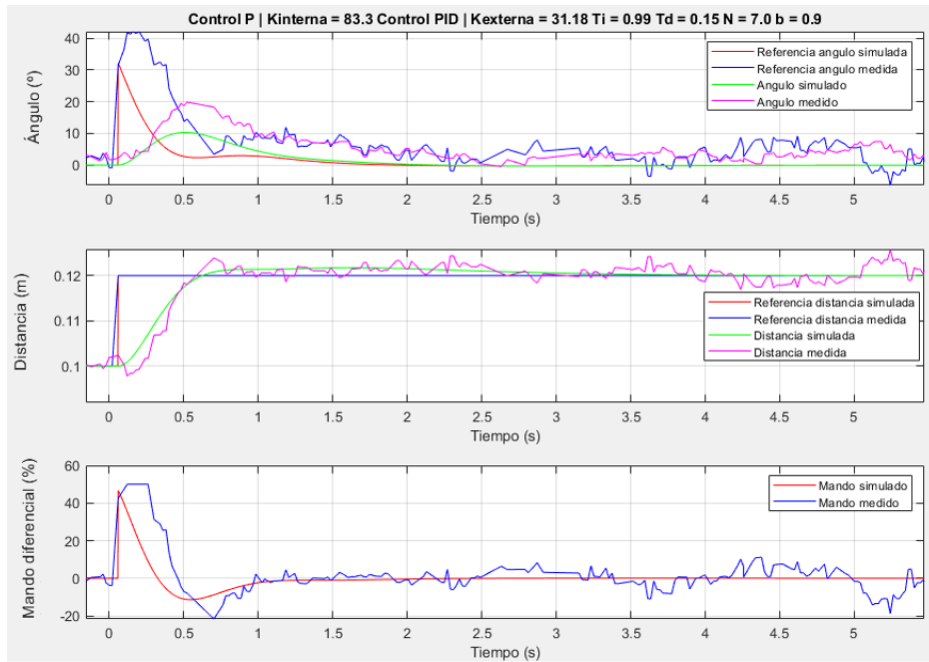


Figura 3. Control PID escogido para el control de seguimiento de pared con el ToF – respuesta a escalón en referencia.

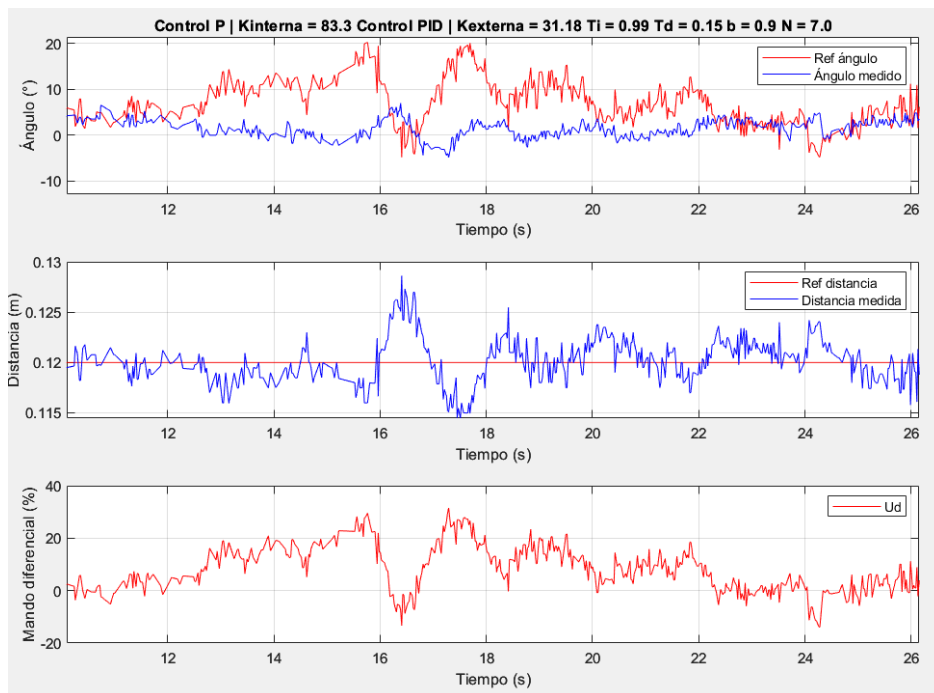


Figura 4. Control PID escogido para el control de seguimiento de pared con el ToF – respuesta a perturbación.

4. Conclusiones

En este proyecto se han diseñado los controles PID de velocidad y seguimiento de pared mostrados en el apartado anterior, a partir de los modelos de las plantas obtenidos. En el caso del control de seguimiento de pared se realizaron con un sensor VL6180X ToF. Una vez resueltos los problemas que presentaba el funcionamiento del control con el LiDAR se modeló la planta con él y se diseñó un control PID. Comparando ambos resultados se concluye que los sensores no son intercambiables, ya que la ganancia estática de la planta es distinta, por lo que se deben diseñar controles específicos para

cada uno de ellos. Además, en el caso del LiDAR sí se ha modelado el retardo de 0,2 segundos identificado, ya que contribuía significativamente a que la simulación se pareciera más al ensayo.

Por último, se ha estudiado como varía el comportamiento del robot si se cambia la velocidad del punto de operación (v_o) ya que, como se ha comentado anteriormente, la planta del lazo externo del control de seguimiento de pared ($P_2(s)$) depende de ella. De los ensayos realizados se puede concluir que es necesario trabajar con la velocidad con la que se ha diseñado ya que en caso contrario los ensayos presentan sobrepasos excesivos y son demasiado oscilantes, especialmente si se ensaya a una velocidad mayor que la usada para diseñar.

5. Referencias

- [1] F. Barragán Castro, «Implementación de algoritmos de navegación en un TurtleBot3,» Madrid, 2022.
- [2] Robotis, «ROBOTIS e-Manual,» [En línea]. Available: <https://manual.robotis.com/docs/en/platform/turtlebot3/overview/#overview>. [Último acceso: 11 Julio 2023].
- [3] MathWorks, «Get Started with ROS,» [En línea]. Available: <https://www.mathworks.com/help/ros/ug/get-started-with-ros.html>. [Último acceso: 11 Julio 2023].
- [4] MathWorks, «Exchange Data with ROS Publishers and Subscribers,» [En línea]. Available: <https://www.mathworks.com/help/ros/ug/exchange-data-with-ros-publishers-and-subscribers.html>. [Último acceso: 29 Junio 2023].
- [5] MathWorks, «ROS 2 Node Generation and Deployment,» [En línea]. Available: <https://es.mathworks.com/help/ros/ros2-node-generation-and-deployment.html>. [Último acceso: 26 Junio 2023].

DESIGN AND IMPLEMENTATION OF SPEED AND WALL FOLLOWING PID CONTROLLERS FOR A TURTLEBOT3

Author: Tabasco Ruiz, Alejandra.

Supervisors: Boal Martín-Larrauri, Jaime; Rodríguez Pecharromán, Ramón.

Collaborating Entity: ICAI – Universidad Pontificia Comillas.

ABSTRACT

The main objective of this project is to acquire experience and master the use of the TurtleBot3 Burger for control and mobile robotics laboratory assignments. To this end, PID speed and wall following controls have been designed and implemented, studying their reference tracking response and disturbance rejection.

Keywords: TurtleBot3, ROS2, MATLAB, Simulink, PID Controller.

1. Introduction and project definition

This project builds upon a final degree project carried out during the 2021-2022 academic year [1]. This project aims to update the platform that is currently used in the control laboratories of different subjects at the university. This master's thesis main objective is to acquire experience and master the use of the robot for control and mobile robotics laboratory assignments. To this end, PID speed and wall following controls have been designed, based on the plant models obtained. Subsequently, the most appropriate way to implement these controls in the robot has been determined, considering that its use is in teaching practices. Finally, tests have been carried out with the designed controls, studying their reference tracking and disturbance rejection behavior.

2. System's description

Figure 1 shows a diagram of the system's communications architecture. On the one hand, there is the computer in which MATLAB is installed, from which the references and necessary parameters are sent, in addition to collecting the results of the tests for their subsequent monitorization. On the other hand, there is the TurtleBot3 Burger, a mobile robot based on ROS [2]. Both must be connected to the same network, in order to send and receive information.

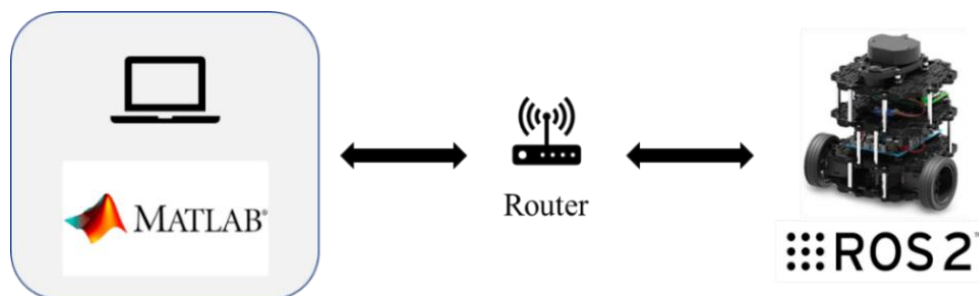


Figure 1. Diagram of the systems communication's architecture.

For the implementation of both controls, two of the possible connection forms between the MATLAB/Simulink environment and the robot have been combined. On the one hand, the ROS node generation method has been used to deploy the complete PID control, designed in Simulink, on the robot's Raspberry Pi, making use of C++ code generation (Simulink CODER) and the MATLAB ROS Toolbox [3] [4]. On the other

hand, the ROS connection method has been used to send parameters and instructions, as well as to collect the results of the tests for their monitorization [5]. This avoids waiting times when deploying the control again, in case the user needs to implement changes to the parameters or references.

3. Results

Speed control

The plant model of this control has been obtained using a least squares adjustment process. The result is shown in Equation 1 where, in addition to the time constant and the static gain, there is a delay of 120 ms, caused by the delays in the communications of the different elements of the network.

$$P(s) = \frac{V(s)}{U(s)} = \frac{K_m}{T_m \cdot s + 1} \cdot e^{-\rho s} = \frac{0.055}{0.040 \cdot s + 1} \cdot e^{-0.12s}$$

Equation 1. Speed control plant model.

With this model of the plant, the PID controls were designed using frequency response techniques, finally choosing a PI control with the following parameters: $K = 8.0$, $T_i = 0.09$ and $b = 1.5$ (see Figure 2). This control has been chosen since, thanks to the integral part, it allows zero error in reference tracking and disturbance rejection, in this case caused by the friction torque due to the circuit ramp. A control with a differential part was not deemed necessary since, due to the delay of the plant, it is not possible to obtain a significant speed increase with respect to the PI.

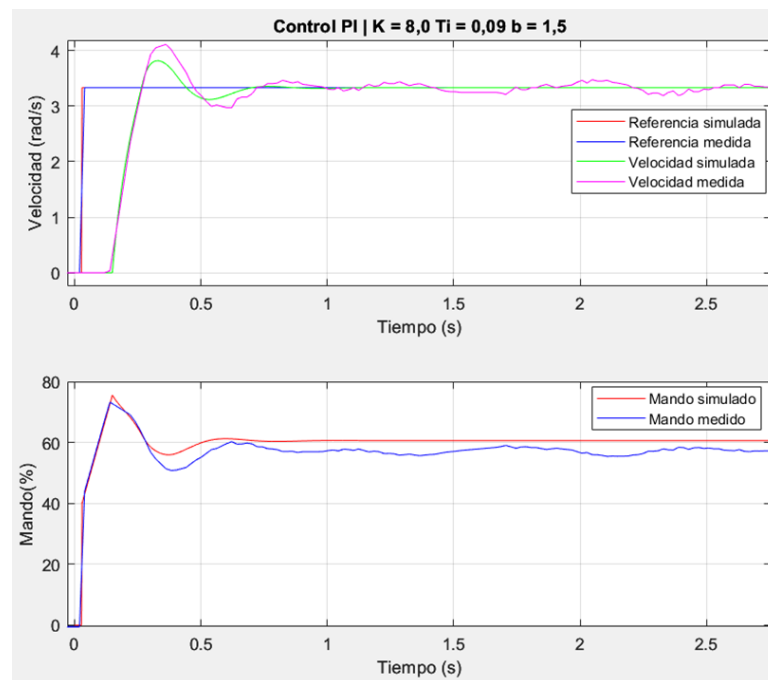


Figure 2. PI control chosen for the speed control.

Wall follower control

For the wall follower control, a cascade control with two loops has been implemented. The outer loop that has the desired distance reference as input and the angle reference for the inner loop as output. This loop tries to reach the reference by applying the necessary differential voltage to the motors. The plant of the internal loop ($P_1(s)$) was identified through a least squares adjustment process and the external loop plant ($P_2(s)$)

was identified obtaining the necessary values from the operating point (velocity v_o and the distance to the point from where the distance is measured x_A), as shown in Equation 2. These measurements are taken with a VL6180X ToF sensor, since the LiDAR, which was the one to be used, did not measure correctly due to the brightness of the material of the circuit wall of the laboratory. Once that problem was solved, the plants were identified again with the LiDAR (see Equation 3) and a PID control was designed with it, to compare the operation of the robot with each one.

$$P_1(s) = \frac{\theta(s)}{U_d(s)} = \frac{K_m}{(T_m \cdot s + 1) \cdot s} = \frac{0,04}{(0,2 \cdot s + 1) \cdot s} \quad P_2(s) = \frac{D_n(s)}{\theta(s)} = \frac{v_o \cdot \left(1 + \frac{x_A}{v_o} \cdot s\right)}{s} = \frac{0,15 \cdot \left(1 + \frac{0,06}{0,15} s\right)}{s}$$

Equation 2. Wall following control plants with the ToF sensor.

$$P_1(s) = \frac{\theta(s)}{U_d(s)} = \frac{K_m}{(T_m \cdot s + 1) \cdot s} = \frac{0,023}{(0,2 \cdot s + 1) \cdot s} \quad P_2(s) = \frac{D_n(s)}{\theta(s)} = \frac{v_o \cdot \left(1 + \frac{x_A}{v_o} \cdot s\right)}{s} = \frac{0,11 \cdot \left(1 + \frac{0,04}{0,11} s\right)}{s}$$

Equation 3. Wall following control plants with the LiDAR.

With the model shown in Equation 2, the controls were designed using frequency response techniques, finally choosing a PID with the following parameters: $K = 31.18$, $T_i = 0.99$, $T_d = 0.15$, $b = 0.9$ and $N = 7$. This control was chosen since with it zero error was achieved in reference tracking. In addition, thanks to the differential part, the control is faster than P and PI controls. On the other hand, this control achieves a better response to the disturbance caused by the circuit curve, since it manages to eliminate the steady state error thanks to the integral part, and has a lower peak thanks to the differential part, which compensates the disturbance faster and keeps the robot at a safer distance from the wall during the curve. Figure 3 shows its response to a step in the distance reference and Figure 4 shows the response to the disturbance caused by the change in angle due to the arrival of the circuit curve. The presence of a delay of 0.2 seconds can be observed in the distance graph. However, it was not included in the model since it does not contribute to the simulation having a greater resemblance to the test.

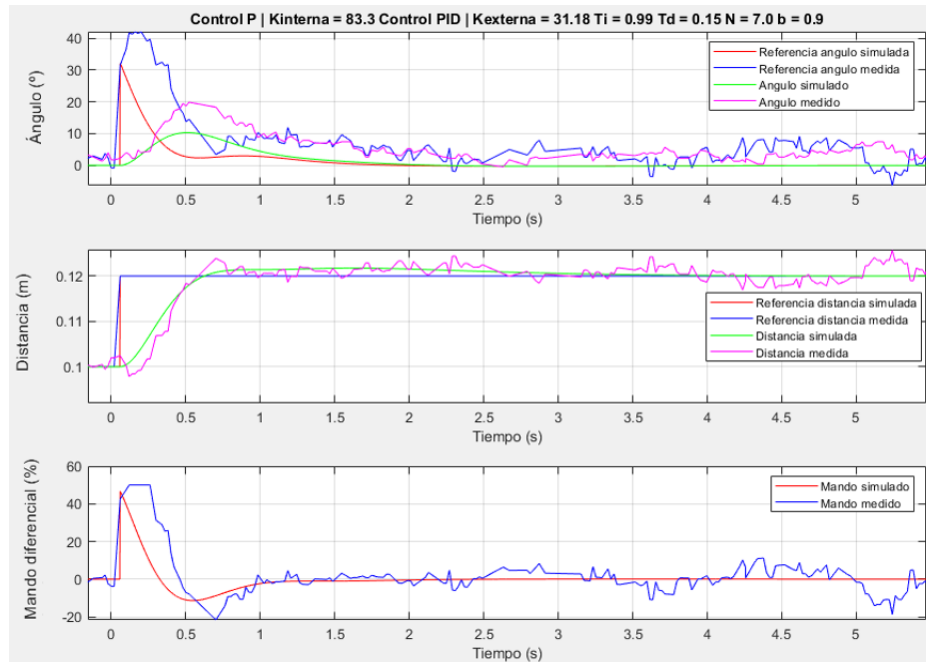


Figure 3. PID control chosen for wall following control with the ToF sensor - response to a step in the distance reference.

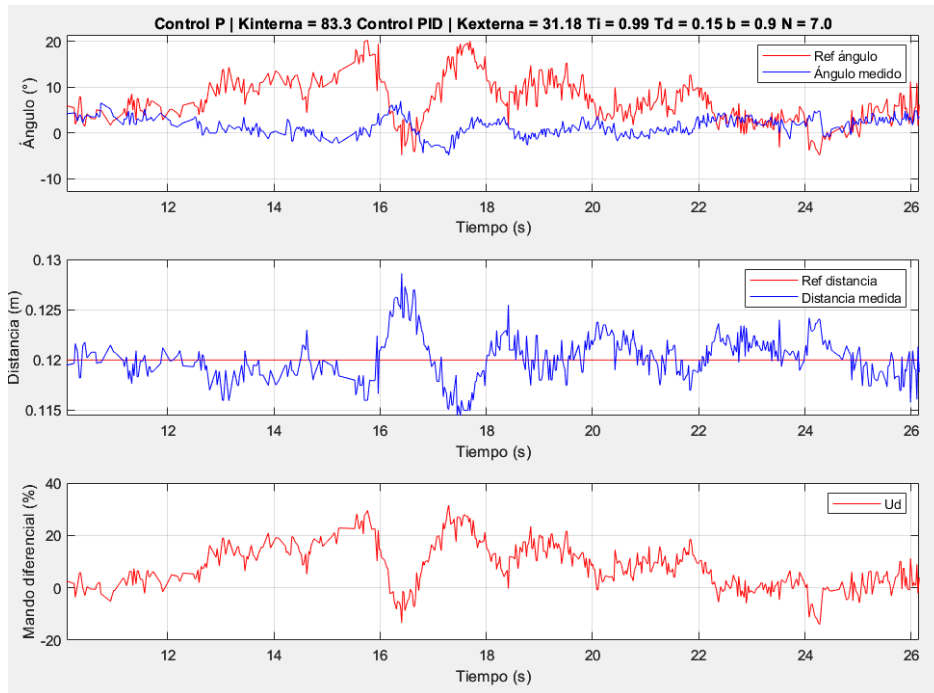


Figure 4. PID control chosen for wall following control with the ToF sensor – response to the disturbance.

4. Conclusions

In this project, the PID speed and wall following controls shown in the previous section have been designed, based on the plant models obtained. In the case of wall following control, they were carried out with a VL6180X ToF sensor. Once the problems presented by the performance of the control with the LiDAR were solved, the plant was modeled with it and a PID control was designed as well. Comparing both results, it can be concluded that the sensors are not interchangeable, since the static gain of the plant is different, so specific controls must be designed for each of them. In addition, in the case of the LiDAR, a 0.2 second delay has been modeled, since it contributes significantly to making the simulation look more similar to the test.

Finally, it has been studied how the behavior of the robot varies if the speed of the operating point (v_o) is changed since, as aforementioned, the plant of the external loop of the wall following control ($P_2(s)$) depends on its value. From the tests carried out, it is necessary to work with the speed used to design the control; otherwise, the tests present excessive overshoot and become too oscillating, especially if tested at a speed greater than that used for the design.

5. References

- [1] F. Barragán Castro, «Implementación de algoritmos de navegación en un TurtleBot3,» Madrid, 2022.
- [2] Robotis, «ROBOTIS e-Manual,» [En línea]. Available: <https://emanual.robotis.com/docs/en/platform/turtlebot3/overview/#overview>. [Último acceso: 11 Julio 2023].

- [3] MathWorks, «Get Started with ROS,» [En línea]. Available: <https://www.mathworks.com/help/ros/ug/get-started-with-ros.html>. [Último acceso: 11 Julio 2023].
- [4] MathWorks, «Exchange Data with ROS Publishers and Subscribers,» [En línea]. Available: <https://www.mathworks.com/help/ros/ug/exchange-data-with-ros-publishers-and-subscribers.html>. [Último acceso: 29 Junio 2023].
- [5] MatWorks, «ROS 2 Node Generation and Deployment,» [En línea]. Available: <https://es.mathworks.com/help/ros/ros2-node-generation-and-deployment.html>. [Último acceso: 26 Junio 2023].

Índice de la memoria

Capítulo 1. Introducción	9
1.1 Motivación del proyecto.....	9
1.2 Objetivos del proyecto.....	9
1.3 Estado de la cuestión	9
Capítulo 2. Descripción del hardware	15
2.1 OpenCR.....	15
2.2 Single Board Computer (SBC).....	16
2.3 Dynamixel XL430-W250-T	16
2.4 Sensor LDS-02	17
2.5 Sensor VL6180X ToF.....	18
Capítulo 3. Arquitectura de comunicaciones	20
3.1 ROS2	20
3.2 ROS2 Toolbox de MATLAB	24
Capítulo 4. Control de velocidad de avance.....	27
4.1 Identificación del modelo de la planta.....	27
4.2 Diseño de los controles PID	29
4.2.1 Control Proporcional, P.....	30
4.2.2 Control proporcional más integral, PI.....	32
4.2.3 Control proporcional diferencial, PD.....	34
4.2.4 Control proporcional integral diferencial, PID.....	36
4.3 Implementación de los controles de velocidad de avance.....	41
4.4 Comparación entre simulación y ensayo.....	43
4.5 Conclusiones del control de velocidad de avance	48
Capítulo 5. Control de seguimiento de distancia a la pared	49
5.1 Identificación de la planta [30].....	49
5.2 Diseño de los controles PID	53
Diseño del control del lazo interno	53
5.2.1 Control Proporcional, P.....	54
Diseño del control del lazo externo.....	55
5.2.2 Control Proporcional, P.....	56
5.2.3 Control proporcional integral, PI.....	57
5.2.4 Control proporcional diferencial, PD.....	59

5.2.5 Control proporcional integral diferencial, PID	61
5.3 Implementación del control de seguimiento de pared	65
5.4 Comparación entre simulación y ensayo para el control de seguimiento de pared	68
5.5 Respuesta a perturbación de la curva del control de seguimiento de pared	73
5.6 Conclusiones del control de seguimiento de distancia a la pared.....	77
5.7 Estudios complementarios.....	78
5.8 Control de seguimiento de pared con el LiDAR	81
Capítulo 6. Conclusiones y Trabajos Futuros.....	87
6.1 Conclusiones	87
6.1.1 Control de velocidad de avance	87
6.1.2 Control de seguimiento de pared.....	87
6.2 Trabajos futuros.....	88
Capítulo 7. Bibliografía.....	90
ANEXO I. Alineación con los objetivos de desarrollo sostenible	93

Índice de figuras

Figura 1. QBot 3 [7].	10
Figura 2. Alfabot2 [8].	11
Figura 3. TurtleBot3 [9].	11
Figura 4. Thymio 2 [10].	12
Figura 5. LIMO [11].	12
Figura 6. Bveeta Mini Type R007 [12].	13
Figura 7. Vehículo con tracción diferencial utilizado actualmente en el laboratorio [13]. .	13
Figura 8. Componentes del Turtlebot3 Burger [14].	15
Figura 9. Placa OpenCR [16].	16
Figura 10. Gráfico de rendimiento del Dynamixel XL430-W250-T [18].	17
Figura 11. Sensor LDS-02 [19].	18
Figura 12. Disposición de los sensores VL6180X ToF en el robot.	19
Figura 13. Diagrama de la arquitectura de comunicaciones.	20
Figura 14. Modo de funcionamiento de los elementos de una red de ROS2 [24].	21
Figura 15. Ejemplo de gráfico de nodos generado con rqt con el LiDAR.	24
Figura 16. Ejemplo de gráfico de nodos generado con rqt con el sensor ToF.	24
Figura 17. Conexión con la ROS2 Toolbox de MATLAB [25].	25
Figura 18. Comparativa ensayos.	27
Figura 19. Ensayo utilizado para la identificación de la planta del control de velocidad. ..	28
Figura 20. Resultado de la identificación de la planta del control de velocidad.	29
Figura 21. Diagrama de bloques del sistema.	29
Figura 22. Gráfica de Black para la planta del control de velocidad.	30
Figura 23. Gráfico de Black para la planta $P(s)$ y el lazo abierto $G(s)$ para el control P. ...	32
Figura 24. Respuesta a un escalón unitario en referencia y perturbación para el control P.32	32
Figura 25. Gráfico de Black para la planta $P(s)$ y el lazo abierto $G(s)$ para el control PI. ..	34
Figura 26. Respuesta a un escalón unitario en referencia y perturbación para el control PI.	34
Figura 27. Gráfico de Black para la planta $P(s)$ y el lazo abierto $G(s)$ para el control PD. 36	36
Figura 28. Respuesta a un escalón unitario en referencia y perturbación para el control PD.	36
Figura 29. Gráfico de Black para la planta $P(s)$ y el lazo abierto $G(s)$ para el control PID.38	38
Figura 30. Respuesta a un escalón unitario en referencia y perturbación para el control PID.	38
Figura 31. Comparativa de la respuesta a un escalón unitario en referencia para los controles diseñados.	39
Figura 32. Comparativa a un escalón unitario en perturbación para los controles diseñados.	40
Figura 33. Diagrama del control de velocidad desplegado.	42
Figura 34. Archivo de monitorización y envío de referencia.	43
Figura 35. Gráfico de nodos del control de velocidad.	43
Figura 36. Comparación entre ensayo y simulación para el control P.	44

Figura 37. Comparación entre ensayo y simulación para el control PI.....	45
Figura 38. Comparación entre ensayo y simulación para el control PD con acción diferencial al error.....	46
Figura 39. Comparación entre ensayo y simulación para el control PD con acción diferencial a la salida.	46
Figura 40. Comparación entre ensayo y simulación para el control PID con acción diferencial a la salida.	47
Figura 41. Comparación entre ensayo y simulación para el control PID con acción diferencial al error.....	48
Figura 42. Medidas del sensor ToF.	50
Figura 43. Ensayo utilizado para la identificación de la planta $P_1(s)$	52
Figura 44. Resultado de identificación de la planta $P_1(s)$	52
Figura 45. Diagrama del sistema a controlar.	53
Figura 46. Diagrama de Black de la planta del lazo interno.....	54
Figura 47. Gráfico de Black para la planta $P(s)$ y el lazo abierto $G(s)$ para el control P del lazo interno.	55
Figura 48. Respuesta a un escalón unitario en referencia y perturbación para el control P del lazo interno.	55
Figura 49. Diagrama de Black de la planta del lazo externo.	56
Figura 50. Gráfico de Black para la planta $P(s)$ y el lazo abierto $G(s)$ para el control P del lazo externo.....	56
Figura 51. Respuesta a un escalón unitario en referencia y perturbación para el control P del lazo externo.....	57
Figura 52. Gráfico de Black para la planta $P(s)$ y el lazo abierto $G(s)$ para el control PI del lazo externo.....	58
Figura 53. Respuesta a un escalón unitario en referencia y perturbación para el control PI del lazo externo.....	59
Figura 54. Gráfico de Black para la planta $P(s)$ y el lazo abierto $G(s)$ para el control PD del lazo externo.....	61
Figura 55. Respuesta a un escalón unitario en referencia y perturbación para el control PD del lazo externo.....	61
Figura 56. Gráfico de Black para la planta $P(s)$ y el lazo abierto $G(s)$ para el control PID del lazo externo.....	63
Figura 57. Respuesta a un escalón unitario en referencia y perturbación para el control PID del lazo externo.....	63
Figura 58. Comparativa de la respuesta a un escalón unitario en referencia para los controles diseñados.	64
Figura 59. Comparativa a un escalón unitario en perturbación para los controles diseñados.	64
Figura 60. Diagrama del control de seguimiento de pared desplegado.....	67
Figura 61. Subsistema $U(s)$	67
Figura 62. Archivo de monitorización y envío de referencia para el control de seguimiento de pared.....	68
Figura 63. Gráfico de nodos del control de seguimiento de pared.	68
Figura 64. Comparación entre ensayo y simulación para el control P.	70

Figura 65. Comparación entre ensayo y simulación para el control PI.	71
Figura 66. Comparación entre ensayo y simulación para el control PD con acción diferencial a la salida.	72
Figura 67. Comparación entre ensayo y simulación para el control PID con acción diferencial a la salida.	73
Figura 68. Respuesta a perturbación para el control P.	74
Figura 69. Respuesta a perturbación para el control PI.	75
Figura 70. Respuesta a perturbación para el control PD con acción diferencial a la salida.	76
Figura 71. Respuesta a perturbación para el control PID con acción diferencial a la salida.	77
Figura 72. Control PID diseñado y ensayado con una velocidad de 0,1 m/s.	78
Figura 73. Control PID diseñado y ensayado con una velocidad de 0,15 m/s.	79
Figura 74. Control PID diseñado para una velocidad de 0,15 m/s y ensayado con una velocidad de 0,1 m/s.	80
Figura 75. Control PID diseñado para una velocidad de 0,1 m/s y ensayado con una velocidad de 0,15 m/s.	81
Figura 76. Medidas de ángulo y distancia a la pared con el LiDAR.	82
Figura 77. Ensayo utilizado para la identificación de la planta $P_1(s)$ con el LiDAR.	83
Figura 78. Resultado de identificación de la planta $P_1(s)$ con el LiDAR.	83
Figura 79. Diagrama de bloques del sistema a controlar con el LiDAR.	84
Figura 80. Gráfico de Black para la planta $P(s)$ y el lazo abierto $G(s)$ para el control PID del lazo externo con el LiDAR.	85
Figura 81. Respuesta a un escalón unitario en referencia y perturbación para el control PID del lazo externo con el LiDAR.	85
Figura 82. Gráfico de nodos del control de seguimiento de pared con el LiDAR.	86
Figura 83. Comparación entre ensayo y simulación para el control PID con el LiDAR. ...	86

Índice de tablas

Tabla 1. Comparativa de características.	14
Tabla 2. Especificaciones del rendimiento de las medidas del LDS-02 [19].	18
Tabla 3. Estructura del mensaje publicado en el topic /imu.	22
Tabla 4. Estructura del mensaje publicado en el topic /joint_states.	23
Tabla 5. Estructura del mensaje publicado en el topic /cmd_vel.	23
Tabla 6. Estructura del mensaje publicado en el topic /scan.	23
Tabla 7. Estructura del mensaje publicado en el topic /lidar.	24
Tabla 8. Resumen del diseño de controles de velocidad.	40
Tabla 9. Comparativa de la respuesta temporal para los controles diseñados.	41
Tabla 10. Resumen del diseño de los controles de seguimiento de pared.	65
Tabla 11. Comparativa de la respuesta temporal para los controles diseñados.	65

Índice de ecuaciones

Ecuación 1. Control proporcional (formato serie).....	30
Ecuación 2. Función de transferencia para el control PI (formato serie).	32
Ecuación 3. Función de transferencia para el control PD (formato serie).....	35
Ecuación 4. Función de transferencia para el control PID (formato serie).	36
Ecuación 5. Medidas de distancia y ángulo de orientación con respecto a la pared.	49
Ecuación 6. Velocidad del punto A.	50
Ecuación 7. Componente normal de la velocidad del punto A.	50
Ecuación 8. Relación entre velocidad angular y tensión diferencial.	50
Ecuación 9. Derivada del ángulo del robot con respecto a la pared.	51
Ecuación 10. Función de transferencia de la planta del control de seguimiento de pared. .	51
Ecuación 11. Función de transferencia de la planta $P_1(s)$	51
Ecuación 12. Función de transferencia de la planta $P_2(s)$	53
Ecuación 13. Control proporcional (formato serie).....	54
Ecuación 14. Función de transferencia para el control PI (formato serie).	57
Ecuación 15. Función de transferencia para el control PD (formato serie).....	59
Ecuación 16. Función de transferencia para el control PID (formato serie).	61
Ecuación 17. Velocidad de rotación de la pared.	73
Ecuación 18. Medidas de ángulo y distancia a la pared con el LiDAR.	81
Ecuación 19. Velocidad del punto A con el LiDAR.	82
Ecuación 20. Función de transferencia de la planta del control de seguimiento de pared con el LiDAR.	82
Ecuación 21. Función de transferencia de la planta $P_1(s)$ con el LiDAR.	84
Ecuación 22. Función de transferencia de la planta $P_2(s)$	84

Acrónimos

ToF	Time of Flight
SBC	Single Board Computer
ROS	Robot Operating System
PWM	Pulse-Width Modulation
OpenCR	Open-source Control Module for ROS
LiDAR	Laser Imaging Detection and Ranging

Capítulo 1. INTRODUCCIÓN

Este trabajo fin de máster tiene por objetivo el diseño e implementación de los controles PID de velocidad y seguimiento de pared para un TurtleBot 3, un robot móvil con cinemática diferencial. Estos controles se han realizado usando MATLAB y Simulink. En el robot se ha utilizado Robot Operating System (ROS), un conjunto de herramientas y librerías de *software* para la creación de aplicaciones robóticas. Además, gracias a ser de código abierto aporta gran flexibilidad para adaptarlo a las necesidades de cada caso de uso [6].

1.1 MOTIVACIÓN DEL PROYECTO

Este proyecto surge como continuación de un trabajo fin de grado realizado durante el curso 2021-2022 [1]. Con este proyecto se pretende actualizar la plataforma que se usa actualmente en los laboratorios de control de distintas asignaturas de la universidad. Esta plataforma se trata de un diseño propio, pero se quiere analizar una solución comercial que pueda ser también utilizada en asignaturas de robótica. Por ello este proyecto trata de desarrollar un robot válido, que permita controlar el movimiento (controles de velocidad y distancia a la pared) para, a continuación, introducir las tareas propias de navegación de un robot móvil con fines docentes para el nuevo grado en iMAT.

1.2 OBJETIVOS DEL PROYECTO

El objetivo general de este proyecto es adquirir experiencia y dominio en el uso del robot para prácticas de control y robótica móvil.

Los objetivos específicos son:

1. Determinar ventajas e inconvenientes de cada forma de conexión entre el entorno de MATLAB/Simulink y el robot.
2. Desarrollar una metodología para obtener el modelo del robot apta para su uso en prácticas docentes.
3. Desarrollar e implementar un control de velocidad.
4. Desarrollar e implementar un control de distancia a la pared.

1.3 ESTADO DE LA CUESTIÓN

En este apartado se exponen distintos robots que son utilizados en entornos educativos universitarios para enseñar asignaturas de control. Se presenta una breve descripción de sus

aspectos más relevantes, así como una tabla comparativa de sus características más importantes.

QBot 3 [7]

Se trata de un robot autónomo como el mostrado en la Figura 1, de arquitectura abierta, que está construido sobre una plataforma móvil y comercializado por Quanser. Este tipo de arquitectura permite la personalización del robot, gracias a la posibilidad de añadir distintos sensores, para adaptarlo al área en el que se va a utilizar. Cuenta también con la posibilidad de poder implementar aplicaciones desde Simulink, Python o ROS. Por otro lado, es necesario contar con un *software* propio (Quanser QUARC Rapid Control Prototyping *software*) que se integre con Simulink. Dicho *software* requiere de una suscripción para poder usarlo por lo que no es idóneo para este proyecto, ya que es preferible no tener dependencia de fabricantes.



Figura 1. QBot 3 [7].

Alphabot2 [8]

Es un robot de la empresa Waveshare diseñado para poder usarse con Raspberry Pi 3 Modelo B, Arduino o Raspberry Pi Zero W, como el mostrado en la Figura 2. Permite un montaje rápido gracias a su diseño modular y cuenta con funcionalidades como seguimiento de línea o detección de obstáculos. Como aspecto negativo se podría destacar que requeriría más trabajo realizar la integración con MATLAB.



Figura 2. AlphasBot2 [8].

TurtleBot3 [9]

Se trata de un robot pequeño que cuenta con una gran comunidad de usuarios (ver Figura 3). Surgió como alternativa a sus versiones anteriores, con el objetivo de mantener la calidad y la funcionalidad, pero disminuyendo el coste y el tamaño. Es un robot personalizable para adaptarlo a las necesidades de uso y cuenta con una autonomía de dos horas y media. Se puede operar desde ROS y permite la conexión con MATLAB mediante el uso de la ROS *Toolbox*.

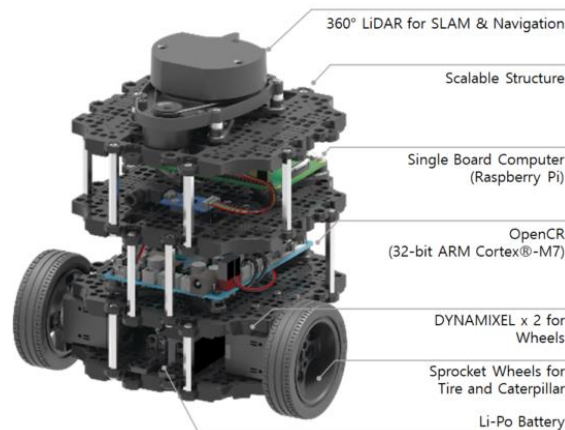


Figura 3. TurtleBot3 [9].

Thymio 2 [10]

Este robot fue diseñado en colaboración por el Instituto Federal Suizo de Tecnología en Lausana (EPFL) y la Escuela Cantonal de Arte de Lausana (ECAL) con el objetivo de desarrollar un robot móvil para ámbitos educativos y de coste reducido, como el mostrado en la Figura 4. Es un robot *open-source* tanto a nivel de *hardware* como de *software* y cuenta con una autonomía de más de dos horas. Permite cinco lenguajes de programación (VPL, VPL3, Scratch, Blockly, Aseba Studio) y es compatible con ROS.



Figura 4. Thymio 2 [10].

LIMO [11]

Se trata de un robot móvil y compacto, como el mostrado en la Figura 5, de la empresa Agile X que ofrece posibilidad de operar desde ROS. La plataforma cuenta con varios puertos USB para poder personalizarlo según las necesidades y su autonomía en funcionamiento sería de cuarenta minutos. Cuenta con un control de movimiento compatible con *bluetooth* para permitir el control diferencial en todas las ruedas.



Figura 5. LIMO [11].

Bveeta Mini Type R007 [12]

Es un robot móvil desarrollado en 2021 por la *start-up* Bizbot Technology como el mostrado en la Figura 6. Está pensado para ser usado tanto en entornos educativos como de investigación y permite la programación con lenguajes como Python o C y puede ser operado desde ROS.



Figura 6. Bveeta Mini Type R007 [12].

Vehículo con tracción diferencial utilizado actualmente en el laboratorio [13]

En la Figura 7 se muestra el vehículo que se utiliza actualmente en las asignaturas de control en la universidad. Cuenta con dos motores DC (EMG30), con un *encoder* en cada uno de ellos para medir el ángulo girado. Está diseñado para trabajar desde el entorno de MATLAB y Simulink y cuenta con una autonomía de 4 horas aproximadamente.

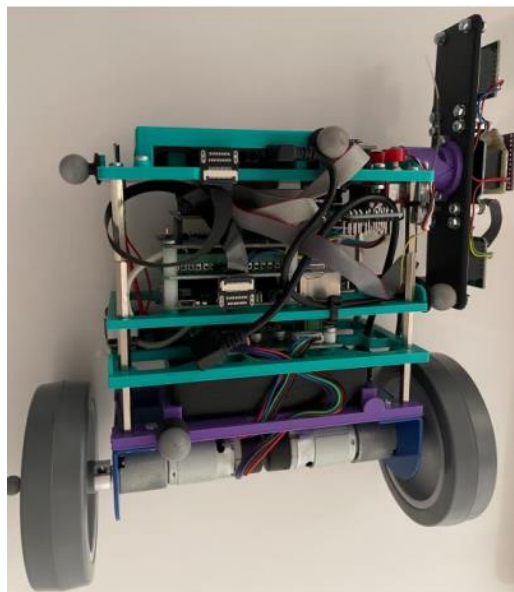


Figura 7. Vehículo con tracción diferencial utilizado actualmente en el laboratorio [13].

En la Tabla 1 se muestra una comparativa de distintas características de los robots mencionados.

Robot	Autonomía	Dimensiones	Cinemática	Hardware soportado	Cámara	Sensor de distancia	Otros Sensores
QBot 3	Tres horas	Base 35 cm de diámetro Altura 16 cm	Plataforma móvil de dos ruedas	Raspberry Pi 4B 4GB	✓	Sensor ultrasónico	3 sensores de golpes, 2 sensores de caída de rueda y 3 sensores de desnivel.
Alphabot2	Soporta baterías 14500	Base 98 mm x 110 mm Altura 56 mm	Dos ruedas de goma + 1 rueda omnidireccional	<ul style="list-style-type: none"> ▪ Arduino ▪ Raspberry Pi 3 Model B ▪ Raspberry Pi Zero W 	✓	Sensor fotoeléctrico infrarrojo reflectante: ST188	Seguimiento de línea: ITR20001/T
TurtleBot3	2h 30 min	Base 138 mm x 178 mm Altura 192 mm	Dos ruedas + esfera de apoyo	Raspberry Pi		LiDAR: LDS-02	IMU (giróscopo de 3 ejes y acelerómetro de 3 ejes.
Thymio 2	2h	Base 11 cm x 11,2 cm Altura 5,3 cm	Dos ruedas	Raspberry Pi		Sensor de infrarrojos (IR)	Seguimiento de línea y sensor de temperatura
LIMO	40 min trabajando 2h en espera	Base 322 mm x 215 mm Altura 247 mm	<ul style="list-style-type: none"> ▪ Cuatro ruedas con dirección diferencial ▪ Oruga ▪ Ackerman ▪ Cuatro ruedas omnidireccionales 	NVIDIA Jetson Nano (4G)	✓	LiDAR: EAI X2L	
Bveeta Mini Type R007	2h 30 min	Base 217,89 mm x 168,94 mm Altura 192,38 mm	Dos ruedas	Raspberry Pi Arduino Mega 2560	✓	LiDAR: YDLidar X2	
Vehículo con tracción diferencial	4h	Base 25,4 cm x 22 cm Altura 17 cm	Dos ruedas	Raspberry Pi 3B+		VL6180X ToF	

Tabla 1. Comparativa de características.

Capítulo 2. DESCRIPCIÓN DEL *HARDWARE*

El TurtleBot3 es un robot móvil basado en ROS que fue creado con la idea de conseguir un diseño que permitiese reducir tanto el tamaño como el coste, sin tener que comprometer la calidad y la funcionalidad. Su diseño modular permite que pueda ser personalizado de distintas formas por el usuario, además de ofrecer una gran extensibilidad. Tanto el *hardware* como el *software* son *open source*, para fomentar la colaboración entre usuarios. Estas características lo convierten en una buena alternativa para proyectos de distinto tipo, como investigación o educativos [14].

En la Figura 8 se muestra el robot con sus componentes principales, que se explican con más detalle a continuación. Además, el TurtleBot3 cuenta con una batería LIPO de 11,1V y 1.800mAh.

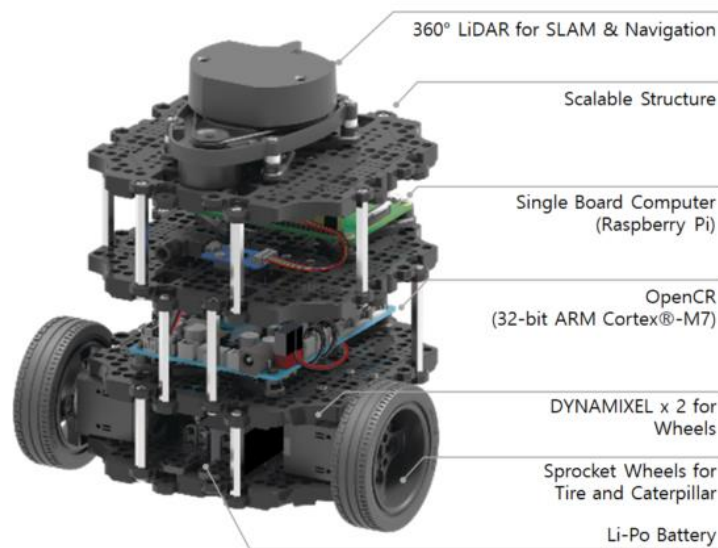


Figura 8. Componentes del Turtlebot3 Burger [14].

2.1 *OPENCR*

OpenCR (Open-source Control Module for ROS) (ver Figura 9) es una placa desarrollada para proporcionar *hardware* y *software open source* para sistemas integrados de ROS [15]. Proporciona los pines de entrada y salida, tanto digitales como analógicos, para interactuar con los sensores y los motores del robot. Entre las interfaces de comunicación que tiene disponible están: USB, UART, SPI, I2C y CAN. El núcleo de la tarjeta es un microcontrolador STM32F7 basado en un ARM Cortex-M7 de 32-bits con unidad de coma flotante. Cuenta además con sensores como un giroscopio y un acelerómetro.

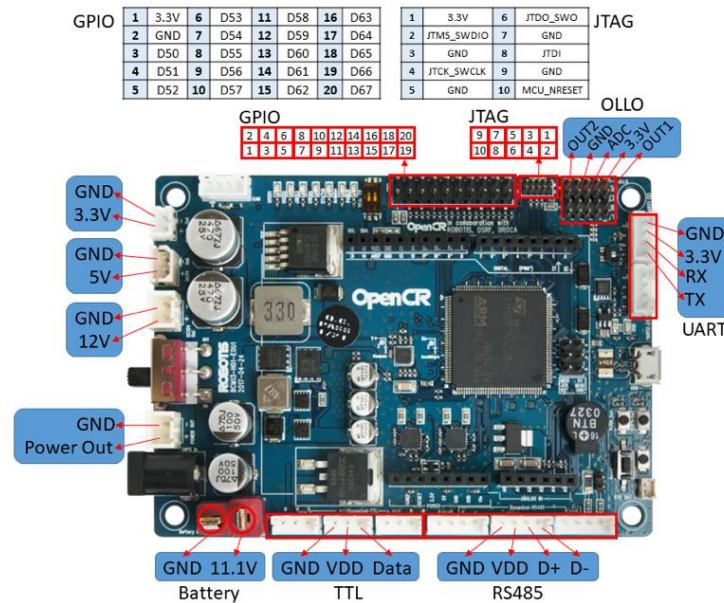


Figura 9. Placa OpenCR [16].

2.2 SINGLE BOARD COMPUTER (SBC)

En este caso la SBC incluida en el TurtleBot3 es una Raspberry Pi 4B con un procesador Broadcom BCM2711, un Cortex-A72 (ARM v8) de cuatro núcleos [17]. Esta tarjeta se encarga de interactuar con la OpenCR y con los sensores de distancia a la pared (LiDAR o ToF). Además, permite que el TurtleBot se conecte a la red para poder comunicarse con el ordenador desde el que se controla. En este proyecto dicho ordenador tendrá instalado MATLAB.

2.3 DYNAMIXEL XL430-W250-T

Se trata de un servomotor creado por la empresa ROBOTIS [18]. Cuenta con un *encoder* magnético que permite un control 360°. En la Figura 10 se muestra el rendimiento del mismo.

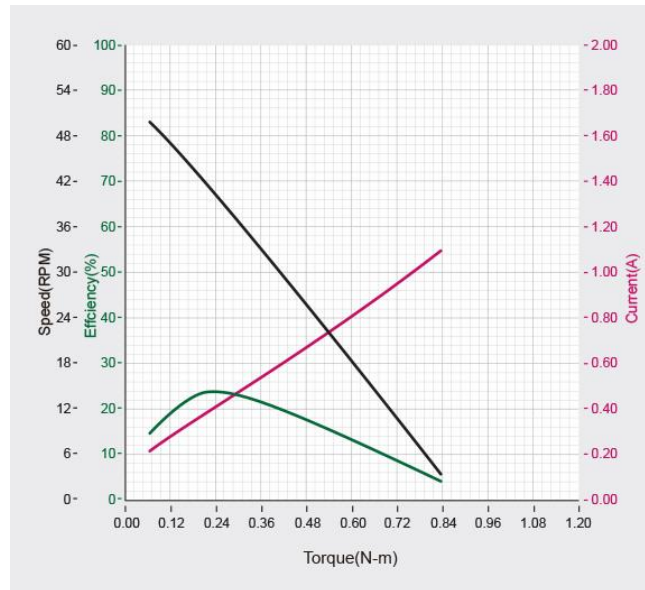


Figura 10. Gráfico de rendimiento del Dynamixel XL430-W250-T [18].

Tiene un microcontrolador ARM CORTEX-M3 para controlar el funcionamiento del motor DC con núcleo. Para poder controlar el dispositivo se utiliza el protocolo Dynamixel 2.0 para leer y escribir de la tabla de control.

Tiene cuatro modos de funcionamiento, que se especifican mediante el valor que se indique en la dirección '11' de la tabla de control del motor:

- Modo control de velocidad ($0^\circ \sim 360^\circ$): se indica con el valor '1' y posibilita el control de la velocidad del motor.
- Modo control de posición: se indica con el valor '3' y es el modo por defecto. Permite el control de la posición del motor entre un rango limitado de una rotación.
- Modo de control de posición extendida (*multi-turn*): se indica con el valor '4' y posibilita el control de la posición del motor, permitiendo un máximo de 512 vueltas.
- Modo de control PWM: se indica con el valor '16' y permite controlar directamente la salida PWM, aplicando un voltaje determinado al motor. Esta es la forma escogida para controlar el motor en este proyecto. Este factor de PWM está limitado a valores entre el rango de 0 (0%) a 885 (100%).

2.4 SENSOR LDS-02

El sensor LDS-02 (ver Figura 11) es un escáner láser 2D que permite recoger información del entorno del robot. Es capaz de detectar 360° y utilizar esos datos para SLAM (*Simultaneous Localization and Mapping*) y tareas de navegación [19].



Figura 11. Sensor LDS-02 [19].

En la Tabla 2 se muestran las especificaciones de rendimiento de las medidas que toma el sensor. Se puede observar que tiene un rango de distancia muy amplio, desde los 0,16 metros hasta los 8 metros. Tiene un rango angular de 360°, con una resolución de 1°. Sin embargo, hay que tener en cuenta que el fabricante especifica que esta resolución puede variar en función de la frecuencia de escaneo, debido al tiempo de muestreo fijo. Para este proyecto la resolución es de 1,4°.

Items	Specifications
Distance Range	160 ~ 8.000mm
Distance Accuracy (160 ~ 300 mm)	±10mm
Distance Accuracy (300 ~ 6.000 mm)	±3,0%
Distance Precision (6.000 ~ 8.000 mm)	±5,0%
Scan Frequency	5Hz or above
Angular Range	360°
Angular Resolution	1°

Tabla 2. Especificaciones del rendimiento de las medidas del LDS-02 [19].

El sensor se conecta a la SBC mediante la interfaz USB (USB2LDS). La información recogida por el LiDAR se publica a través del *topic* /scan en un mensaje de tipo *sensor_msgs/LaserScan* (ver Tabla 6). Para este proyecto se usa la información recogida en el vector *ranges*, que se procesa para obtener la distancia y orientación del robot con respecto a la pared, como se explica más adelante en el Capítulo 5. Control de seguimiento de distancia a la pared.

2.5 SENSOR VL6180X TOF

Debido a las características que tenían las paredes del circuito del laboratorio, las medidas obtenidas por el LiDAR presentaban picos puntuales, causados por los brillos del material del circuito. Esto hacía que los controles PID no funcionasen correctamente, ya que intentaban corregir estas medidas erróneas. Mientras se buscaba la forma de adaptar el

circuito para solucionar este problema se decidió continuar el proyecto diseñando el control de seguimiento de pared con el sensor VL6180X.

Se trata de un sensor de distancia ToF (*Time of Flight*) que contiene una fuente que emite un láser y su correspondiente sensor, que mide el tiempo que tarda en volver el láser [20]. Esto hace que sea más adecuado para medir la distancia justo en frente de él hasta 255mm. En el proyecto se han usado dos de ellos, colocados como se muestra en la Figura 12, de forma que se calcula la distancia desde el punto medio entre ellos hasta la pared. La manera de calcular estas medidas a partir de la información proporcionada por los sensores se explica con más detalle en el Capítulo 5. Control de seguimiento de distancia a la pared. Las comunicaciones con los sensores se realizan mediante I2C. Además, se tiene un dispositivo TCA9548A [21] que permite alternar entre ambos sensores para poder recibir información de cada uno de ellos.

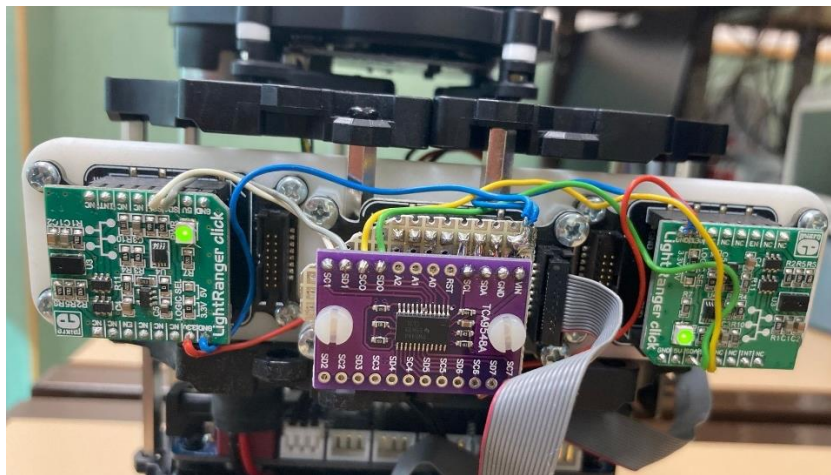


Figura 12. Disposición de los sensores VL6180X ToF en el robot.

Capítulo 3. ARQUITECTURA DE COMUNICACIONES

En la Figura 13 se muestra un esquema de la arquitectura de comunicaciones. Por un lado, se tiene al TurtleBot3 Burger, un robot basado en ROS como se ha descrito en el Capítulo 2. Descripción del *hardware*. Por otro lado, se tiene el ordenador donde está instalado MATLAB, en concreto la *Toolbox* de ROS. Desde el mismo se envían las distintas consignas y parámetros necesarios, además de recoger los resultados de los ensayos para su posterior monitorización. Tanto el robot como el ordenador deben estar conectados a la misma red, para poder enviar y recibir información.

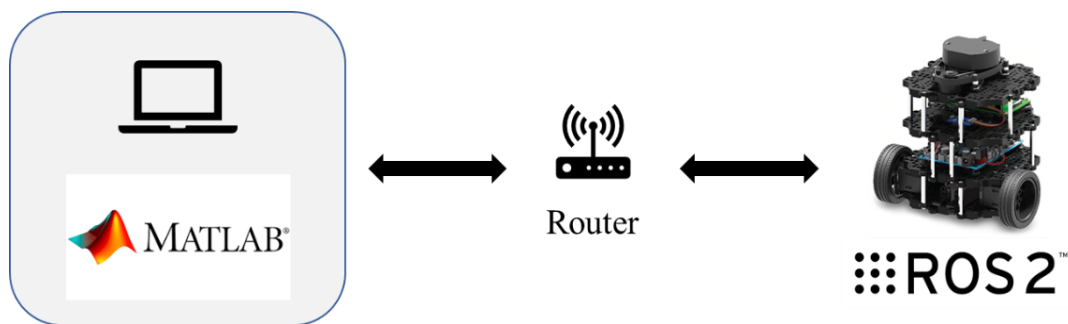


Figura 13. Diagrama de la arquitectura de comunicaciones.

A continuación, se detallan las características de las principales herramientas utilizadas.

3.1 ROS2

ROS2 (*Robot Operating System*) es un conjunto de herramientas y librerías de *software* para el desarrollo de aplicaciones robóticas [22]. Está basado en un mecanismo anónimo de suscripción y publicación, para permitir el paso de mensajes entre los diferentes procesos [23]. La versión de ROS2 que se ha utilizado en este proyecto es Foxy Fitzroy.

Para comprender mejor el funcionamiento de una red de ROS se van a explicar a continuación sus elementos básicos (ver Figura 14), que conforman lo que se conoce como gráfico de nodos o *node graph*.

Nodos

Se representan como elipses dentro del gráfico de nodos y deben ser responsables únicamente de una función (como puede ser la lectura de información de sensores o el control de los motores, por ejemplo). Pueden estar ubicados o no dentro del mismo proceso o incluso estar en máquinas distintas. Los nodos pueden recibir o enviar información a otros nodos a través de *topics*, servicios, parámetros o acciones.

Topics

Se trata de una de las formas de comunicación entre nodos, actuando como un bus para que los nodos puedan intercambiar mensajes. Se representan mediante rectángulos en el gráfico

de nodos. Cada uno de ellos puede estar suscrito a múltiples *topics* y, a su vez, publicar información a otros, ya que esta comunicación no tiene por qué ser uno a uno.

Mensajes

Es la estructura de datos mediante la cual los nodos intercambian información.

Publicadores

Es un elemento que se encarga de mandar un mensaje a un *topic*. En el gráfico de nodos se representa como una flecha desde un nodo hasta un *topic*.

Suscriptores

Es el elemento que se encarga de recibir los mensajes que han sido publicados en un *topic*. Si un nodo quiere recibir la información que hay en un *topic* tiene que estar suscrito al mismo. En el gráfico de nodos se representan como una flecha que va desde un *topic* hasta un nodo.

Parámetros

Están asociados a nodos de forma individual. Permiten establecer un valor de configuración del nodo, que dura lo que dure la vida del nodo, sin necesidad de cambiar el código.

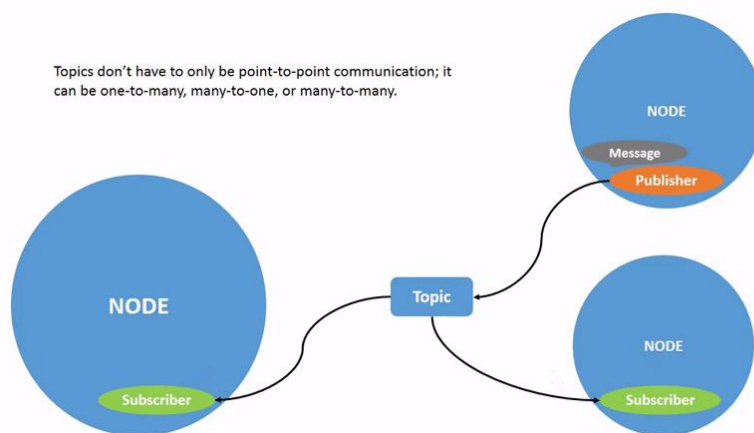


Figura 14. Modo de funcionamiento de los elementos de una red de ROS2 [24].

En la Figura 15 y la Figura 16 se muestra un ejemplo del gráfico de nodos del robot. La diferencia entre ambas figuras es el sensor que se ha utilizado para medir la distancia a la pared. Se puede observar que solo está activo y publicando medidas en el *topic* /lidar el nodo del sensor que se esté utilizando en ese momento (el LiDAR o el ToF). En el gráfico de la Figura 15 se puede ver que hay siete nodos (elipses), de los cuales /WallFollowerNode y /Envio_referencia_90580 pertenecen al control de seguimiento de pared implementado y se explicarán con más detalle en el Capítulo 5. Control de seguimiento de distancia a la pared. El resto de los nodos que crea el robot son:

- /ld08_driver: publica las medidas del LiDAR a través del *topic* /scan.

- `/lidar_node`: está suscrito al *topic* `/scan` a través del cual recibe las medidas del LiDAR y las procesa según la trigonometría que se explica en el Capítulo 5. Control de seguimiento de distancia a la pared, para estimar la distancia y el ángulo a la pared.
- `/tof_node`: recibe las medidas del sensor de distancia ToF y realiza los cálculos necesarios, según se explica en el Capítulo 5. Control de seguimiento de distancia a la pared, para estimar el ángulo y la distancia a la pared en caso de usar el sensor ToF.
- `/turtlebot3_node`: se encarga de recibir las consignas de PWM que tiene que enviar a los motores a través del *topic* `/cmd_vel`.
- `/diff_drive_controller`: está suscrito a los *topics* `/imu` y `/joint_states`. Estos *topics* se explican en más detalle más adelante.
- `/robot_state_publisher`: está suscrito a `/joint_states` y permite publicar el estado del robot.

Además de los siete nodos hay ocho *topics* (rectángulos). Tres de ellos (`/measurements`, `/control_reference_dn` y `/control_on`) se han creado a la hora de implementar el control de distancia a la pared y se explican en el Capítulo 5. Control de seguimiento de distancia a la pared. Los otros cinco *topics* son:

- `/imu`: recibe un mensaje de tipo `sensor_msgs/Imu` (ver Tabla 3) con la información sobre la aceleración lineal y la velocidad angular del robot.

Parámetro	Información
<code>std_msgs/Header header</code>	Marca de tiempo y <code>frame_id</code>
<code>geometry_msgs/Quaternion orientation</code>	Orientación
<code>float64[9] orientation_covariance</code>	Covarianza de la orientación
<code>geometry_msgs/Vector3 angular_velocity</code>	Velocidad angular (rad/s)
	x
	y
	z
<code>float64[9] angular_velocity_covariance</code>	Covarianza de la velocidad angular
<code>geometry_msgs/Vector3 linear_acceleration</code>	Aceleración (m/s ²)
	x
	y
	z
<code>float64[9] linear_acceleration_covariance</code>	Covarianza de la aceleración

Tabla 3. Estructura del mensaje publicado en el *topic* `/imu`.

- `/joint_states`: publica un mensaje de tipo `sensor_msgs/JointState` (ver Tabla 4) con información sobre la velocidad y ángulo girado de cada motor.

Parámetro	Información
std_msgs/Header header	Marca de tiempo y frame_id
string [] name	Nombre
float64 [] position	Posición (rad o m)
float64 [] velocity	Velocidad (rad/s o m/s)
float64 [] effort	Esfuerzo aplicado (Nm o N)

Tabla 4. Estructura del mensaje publicado en el topic /joint_states.

- /cmd_vel: contiene un mensaje de tipo geometry_msgs/Twist (ver Tabla 5) con las consignas de tensión común y diferencial del robot.

Parámetro	Información
geometry_msgs/Vector3 linear	
linear.x	Tensión común
linear.y	
linear.z	
geometry_msgs/Vector3 angular	
angular.x	Tensión diferencial
angular.y	
angular.z	

Tabla 5. Estructura del mensaje publicado en el topic /cmd_vel.

- /scan: contiene un mensaje de tipo sensor_msgs/LaserScan (ver Tabla 6) con la información del entorno recogida por el LiDAR.

Parámetro	Información
std_msgs/Header header	Marca de tiempo y frame_id
float32 angle_min	Ángulo inicial del escaneo (rad)
float32 angle_max	Ángulo final del escaneo (rad)
float32 angle_increment	Distancia angular entre las muestras (rad)
float32 time_increment	Tiempo entre las medidas (segundos)
float32 scan_time	Tiempo entre escaneos (segundos)
float32 range_min	Valor mínimo del rango (metros)
float32 range_max	Valor máximo del rango (metros)
float32[] ranges	Datos de medida (metros)
float32[] intensities	Información de la intensidad

Tabla 6. Estructura del mensaje publicado en el topic /scan.

- /lidar: contiene un mensaje de tipo geometry_msgs/TwistStamped (ver Tabla 7) con la medida de distancia y ángulo. Esta información se calcula en el nodo /tof_node o en el /lidar_node, en función de que sensor se desee utilizar.

Parámetro	Información
std_msgs/Header header	Marca de tiempo y frame_id
geometry_msgs/Twist linear	
twist.linear.x	Distancia medida (metros)
twist.linear.y	
twist.linear.z	
geometry_msgs/Twist angular	
twist.angular.x	Ángulo medido (rad)
twist.angular.y	
twist.angular.z	

Tabla 7. Estructura del mensaje publicado en el topic /lidar.

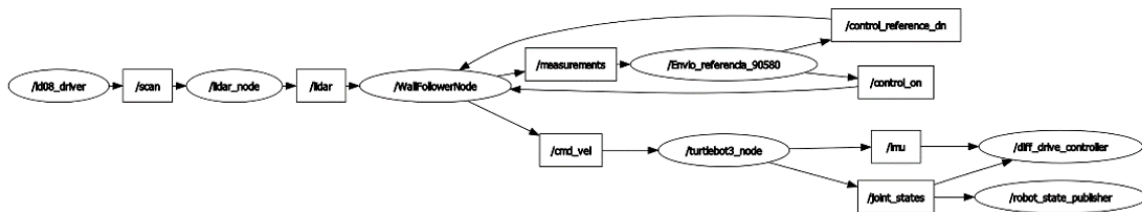


Figura 15. Ejemplo de gráfico de nodos generado con rqt con el LiDAR.

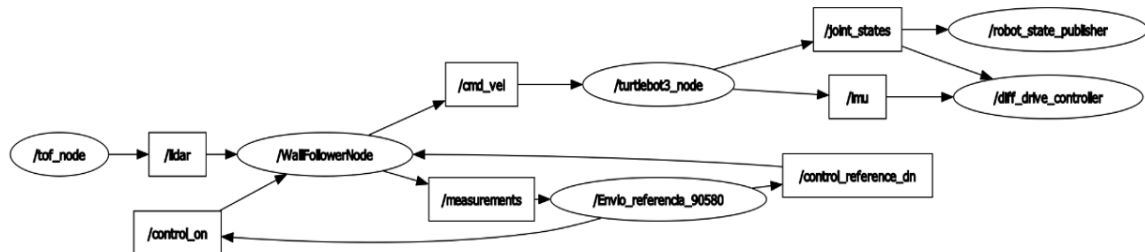


Figura 16. Ejemplo de gráfico de nodos generado con rqt con el sensor ToF.

3.2 ROS2 TOOLBOX DE MATLAB

La ROS2 *Toolbox* de MATLAB es uno de los productos que proporciona una interfaz para poder conectar MATLAB y Simulink con ROS y ROS2. Esta *Toolbox* incluye funciones de MATLAB y bloques de Simulink que permiten diseñar una red de nodos de ROS [25].

En la Figura 17 se muestra que MATLAB permite varias formas de conexión con ROS2.

- Rosbag *import*: mediante este método MATLAB puede leer los ficheros *rosvbag* y extraer la información contenida en los mensajes. Estos ficheros se crean al suscribirse a los distintos *topics* de ROS y almacenando la información recibida [26].
- Conexión con ROS: este método permite conectarse a una red de ROS desde el entorno de MATLAB y enviar y recibir la información necesaria mediante *publishers*, *subscribers* y servicios. Este proceso se puede hacer tanto desde MATLAB

como desde Simulink. En el primer caso se deben crear los nodos de forma manual en el código de MATLAB y establecer los *publishers* y *suscribers* necesarios. Al terminar se deben incluir los comandos para parar los nodos creados [27] [4]. En el segundo caso, Simulink cuenta con una serie de bloques específicos que permiten publicar y suscribirse a distintos *topics*, para poder intercambiar mensajes. Al simular el archivo se genera un nodo con el nombre del modelo y los *publishers* y *suscribers* asociados. Todo ello se elimina al terminar la simulación [28].

- Generación de nodos ROS: este método permite la creación de una implementación independiente de MATLAB, a través de la generación de código en C++ de los nodos de ROS y desplegarlo en el *hardware*. Este método funciona tanto para archivos de MATLAB como de Simulink, ya que se puede generar el código mediante MATLAB Coder y Simulink Coder [5].

Finalmente se ha decidido utilizar una combinación de los dos últimos métodos, ya que los ensayos realizados de esta manera tenían un mejor desempeño. Por un lado, se utiliza la generación de nodos ROS para desplegar el control PID en el robot. Por otro lado, se hace uso de la conexión con ROS para mandar los parámetros y consignas necesarios y realizar la monitorización.

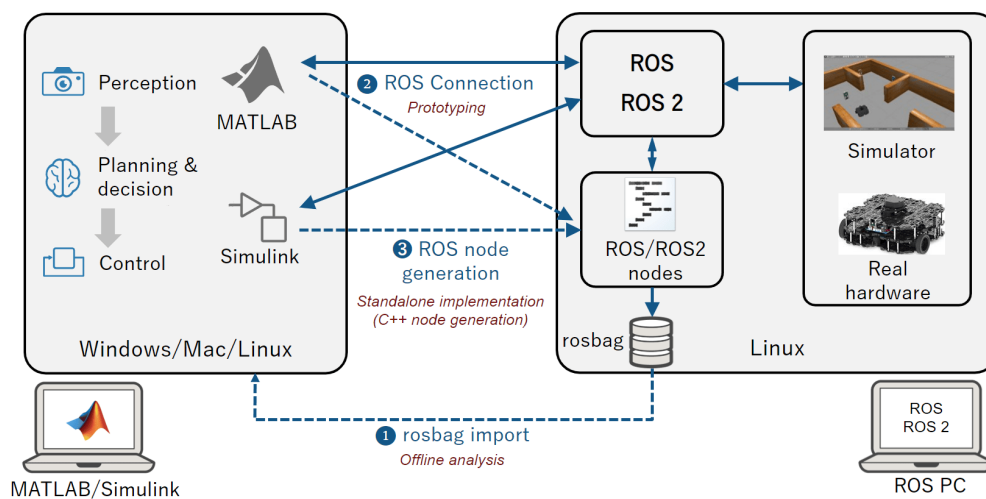


Figura 17. Conexión con la ROS2 Toolbox de MATLAB [25].

Además de la ROS *Toolbox* de MATLAB es necesario tener instalados los siguientes productos:

- Control System Toolbox.
- DSP System Toolbox.
- Embedded Coder.
- MATLAB Coder.
- MATLAB Compiler.
- Model Predictive Control Toolbox.
- Optimization Toolbox.

- Signal Processing Toolbox.
- Simulink Coder.
- Simulink Compiler.
- Simulink Desktop Real-Time.
- Symbolic Math Toolbox.
- System Identification Toolbox.

Capítulo 4. CONTROL DE VELOCIDAD DE AVANCE

En este capítulo se van a diseñar de forma teórica los distintos reguladores para el control de velocidad. Posteriormente se compararán los resultados obtenidos en simulaciones con los ensayos reales.

4.1 IDENTIFICACIÓN DEL MODELO DE LA PLANTA

Para poder diseñar los distintos controles es necesario determinar primero el modelo de la planta del sistema que se quiere controlar. Para ello se han realizado una serie de ensayos en lazo abierto y en lazo cerrado para identificar el modelo más adecuado, mediante un ajuste por mínimos cuadrados. La entrada del sistema es el mando (PWM) y la salida es la velocidad medida en rad/s.

La función de transferencia que se quiere modelar, según la física del problema y obviando la constante de tiempo de la parte eléctrica, es:

$$P(s) = \frac{V(s)}{U(s)} = \frac{K_m}{T_m \cdot s + 1}$$

Además, se ha visto en los ensayos realizados que el modelo de la planta cuenta con un retardo de 120ms debido a los retrasos en las comunicaciones entre los distintos elementos de la red. Se ha repetido varias veces el ensayo para poder determinar si este valor se mantenía constante en todos. En la Figura 18 se muestra una comparativa de los ensayos realizados, observándose repetibilidad en cuanto al valor de este retardo.

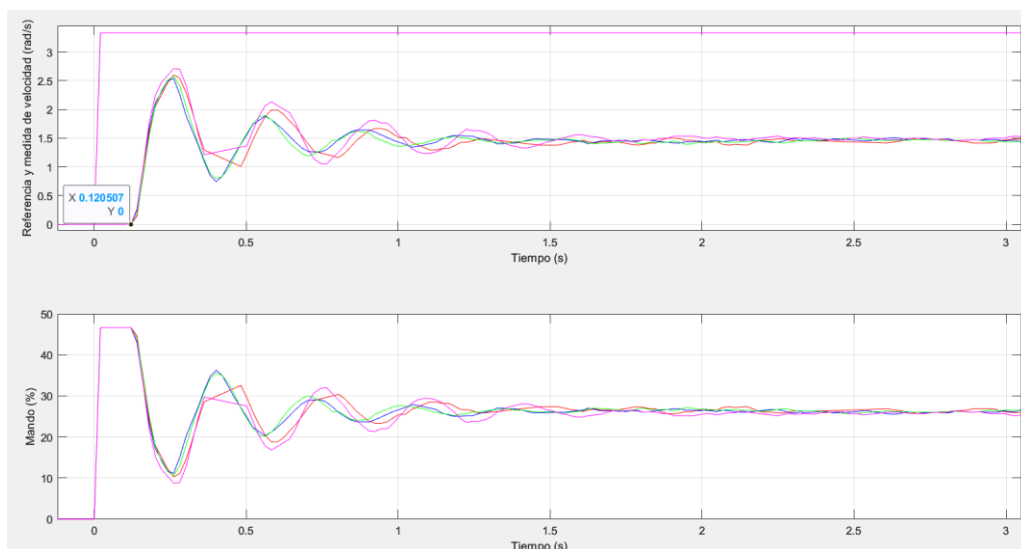


Figura 18. Comparativa ensayos.

En la Figura 20 se muestra el resultado del proceso de identificación, realizado con un ensayo con un control proporcional con $K = 12$ (ver Figura 19). Se puede observar que la respuesta simulada captura bien la respuesta real, especialmente en las primeras oscilaciones. Teniendo en cuenta todo lo anterior, el modelo de la planta obtenido es:

$$P(s) = \frac{0,055}{0,040 \cdot s + 1} \cdot e^{-0,12s}$$

El modelo anterior se obtiene usando la velocidad en rad/s, si se desea utilizar la velocidad en m/s el modelo resultante sería:

$$P(s) = \frac{0,0018}{0,040 \cdot s + 1} \cdot e^{-0,12s}$$

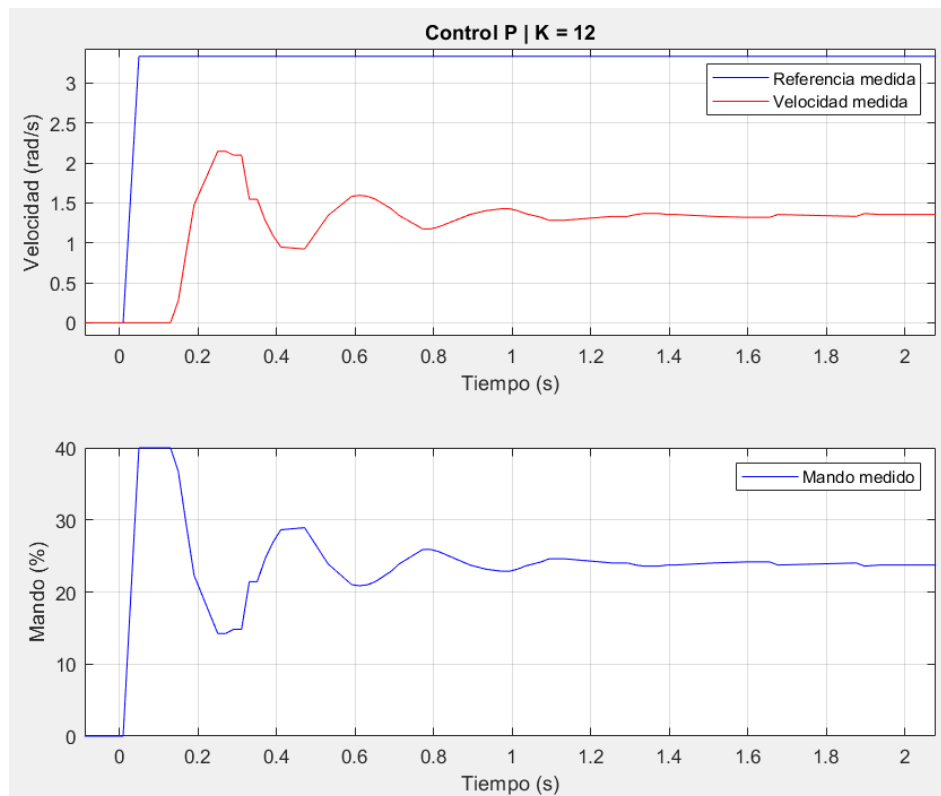


Figura 19. Ensayo utilizado para la identificación de la planta del control de velocidad.

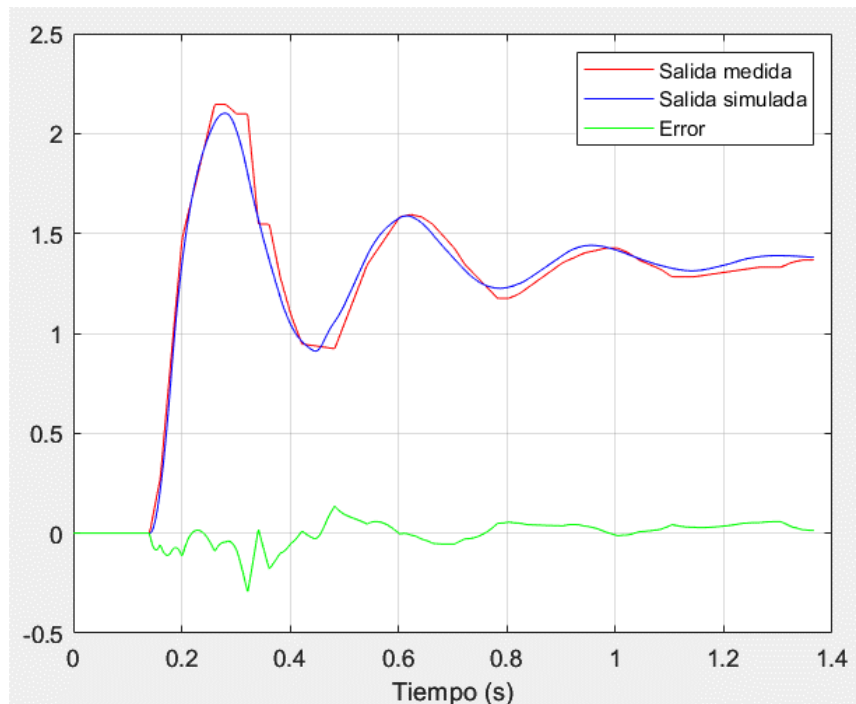


Figura 20. Resultado de la identificación de la planta del control de velocidad.

4.2 DISEÑO DE LOS CONTROLES PID

Con el modelo de la planta obtenido en el apartado anterior se van a diseñar ahora los distintos controles.

En la Figura 21 se muestra un diagrama de bloques del lazo cerrado del sistema que se quiere diseñar. La salida $y(t)$ es la velocidad medida, la referencia $r(t)$ es la velocidad deseada, la variable de control o mando $u(t)$ es el % de PWM aplicado a los motores y la perturbación $d(t)$ corresponde al par de fricción debido a la rampa del circuito. Los bloques $C(s)$ y $Cr(s)$ son las funciones de transferencia de los controles, que se especifican más adelante.

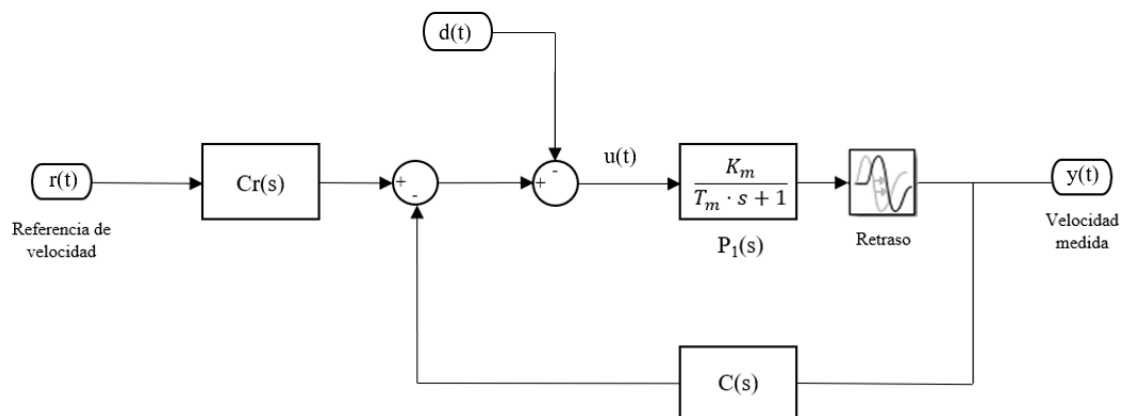


Figura 21. Diagrama de bloques del sistema.

En la Figura 22 se muestra el gráfico de Black de la planta modelada. Se puede observar que tiene una frecuencia de oscilación de 20,5 rad/s. Los reguladores se van a diseñar mediante técnicas basadas en la respuesta en frecuencia, utilizando el formato interactivo o en serie. Posteriormente se transformarán esas expresiones al formato no interactivo o paralelo, ya que es el formato con el que se han implementado los controles en el robot. El formato paralelo permite usar controles de dos grados de libertad, en particular la ponderación de la referencia y/o la posibilidad de poner la acción derivativa sobre la salida en lugar de sobre el error. Para diseñar por respuesta en frecuencia se puede emplear el margen de fase (ϕ_m) o el margen de ganancia (A_m). Debido a la “horizontalidad” del gráfico se van a diseñar por el último mencionado, al ser más restrictivo.

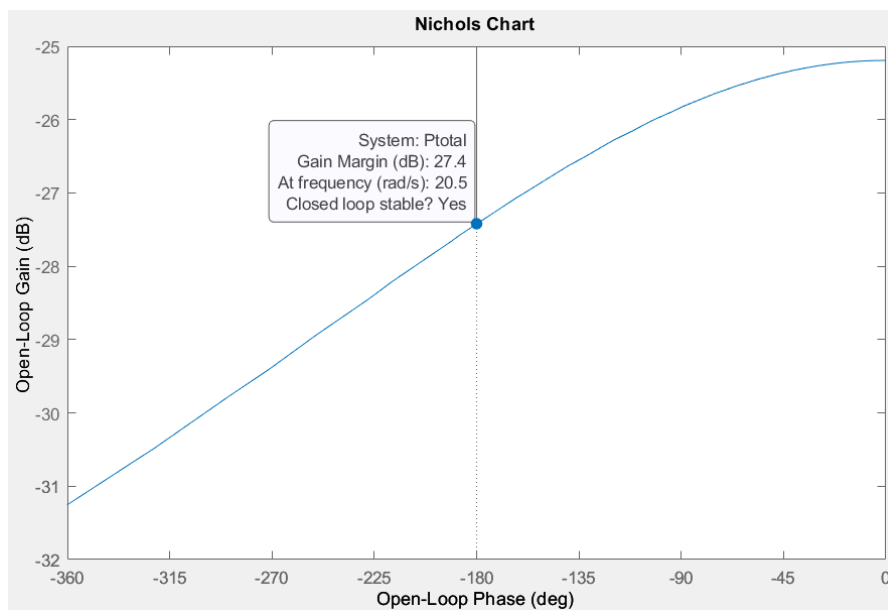


Figura 22. Gráfica de Black para la planta del control de velocidad.

El margen de ganancia es el factor por el cual habría que multiplicar el lazo abierto para obtener un sistema oscilante [29].

4.2.1 CONTROL PROPORCIONAL, P

En este control únicamente se puede especificar el amortiguamiento. Como se indica en la Ecuación 1, este control solo cuenta con un grado de libertad, la ganancia. Al aumentar el valor de K , a valores mayores que 1, se desplaza el gráfico de Black hacia arriba, haciendo que la respuesta sea más rápida. Sin embargo, a la hora de escoger este parámetro es necesario que exista un compromiso entre el amortiguamiento (que puede verse comprometido por valores demasiado altos de K) y la rapidez. Esta última también se ve afectada por incrementos en la ganancia (tiempo de pico y tiempo de alcance).

$$C(s) = K_p$$

Ecuación 1. Control proporcional (formato serie).

Para el diseño del regulador proporcional mediante técnicas de respuesta en frecuencia se han seguido los siguientes pasos:

1. Establecer el margen de ganancia $A_{m\text{ dB}} > 6\text{ dB}$.
2. Calcular la pulsación de oscilación w_u .

$$\angle G(jw_u) = -180^\circ$$

3. Calcular la ganancia K .
 - a. En reales

$$A_m \cdot |G(jw_u)| = 1 \rightarrow A_m \cdot |C(jw_u)| \cdot |P(jw_u)| = 1 \rightarrow K = \frac{1}{A_m \cdot |P(jw_u)|}$$

- b. En dB

$$A_{m\text{ dB}} + |G(jw_u)|_{\text{dB}} = 0\text{ dB} \rightarrow A_{m\text{ dB}} + K_{\text{dB}} + |P(jw_u)|_{\text{dB}} = 0 \rightarrow$$

$$K_{\text{dB}} = -A_{m\text{ dB}} - |P(jw_u)|_{\text{dB}}$$

Finalmente, el control obtenido para un margen de ganancia de 7 dB tiene una ganancia K de 10,5. Para dicho valor de ganancia se obtiene un tiempo de pico de 0,246 segundos con un sobrepaso del 50,2%, y un tiempo de establecimiento al 5% de 0,604 segundos. El sobrepaso obtenido es bastante elevado, sin embargo, se ha decidido mantener este diseño para no reducir la ganancia. Otra de las razones por las que no se ha cambiado el control obtenido es para tener la misma especificación en todos los diseños. En el resto de los controles diseñados no se ha obtenido un sobrepaso tan elevado. El gráfico de Black de la planta $P(s)$ y la función de transferencia en lazo abierto $G(s)$ se muestran en la Figura 23, donde se comprueba de nuevo la horizontalidad de ambas funciones a causa del retardo. La respuesta que tendría el sistema con este control frente a un escalón unitario en la referencia y en la perturbación se puede observar en la Figura 24. En ella se comprueba la relevancia que tiene el retardo de la planta en comparación con los tiempos de respuesta obtenidos.

Además, también se observa que en régimen permanente no se alcanza el valor deseado, quedando por debajo, como es de esperar con este tipo de control si la planta no tiene integraciones.

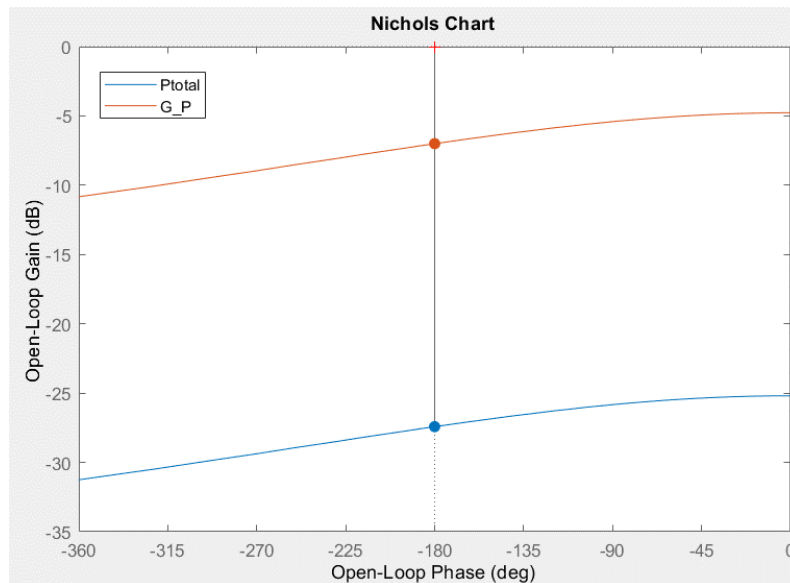


Figura 23. Gráfico de Black para la planta $P(s)$ y el lazo abierto $G(s)$ para el control P .

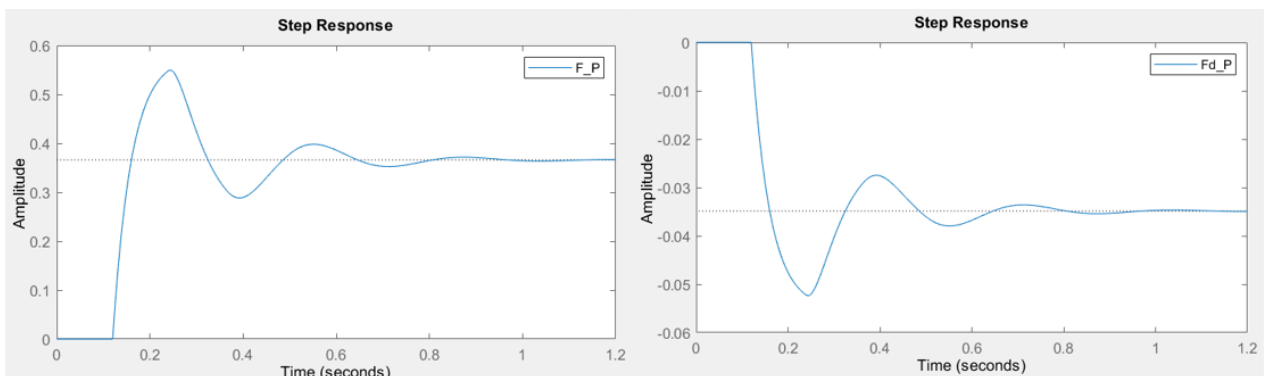


Figura 24. Respuesta a un escalón unitario en referencia y perturbación para el control P .

4.2.2 CONTROL PROPORCIONAL MÁS INTEGRAL, PI

El objetivo de este control es mejorar la precisión del sistema, anulando el error en régimen permanente tanto en seguimiento de referencia como en el rechazo a una perturbación constante. Sin embargo, la rapidez de la respuesta puede verse afectada. En la Ecuación 2 se muestra la función de transferencia para el control PI. Se observa que en este caso sí que existe un retraso en la fase, un desplazamiento del gráfico de Black hacia la izquierda. Valores típicos de este retraso de fase están entre -5° y -15° , aunque pueden ser mayores. Al igual que en el control P , se produce también un desplazamiento hacia arriba para K mayores que 1. Este efecto hace referencia solo al efecto que provoca la ganancia, ya que el efecto de la parte integral también produce un desplazamiento hacia arriba, aunque menor.

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

Ecuación 2. Función de transferencia para el control PI (formato serie).

Para el diseño de este tipo de controles por respuesta en frecuencia lo habitual es fijar el amortiguamiento y la rapidez deseada, menor que la del P para poder ajustar el retraso de fase del control, siguiendo los pasos que se describen a continuación:

1. Establecer el margen de ganancia $A_m \text{ dB} > 6\text{dB}$.
2. Definir la pulsación de oscilación como $w_u < w_{uP}$. Otra alternativa es fijar el retraso de fase aportado por la parte integral del control.
3. Calcular la fase del control.

$$\angle G(jw_u) = \angle C(jw_u) + \angle P(jw_u) = -180^\circ$$

$$\varphi_C = -180^\circ - \angle P(jw_u)$$

4. Calcular la ganancia del control.

- a. En reales

$$A_m \cdot |G(jw_u)| = 1 \rightarrow A_m \cdot |C(jw_u)| \cdot |P(jw_u)| = 1 \rightarrow A_C = \frac{1}{A_m \cdot |P(jw_u)|}$$

- b. En dB

$$A_m \text{ dB} + |G(jw_u)|_{\text{dB}} = 0 \text{ dB} \rightarrow A_m \text{ dB} + A_C \text{ dB} + |P(jw_u)|_{\text{dB}} = 0$$

$$A_C \text{ dB} = -A_m \text{ dB} - |P(jw_u)|_{\text{dB}}$$

5. Calcular los parámetros K_P e I del control.

$$|C(jw_u)| \cdot e^{\angle C(jw_u)} = A_C \cdot e^{j\varphi_C} = K_P \cdot \frac{1 + jw_u I}{jw_u I}$$

$$\varphi_C = -90^\circ + \text{atan}(Iw_u) \rightarrow I = \frac{\tan(90^\circ + \varphi_C)}{w_u}$$

$$A_C = K_P \cdot \frac{\sqrt{1 + (Iw_u)^2}}{Iw_u} \rightarrow K_P = \frac{A_C I w_u}{\sqrt{1 + (Iw_u)^2}}$$

Escogiendo una $w_u = 0,8 \cdot w_{uP}$ se obtiene una fase del control de $-34,22^\circ$, es superior al límite recomendado de -15° . Sin embargo, esto no es de extrañar en sistemas con un retardo como el que se ha modelado. La respuesta obtenida frente a un escalón de referencia se puede ajustar si se aplica una ponderación a la referencia, modificando el cero de la función de transferencia de la salida. Para valores de $b < 1$ se aplica menos referencia, por lo que la respuesta debería tener un pico menor. Lo contrario ocurre si se aumenta el valor de b . Para un margen de ganancia de 7 dB se obtiene una ganancia K de 8,0 y un T_i igual a 0,09 segundos.

En la Figura 25 se muestra el gráfico de Black de la función de transferencia del lazo abierto ($G(s)$) y de la planta ($P(s)$), en ella se comprueba que el margen de fase obtenido en el diseño es de $66,9^\circ$, por lo que el margen limitante es con el que se ha diseñado, el de ganancia. Finalmente se escoge una ponderación $b = 1,5$, obteniéndose un sobrepaso del 15,0 % y un tiempo de pico de 0,298 segundos, algo mayor que el del control P. Sin embargo, pese a ser más lento, con el control PI sí se alcanza el valor de la referencia en régimen permanente, como se observa en la Figura 26.

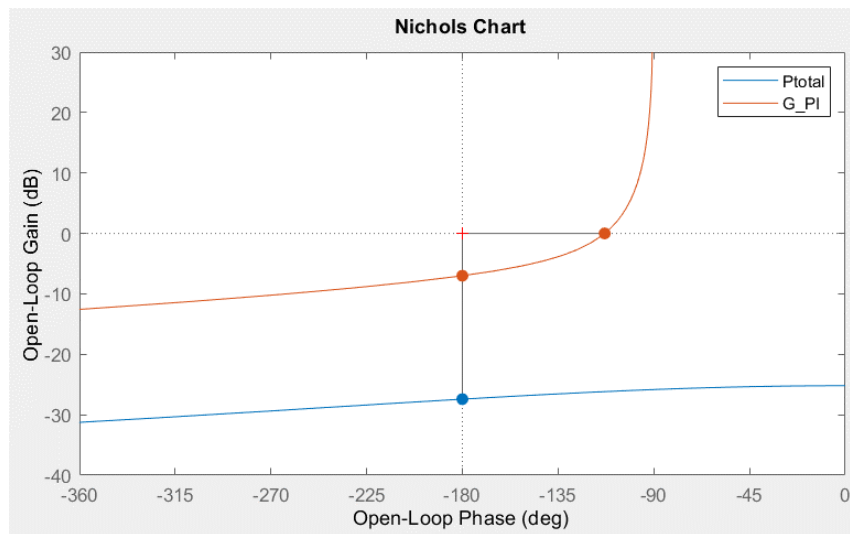


Figura 25. Gráfico de Black para la planta $P(s)$ y el lazo abierto $G(s)$ para el control PI.

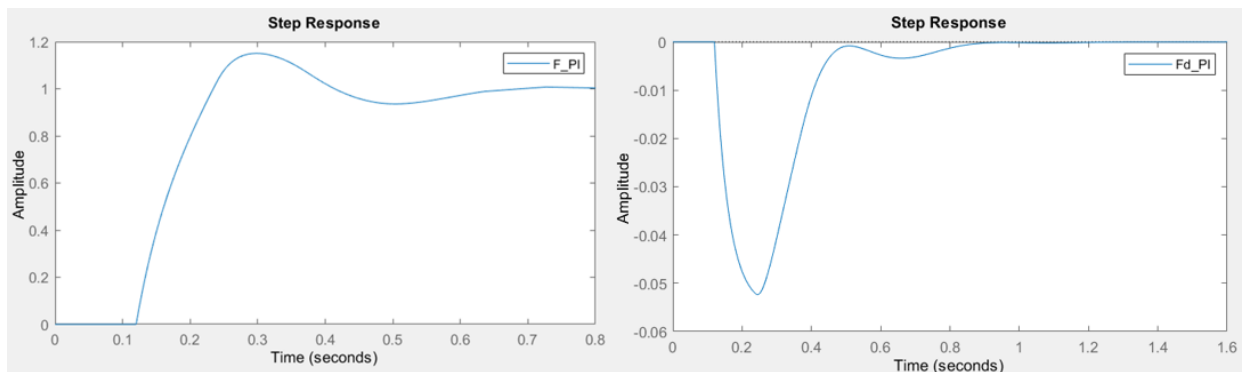


Figura 26. Respuesta a un escalón unitario en referencia y perturbación para el control PI.

4.2.3 CONTROL PROPORCIONAL DIFERENCIAL, PD

El objetivo de este control es mejorar la respuesta transitoria (la rapidez, el amortiguamiento o ambas) y mejorar la precisión del control con respecto al proporcional, ya que se suelen obtener unos valores mayores de K_p . En la Ecuación 3 se muestra la función de transferencia del control, donde f es el factor de filtrado. Se observa que presenta una fase positiva ($\pi/2 > \varphi_c > 0$), que provoca un desplazamiento hacia la derecha en el gráfico de Black.

$$C(s) = K_P \cdot \frac{1 + Ds}{1 + fDs}$$

Ecuación 3. Función de transferencia para el control PD (formato serie).

Los pasos a seguir para diseñar el control por respuesta en frecuencia son:

1. Establecer el margen de ganancia $A_m \text{ dB} > 6\text{dB}$.
2. Determinar la pulsación de oscilación $w_u > w_{uP}$ y fijar el factor de filtrado $0 < f < f_M$.

Este factor de filtrado máximo se calcula como:

$$f_M = \frac{1 - \text{sen}(\varphi_C)}{1 + \text{sen}(\varphi_C)}$$

3. Calcular la fase del control.

$$\angle G(jw_u) = \angle C(jw_u) + \angle P(jw_u) = -180^\circ$$

$$\varphi_C = -180^\circ - \angle P(jw_u)$$

4. Calcular la ganancia del control.

- a. En reales

$$A_m \cdot |G(jw_u)| = 1 \rightarrow A_m \cdot |C(jw_u)| \cdot |P(jw_u)| = 1 \rightarrow A_C = \frac{1}{A_m \cdot |P(jw_u)|}$$

- b. En dB

$$A_m \text{ dB} + |G(jw_u)|_{\text{dB}} = 0 \text{ dB} \rightarrow A_m \text{ dB} + A_C \text{ dB} + |P(jw_u)|_{\text{dB}} = 0 \rightarrow$$

$$A_C \text{ dB} = -A_m \text{ dB} - |P(jw_u)|_{\text{dB}}$$

5. Calcular los parámetros del control.

$$w_u D = \frac{\frac{1}{f} - 1}{2 \text{tg}(\varphi_C)} - \sqrt{\left(\frac{\frac{1}{f} - 1}{2 \text{tg}(\varphi_C)}\right)^2 - \frac{1}{f}}$$

$$K_P = A_C \frac{\sqrt{1 + (fDw_u)^2}}{\sqrt{1 + (w_u D)^2}}$$

Con una w_u un 10% mayor a la del control P y un margen de ganancia de 7 dB se obtienen los siguientes parámetros: $K = 10,36$, $T_d = 0,0135$ y $N = 9$. Con este control se obtiene una mayor rapidez que con los dos controles anteriores con un tiempo de establecimiento al 5%

de 0,546 segundos y un tiempo de pico de 0,238 segundos con un sobrepaso del 51,2%. Sin embargo, a diferencia del control PI, en el control PD no se alcanza el valor de la referencia en régimen permanente (ver Figura 28).

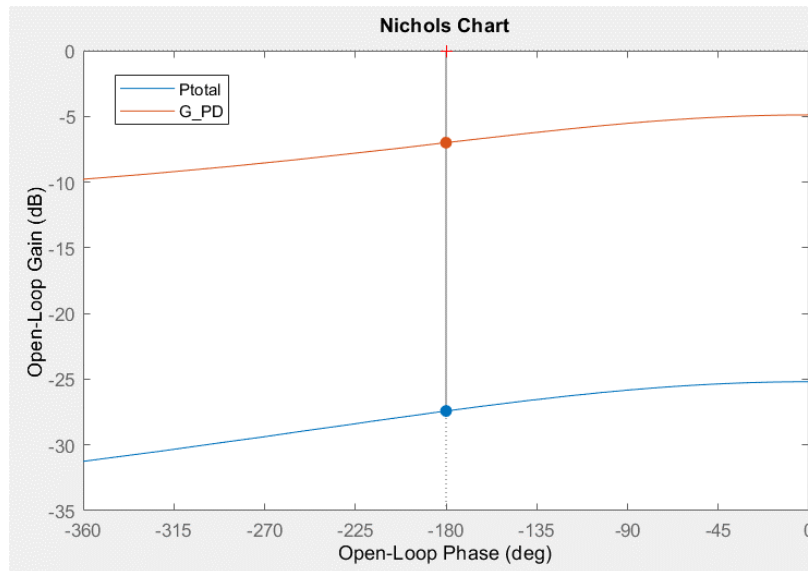


Figura 27. Gráfico de Black para la planta $P(s)$ y el lazo abierto $G(s)$ para el control PD.

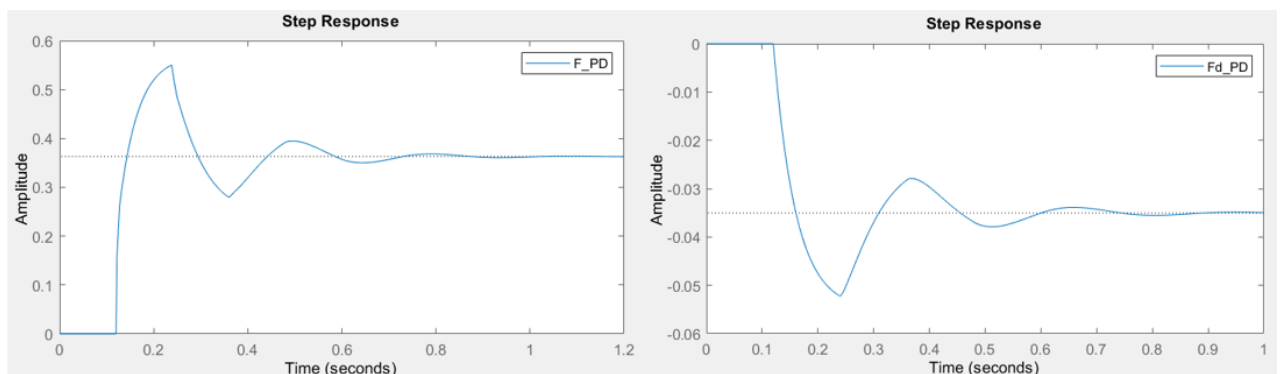


Figura 28. Respuesta a un escalón unitario en referencia y perturbación para el control PD.

4.2.4 CONTROL PROPORCIONAL INTEGRAL DIFERENCIAL, PID

El objetivo de este control es combinar las ventajas de los anteriores. Por un lado, se obtiene una mayor rapidez que con el control P gracias a la acción diferencial y, por otro lado, se consigue eliminar el error en régimen permanente gracias a la acción integral. En la Ecuación 4 se muestra la función de transferencia para este control. Debido a la combinación de ambas acciones se obtiene un adelanto de fase neto, ya que el retraso de la parte integral se ve compensado por el adelanto de la diferencial.

$$C(s) = K_p \cdot \frac{1 + Is}{Is} \cdot \frac{1 + Ds}{1 + fDs}$$

Ecuación 4. Función de transferencia para el control PID (formato serie).

Los pasos a seguir para diseñar el control por respuesta en frecuencia son:

1. Establecer el margen de ganancia $A_m \text{ dB} > 6\text{dB}$.
2. Establecer el retraso de fase asociado a la parte integral (φ_R).
3. Fijar la pulsación de oscilación w_u y el factor de filtrado $0 < f < f_M$. Este factor de filtrado máximo se calcula como:

$$f_M = \frac{1 - \text{sen}(\varphi_A)}{1 + \text{sen}(\varphi_A)}$$

4. Calcular la fase del control (φ_C) y el adelanto de fase de la parte diferencial (φ_A).

$$\angle G(jw_u) = \angle C(jw_u) + \angle P(jw_u) = -180^\circ$$

$$\varphi_C = -180^\circ - \angle P(jw_u)$$

$$\varphi_A = \varphi_C - \varphi_R$$

5. Calcular la ganancia del control A_c .

$$A_m \cdot |G(jw_u)| = A_m \cdot |C(jw_u)| \cdot |P(jw_u)| = 1 \rightarrow A_c = \frac{1}{A_m \cdot |P(jw_u)|}$$

$$|G(jw_u)|_{dB} = A_m \text{ dB} + |C(jw_u)|_{dB} + |P(jw_u)|_{dB} = 0 \rightarrow$$

$$A_c \text{ dB} = -A_m \text{ dB} - |P(jw_u)|_{dB}$$

6. Calcular el parámetro I .

$$\varphi_R = -90^\circ + \text{atan}(Iw_u) \rightarrow I = \frac{\tan(90^\circ + \varphi_R)}{w_u}$$

7. Calcular los parámetros D y K_P del control.

$$w_u D = \frac{\frac{1}{f} - 1}{2 \text{tg}(\varphi_A)} - \sqrt{\left(\frac{\frac{1}{f} - 1}{2 \text{tg}(\varphi_A)}\right)^2 - \frac{1}{f}}$$

$$K_P = A_c \cdot \frac{\sqrt{1 + (fDw_u)^2}}{\sqrt{1 + (wD)^2}} \cdot \frac{Iw_u}{\sqrt{1 + (Iw_u)^2}}$$

Escogiendo una $w_u = 1,12 \cdot w_{uP}$, un margen de ganancia de 7 dB, un factor de filtrado de 0,1 y un retraso de fase de la parte integral de -20° se obtienen los siguientes parámetros del control: $K = 9,70$, $T_i = 0,16$, $T_d = 0,029$, $N = 6,5$ y $b = 1,6$.

En la Figura 29 se muestra el gráfico de Black de la función de transferencia de la planta $P(s)$ y del lazo abierto $G(s)$. En ella se observa que el margen de fase obtenido es de $88,7^\circ$, con ello se comprueba que este margen de estabilidad, no usado para diseñar, es aceptable.

Por otro lado, en la Figura 30 se muestra la respuesta frente a un escalón unitario en referencia y perturbación. Se comprueba que se cumplen los dos objetivos principales del control. En primer lugar, no existe error en régimen permanente gracias a la parte integral. Además, la respuesta es más rápida gracias a la parte diferencial. Sin embargo, debido a las características de la planta modelada el aumento de velocidad obtenido no es demasiado significativo. Se obtiene un valor de sobrepaso del 8,3%, alcanzando el pico en 0,241 segundos y con un tiempo de establecimiento al 5% de 0,674 segundos.

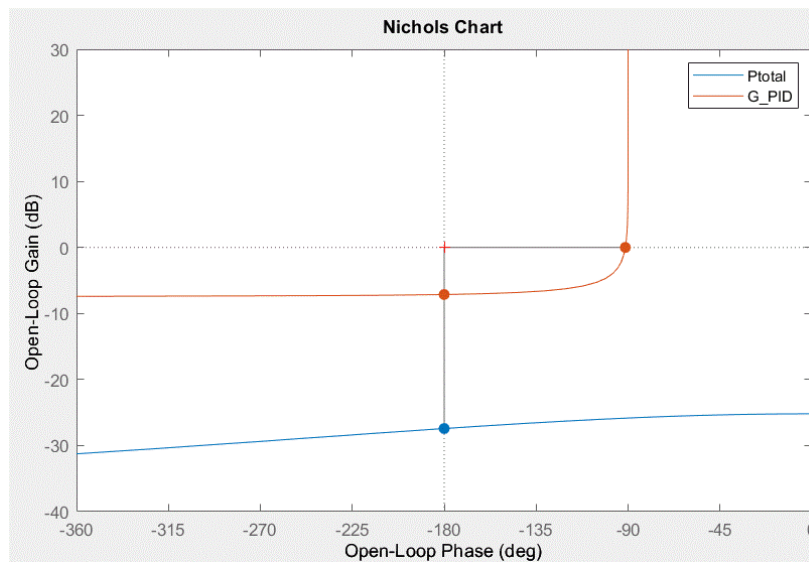


Figura 29. Gráfico de Black para la planta $P(s)$ y el lazo abierto $G(s)$ para el control PID.

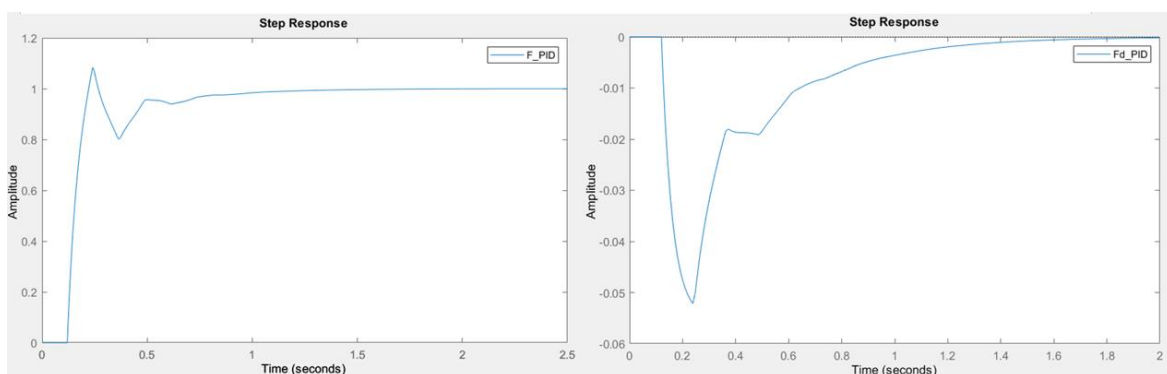


Figura 30. Respuesta a un escalón unitario en referencia y perturbación para el control PID.

En la Tabla 8 se muestra una comparativa de los parámetros obtenidos para el diseño con un margen de ganancia de 7 dB. En la Figura 31 y en la Tabla 9 se muestra una comparación de la respuesta a un escalón unitario en referencia. Se comprueba que con el control PI se ha conseguido un error en régimen permanente nulo, con un sobrepaso aceptable. En el control P y PD se han obtenido sobrepasos mayores; sin embargo, existe error en régimen

permanente en seguimiento a la referencia. La ventaja del control PD frente al P es un aumento de la rapidez, pero, debido a las características de la planta y al retardo, no se alcanza un incremento significativo de la misma. Por último, con el control PID se consigue error nulo en régimen permanente, pero, de nuevo, no se obtiene una rapidez demasiado significativa.

Por otro lado, en la Figura 32 se muestra una comparación de la respuesta a un escalón unitario en perturbación para los cuatro controles. Esta perturbación, como se ha explicado en el apartado de Diseño de los controles PID, se corresponde con el par de fricción debido a la rampa del circuito. Se observa que la acción integral (PI y PID) consigue eliminar el error en régimen permanente, a diferencia de lo que ocurre en el P y PD. El aumento de la rapidez que conlleva la parte diferencial debería traducirse en un menor pico en la respuesta a perturbación, ya que se compensa más rápidamente. Sin embargo, en este caso casi no se puede observar mejoría en este aspecto al añadir la parte diferencial ya que, a causa del retardo, no se ha podido aumentar de forma significativa la rapidez de los controles.

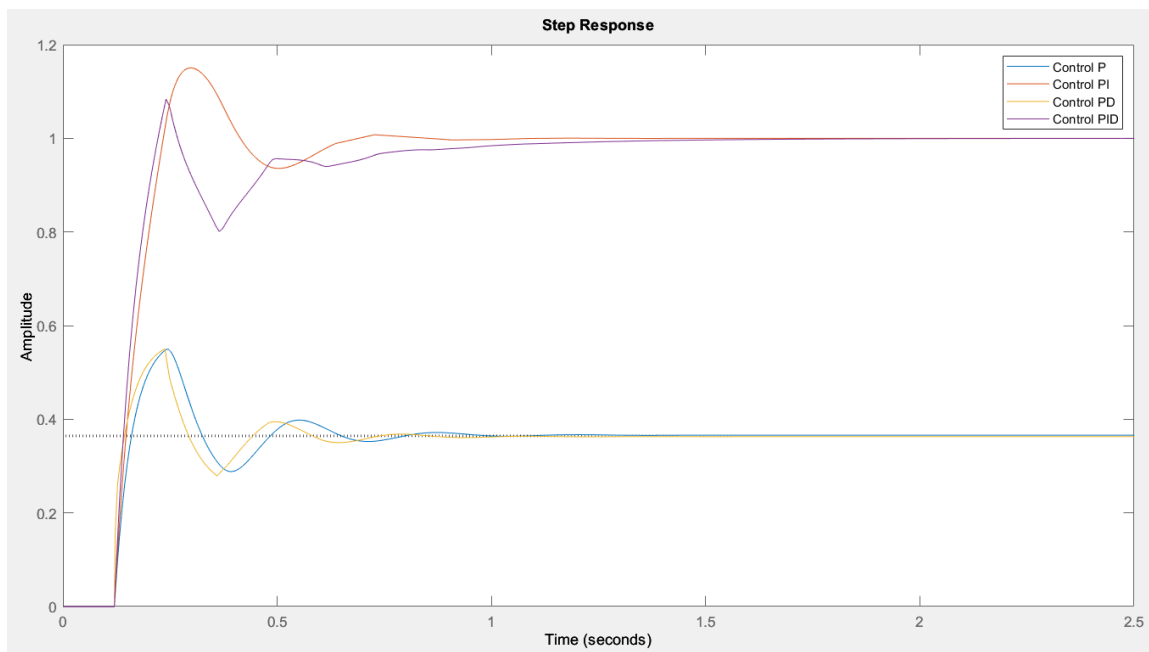


Figura 31. Comparativa de la respuesta a un escalón unitario en referencia para los controles diseñados.

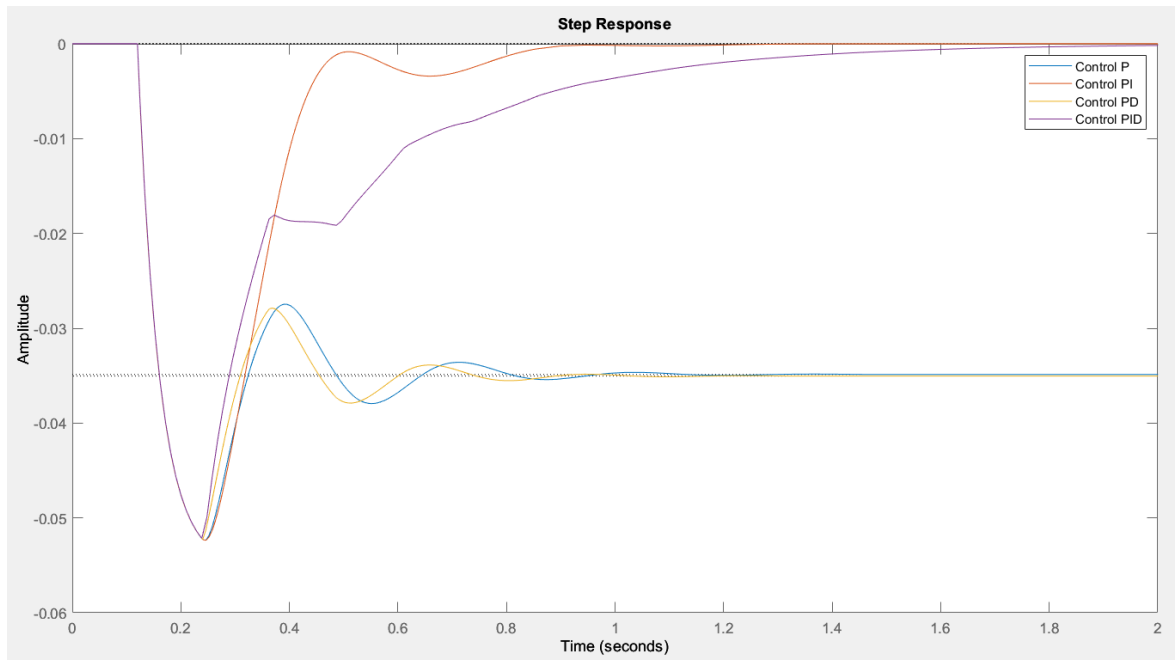


Figura 32. Comparativa a un escalón unitario en perturbación para los controles diseñados.

$A_m = 7\text{dB}$	w_u	K	T_i	T_d	N	f	φ_{PI}	φ_{PD}
P	20,46	10,50	∞	0	-	1	-	-
PI	16,37	8,00	0,09	0	-	1	$-34,22^\circ$	-
PD	22,51	10,36	∞	0,0135	9,0	0,1	-	$16,76^\circ$
PID	22,92	9,70	0,160	0,0290	6,5	0,1	$-20,00^\circ$	$20,09^\circ$

Tabla 8. Resumen del diseño de controles de velocidad.

$A_m = 7\text{dB}$	$M_p(\%)$	$t_p(\text{seg})$	$t_s(\text{seg})$	E_s	E_d
P	50,2	0,246	0,604	0,640	0,035
PI	15,0	0,298	0,556	0	0
PD	51,2	0,238	0,546	0,640	0,034
PID	8,3	0,241	0,674	0	0

Tabla 9. Comparativa de la respuesta temporal para los controles diseñados.

4.3 IMPLEMENTACIÓN DE LOS CONTROLES DE VELOCIDAD DE AVANCE

La implementación del control de velocidad de avance consta de varios archivos. En primer lugar, se ha desarrollado el diagrama del control mostrado en la Figura 33 en Simulink. Este archivo, equivalente al diagrama de bloques simplificado mostrado en la Figura 21, se despliega en la Raspberry Pi del robot utilizando la generación de código C++ (Simulink Coder) y la ROS *Toolbox* de MATLAB. Una vez volcado en el robot si se desea modificar el diseño del control o cambiar sus parámetros es necesario compilar y volver a volcar el archivo de nuevo. Este proceso puede llevar varios minutos. Por ello, se ha implementado el control de forma que todos los parámetros que sea necesario cambiar con frecuencia y las consignas necesarias se envíen desde otro archivo, mediante *topics*. De esta forma, se tienen:

- *Topic* /control_reference: recibe un mensaje de tipo std_msgs/Float32 con la consigna de velocidad.
- *Topic* /control_on: recibe un mensaje de tipo std_msgs/Bool que se encarga de activar o desactivar el control al comienzo y al final del ensayo. Esta variable es necesaria ya que puede darse el caso que, tras finalizar el ensayo, el nodo del control no se apague correctamente y el robot continúe funcionando. Esta variable recibe un cero cuando termina el tiempo del ensayo, que es multiplicado por el mando, por lo que, aunque el control siga generando un valor de mando, el robot recibirá un cero en el mando y se detendrá.
- *Topic* /cmd_vel: está dentro del subsistema ‘U(s)’ y publica un mensaje de tipo geometry_msgs/Twist con el mando resultante del control. Este mando está limitado entre 0 y 100 por el bloque ‘Limita 100%’, para evitar que se envíen mandos superiores a lo razonable. En caso de que la entrada y la salida del bloque no coincidan quiere decir que el mando está saturado, y se activa la variable ‘AW_VEL’ para evitar que la parte integral del control acumule error (*anti-windup*).

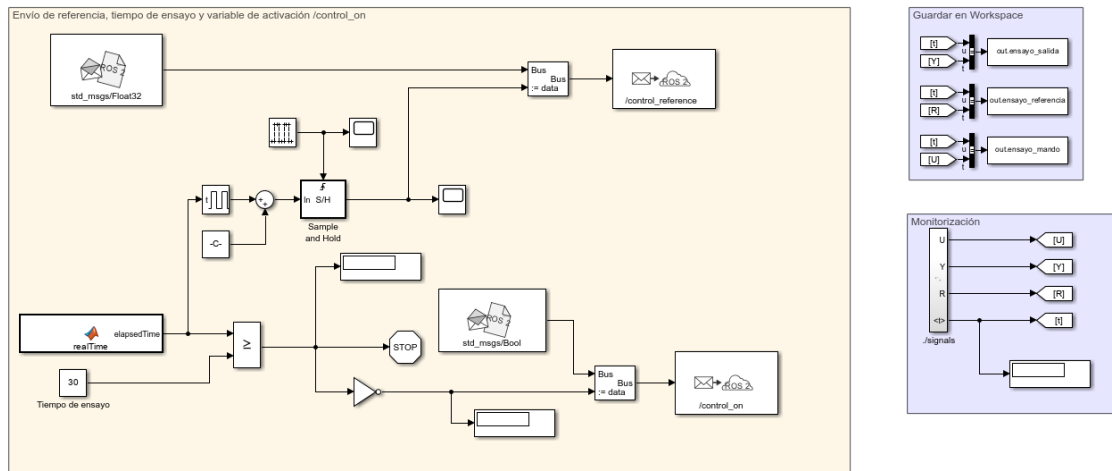


Figura 34. Archivo de monitorización y envío de referencia.

En la Figura 35 se muestra un gráfico de nodos del control de velocidad. En el mismo se observa que hay seis nodos (elipses) y seis *topics* (rectángulos). Los nodos `/VelocityNode` y `/Envio_referencia_91338` corresponden a los archivos de Simulink descritos anteriormente, el resto de los nodos son creados por el robot. Con respecto a los *topics*, el nodo `/Envio_referencia_91338` publica `/control_reference` y `/control_on` mientras que el nodo `/VelocityNode` tiene los suscriptores asociados para recibir dichos mensajes. Este mismo nodo publica la consigna de mando a través de `/cmd_vel`, como se ha explicado anteriormente. Además, también publica `/signals` con las medidas del ensayo.

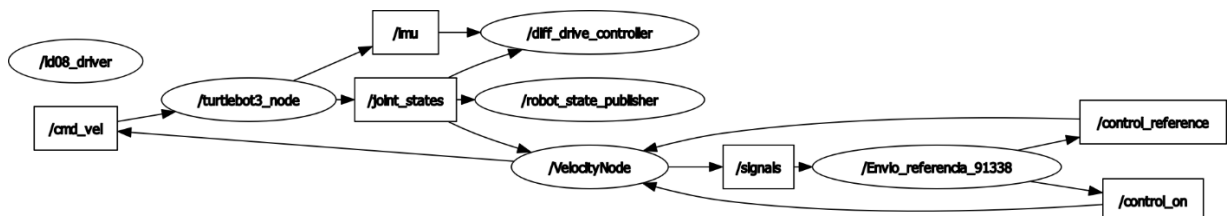


Figura 35. Gráfico de nodos del control de velocidad.

4.4 COMPARACIÓN ENTRE SIMULACIÓN Y ENSAYO

Una vez implementado y diseñado el control, se van a comparar en este apartado los resultados obtenidos de la simulación con los obtenidos con ensayos reales.

En primer lugar, en la Figura 36 se muestra una comparación entre el ensayo y la simulación para un control proporcional con ganancia $K = 10,5$. Se observa que el sobrepaso es igual en ambas, de aproximadamente el 50%. El tiempo en el que se produce el pico también coincide en ambas gráficas (0,242 segundos). Sin embargo, a partir de los 0,4 segundos el ensayo parece ser más lento que la simulación. Por otro lado, en ambas gráficas se observa que no se alcanza la referencia en régimen permanente, existiendo una diferencia importante, por lo que será necesario aplicar acción integral para corregirlo.

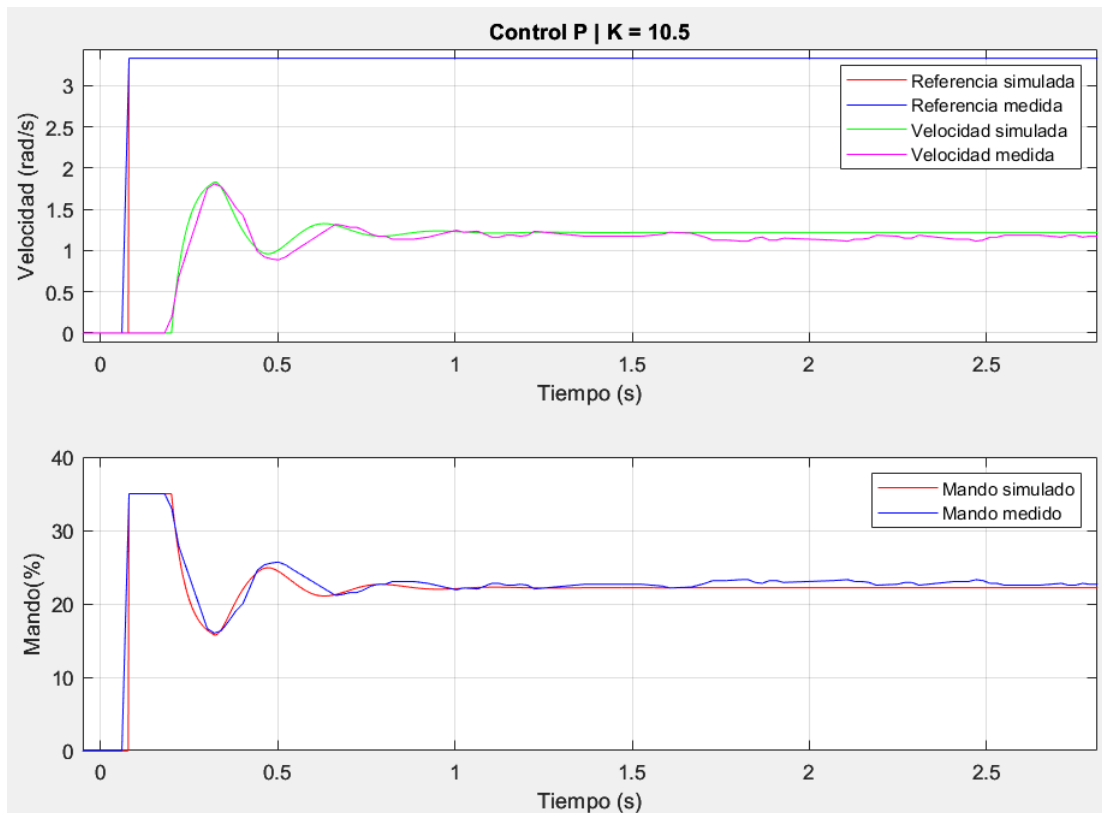


Figura 36. Comparación entre ensayo y simulación para el control P.

En la Figura 37 se muestra la comparación entre un ensayo y una simulación para el control PI, con los parámetros indicados en la imagen. La simulación tiene un sobrepaso del 15% sin embargo, el ensayo tiene un sobrepaso mayor (23% aproximadamente). Con respecto al tiempo de pico, este se produce antes en la simulación (0,328 segundos) que en el ensayo (0,361 segundos). En ambos casos el mando alcanza un valor máximo del 74% aproximadamente y se estabiliza en régimen permanente en un 60% para la simulación y 58% para el ensayo. En ambos casos se alcanza el valor de la referencia de velocidad en régimen permanente.

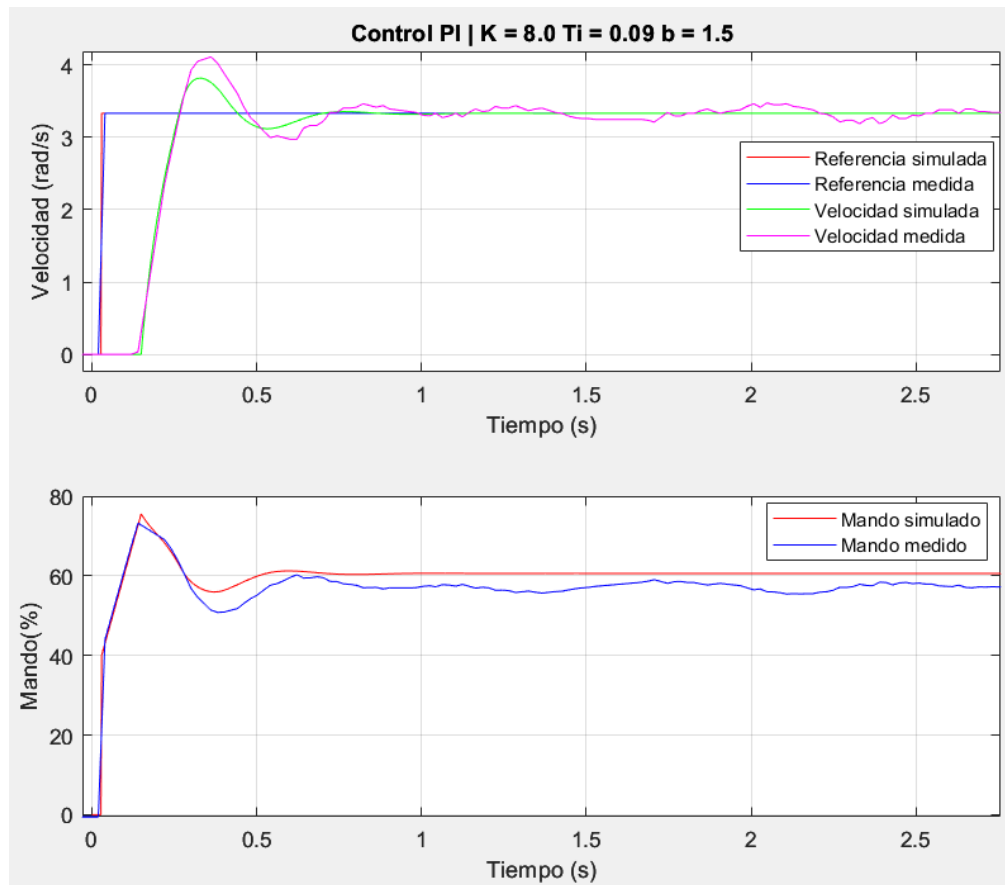


Figura 37. Comparación entre ensayo y simulación para el control PI.

La comparación entre ensayo y simulación para un control PD se muestra en la Figura 38. En ella se puede observar que, al igual que en el control anterior, el sobrepaso de la respuesta real es superior al de la simulada (65,2% frente a 51,2%). Además, los tiempos de pico no coinciden, siendo el ensayo más lento que la simulación. Sin embargo, este tiempo sí es menor a los obtenidos con el control P y el PI (0,301 segundos frente a 0,321 segundos del P y 0,328 segundos del PI). Este incremento en la rapidez no es muy significativo a causa del retardo de la planta, que dificulta el diseño de controles si se aumenta mucho la velocidad. Al igual que ocurría con el control P, al no aplicar la acción integral el error en régimen permanente no es cero y la salida no alcanza el valor de la referencia.

Otra característica relevante de esta gráfica es el pico de mando que se produce al inicio. Este efecto de saturación del mando, producido por la derivada del error, se conoce como patada derivativa y se consigue evitar aplicando la acción diferencial a la salida. Este efecto también existe en el ensayo, pero no se observa en la gráfica, probablemente porque se habrá producido en un instante que no se ha muestreado.

En la Figura 39 se muestra el mismo control PD, pero ahora aplicando la acción diferencial a la salida. Se comprueba que el efecto de la patada derivativa ya no está ni en la simulación ni en el ensayo.

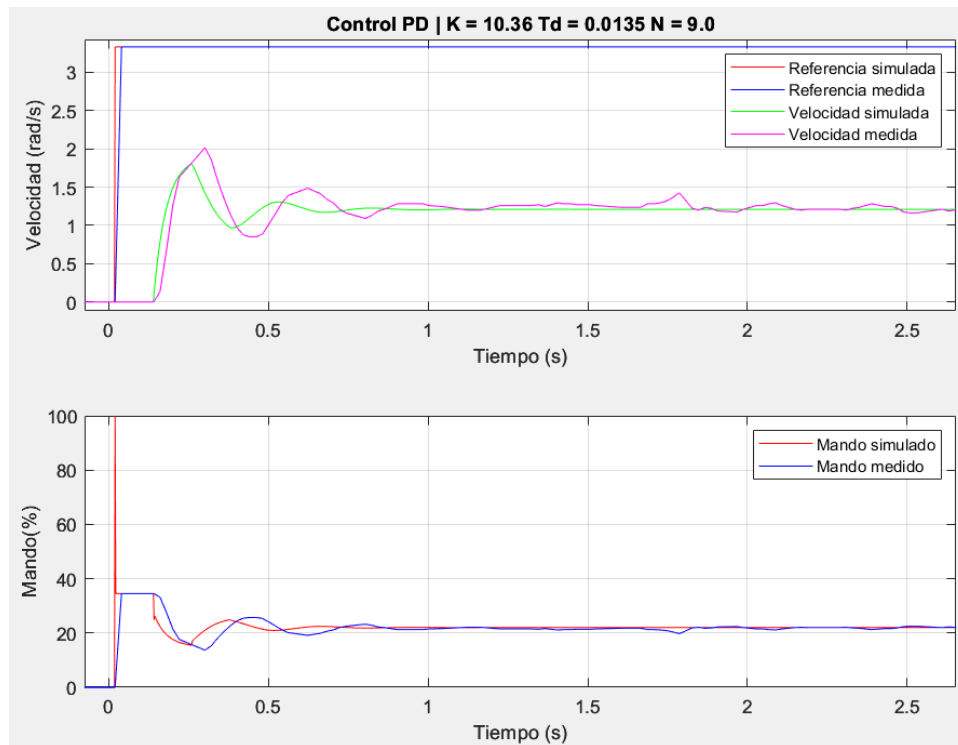


Figura 38. Comparación entre ensayo y simulación para el control PD con acción diferencial al error.

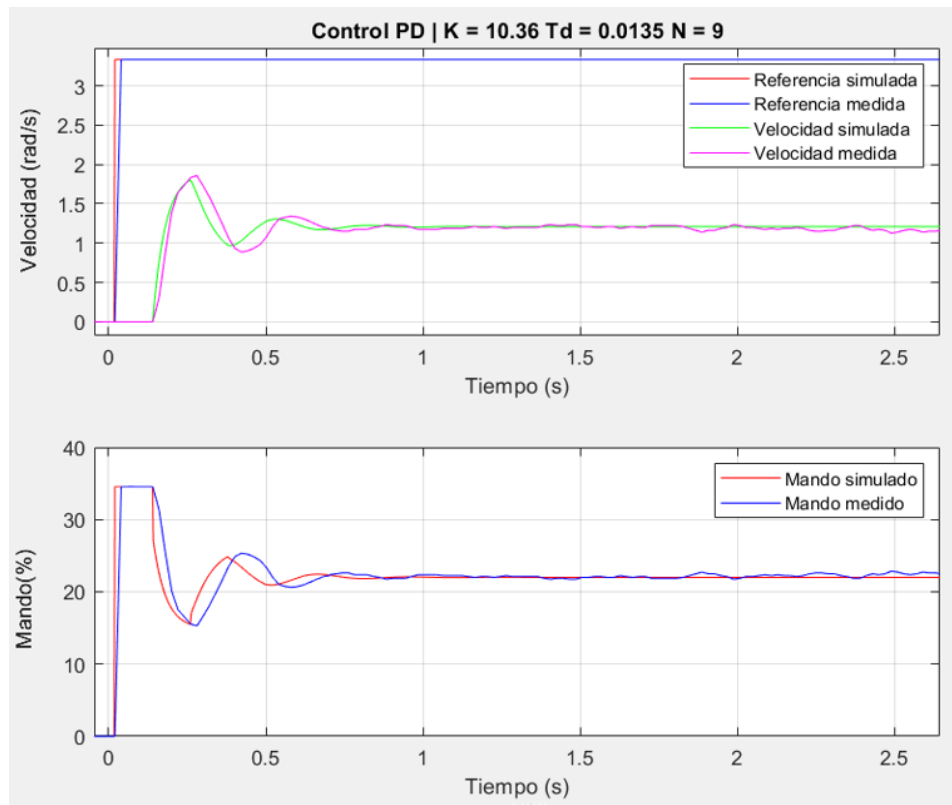


Figura 39. Comparación entre ensayo y simulación para el control PD con acción diferencial a la salida.

Por último, en la Figura 40 se muestra la comparación entre ensayo y simulación para un control PID con acción diferencial a la salida. En este caso el ensayo tiene un sobrepaso mayor que la simulación (16,82% frente a 8%) además de ser más lento que la misma (tiempo de pico del ensayo de 0,306 segundos frente a 0,266 segundos). Al igual que para el control PD, el retardo de la planta dificulta el diseño de controles con incrementos de velocidad significativos, por lo que no se aprecia una ventaja importante con el P o el PI en este aspecto. En ambos casos se observa que, gracias a la parte integral, se elimina el error de seguimiento a referencia en régimen permanente.

En la Figura 41 se muestra la comparación del mismo ensayo PID, pero aplicando la acción diferencial al error. Al igual que con el control PD se observa el efecto de la patada derivativa en los primeros instantes del mando.

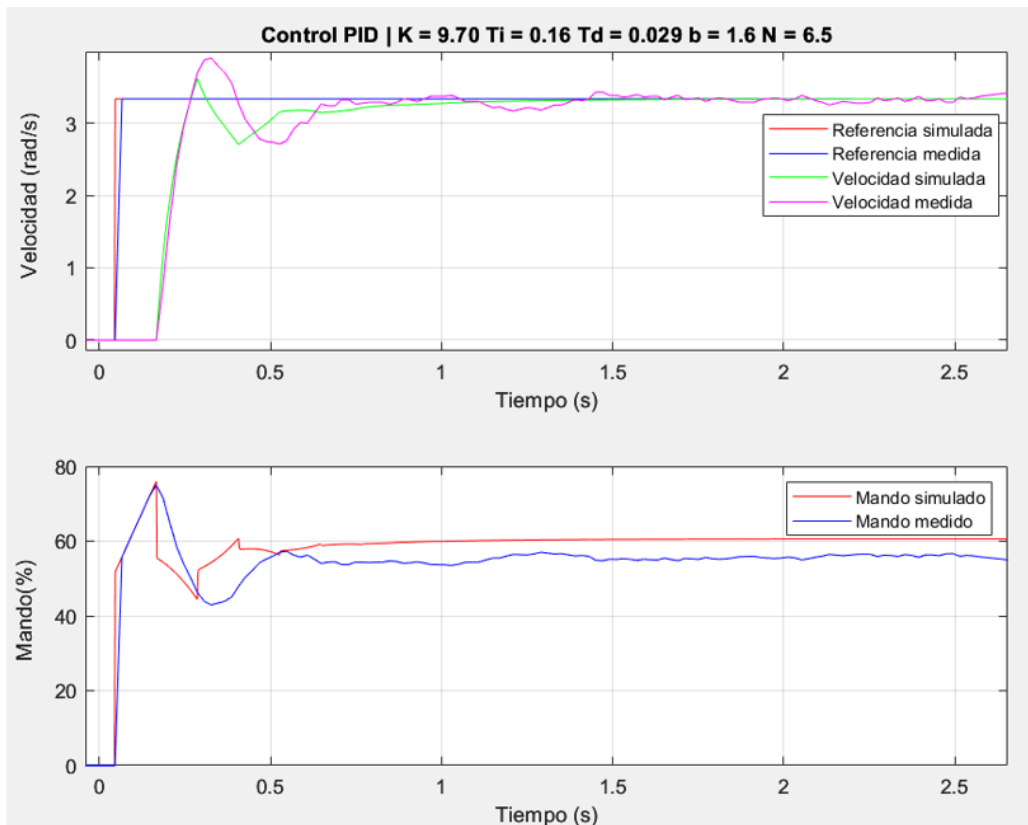


Figura 40. Comparación entre ensayo y simulación para el control PID con acción diferencial a la salida.

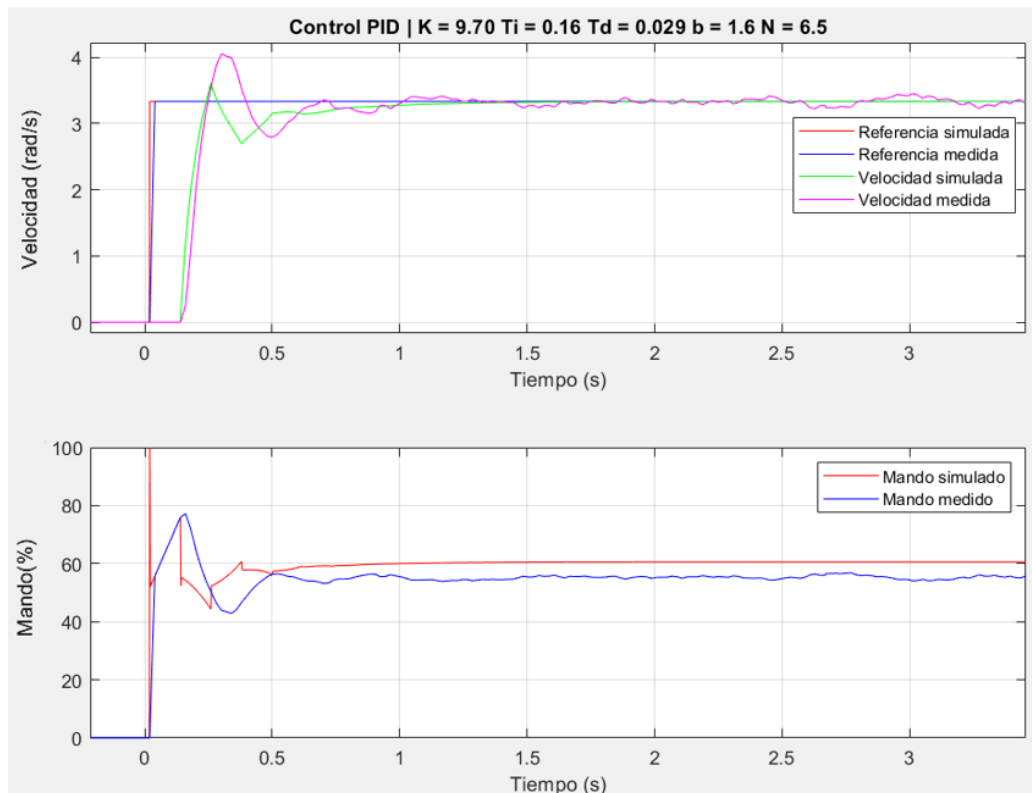


Figura 41. Comparación entre ensayo y simulación para el control PID con acción diferencial al error.

4.5 CONCLUSIONES DEL CONTROL DE VELOCIDAD DE AVANCE

Tras analizar los resultados obtenidos con los distintos controles se puede concluir que es necesario aplicar acción integral para poder eliminar el error en seguimiento a referencia y perturbación en régimen permanente. Finalmente se decide escoger un control PI frente a un PID ya que, por las características de la planta, la rapidez obtenida con los controles PID diseñados frente a un PI no es demasiado significativa.

Capítulo 5. CONTROL DE SEGUIMIENTO DE DISTANCIA A LA PARED

En este capítulo se va a explicar el proceso seguido para el diseño del control de distancia a la pared. Para ello, en primer lugar, se ha realizado una identificación del modelo de la planta. Posteriormente se han diseñado los controles PID de forma teórica para la planta identificada. A continuación, se ha descrito cómo se han implementado los controles en el robot. Finalmente se han comparado los ensayos realizados con la simulación. Además, se ha estudiado la respuesta de los controles frente a la perturbación, provocada por el cambio en el ángulo al llegar la curva del circuito. Por último, se ha analizado cómo se comporta el robot si se cambia el punto de operación utilizado para modelar la planta.

5.1 IDENTIFICACIÓN DE LA PLANTA [30]

Para medir la distancia a la pared del circuito, el robot consta de dos sensores VL6180X ToF (*Time of Flight*). Estos sensores están situados como se indica en la Figura 42, de forma que proporcionan las medidas d_1 y d_2 . Mediante estos valores se obtiene la distancia mínima del punto A (punto medio entre los dos sensores, separados entre sí una distancia d_s) a la pared (d_n) y el ángulo de orientación con respecto a la misma (θ) como:

$$\theta = \operatorname{atan}\left(\frac{d_2 - d_1}{d_s}\right)$$
$$d_n = \frac{(d_1 + d_2)}{2} \cdot \cos(\theta)$$

Ecuación 5. Medidas de distancia y ángulo de orientación con respecto a la pared.

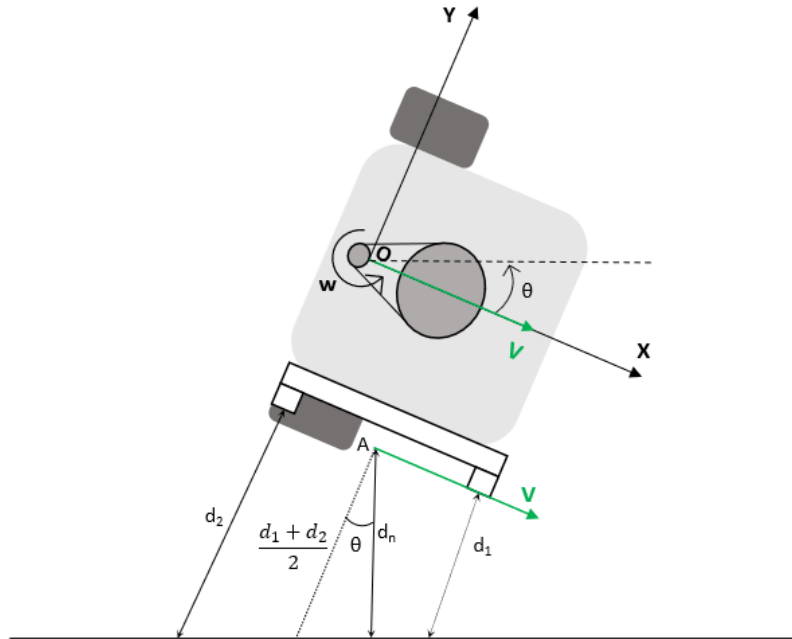


Figura 42. Medidas del sensor ToF.

Si se define V como la velocidad de avance del punto medio de las ruedas, x_A e y_A como las coordenadas del punto A en el sistema de referencia solidario al vehículo y w como la velocidad angular de rotación, se puede definir la velocidad del punto A, como si este perteneciese al vehículo, como:

$$\begin{bmatrix} v_{Ax} \\ v_{Ay} \end{bmatrix} = \begin{bmatrix} v \\ 0 \end{bmatrix} + w \cdot \begin{bmatrix} y_A \\ x_A \end{bmatrix} = \begin{bmatrix} v + w \cdot y_A \\ w \cdot x_A \end{bmatrix}$$

Ecuación 6. Velocidad del punto A.

A partir de las ecuaciones anteriores, se puede obtener la componente normal a la pared del punto A como:

$$v_{An} = v_{Ax} \cdot \text{sen}(\theta) + v_{Ay} \cdot \text{cos}(\theta)$$

Ecuación 7. Componente normal de la velocidad del punto A.

Por otro lado, es posible modelar la relación entre la tensión diferencial de mando u_d (%) y la velocidad angular del robot w (rad/s) a partir de una ecuación diferencial de primer orden, mostrada en la Ecuación 8. T_m es la constante de tiempo de la función de transferencia entre ambas y K_m es la ganancia estática.

$$\frac{dw}{dt} = -\frac{1}{T_m}w + \frac{K_m}{T_m}u_d$$

Ecuación 8. Relación entre velocidad angular y tensión diferencial.

Por último, se puede obtener la derivada del ángulo θ (relativo entre el robot y la pared) como se muestra en la Ecuación 9, tratando a la pared como un cuerpo que gira con una velocidad w_{pared} .

$$\frac{d\theta}{dt} = w - w_{pared}$$

Ecuación 9. Derivada del ángulo del robot con respecto a la pared.

Con estas ecuaciones se puede obtener un modelo, siendo la única no linealidad la componente normal de la velocidad, ya que incluye funciones trigonométricas.

Linealizando para un punto de operación $v = v_o$ se obtiene la función de transferencia entre la tensión diferencial aplicada a los motores u_d y la distancia a la pared d_n . Esta función de transferencia se puede definir como el producto de dos. Por un lado, $P_1(s)(\theta(s)/U_d(s))$, por otro lado, $P_2(s)(D_n(s)/\theta(s))$.

$$P(s) = \frac{D_n(s)}{U_d(s)} = P_1(s) \cdot P_2(s) = \frac{K_m}{s \cdot (T_m s + 1)} \cdot \frac{v_o \cdot \left(1 + \frac{x_A}{v_o} s\right)}{s}$$

Ecuación 10. Función de transferencia de la planta del control de seguimiento de pared.

Un detalle que destacar de dicha función de transferencia es el hecho de que, mientras que el resto de las variables del punto de operación no afectan, la velocidad de avance sí influye en la dinámica de la planta (en concreto en el cero y en la ganancia estática).

Una vez obtenido el modelo de la planta, se procede a la obtención de los valores numéricos de la misma. Para la planta $P_1(s)$ se va a utilizar un ajuste por mínimos cuadrados, como el usado para identificar la planta del control de velocidad de avance. Para ello se ha utilizado un control proporcional en ambos lazos. En el interno se ha escogido una ganancia $K = 90,0$ y en el externo una ganancia $K = 20,0$. El ensayo se puede observar en la Figura 43.

En la Figura 44 se muestra el resultado de la identificación de esta planta. Se puede observar que, debido al ruido presente en las medidas, entre otros factores, la salida simulada no se ajusta de forma perfecta a la real. Los parámetros obtenidos de esta identificación son los mostrados en la Ecuación 11.

$$P_1(s) = \frac{\theta}{u_d} = \frac{0,04}{(0,2 \cdot s + 1) \cdot s}$$

Ecuación 11. Función de transferencia de la planta $P_1(s)$.

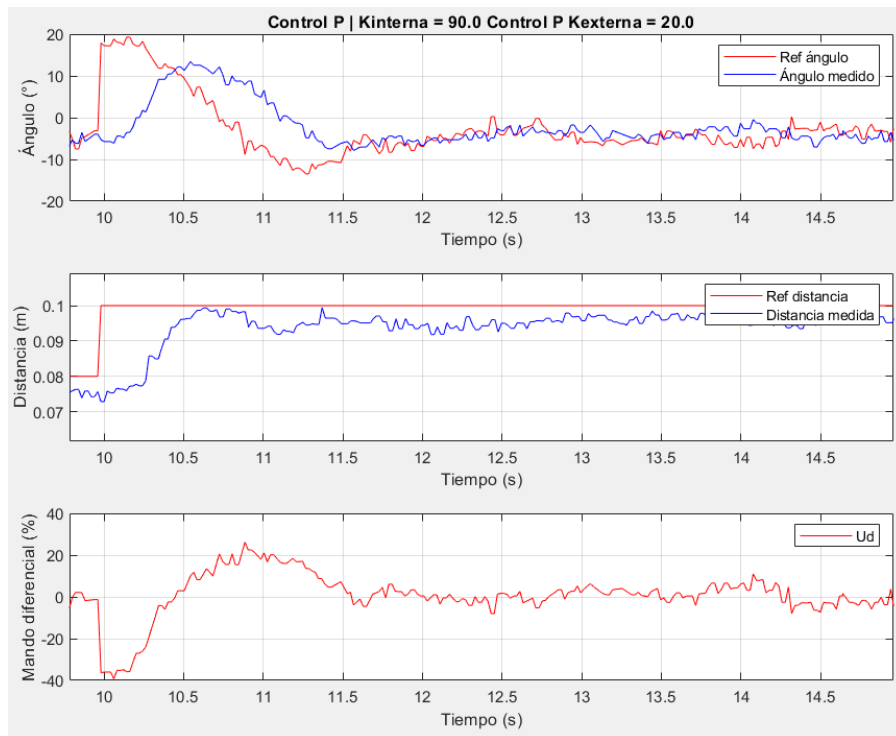


Figura 43. Ensayo utilizado para la identificación de la planta $P_1(s)$.

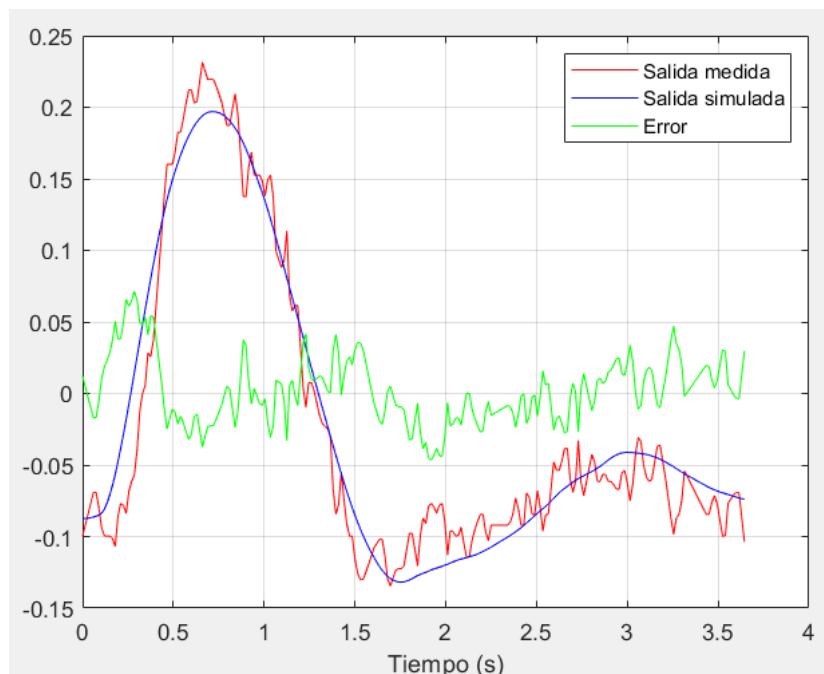


Figura 44. Resultado de identificación de la planta $P_1(s)$.

Para la identificación de la segunda planta $P_2(s)$ no se va a utilizar el método de mínimos cuadrados, ya que son todo valores que se pueden obtener del punto de operación y de la geometría del robot. De esta forma la planta resultante es:

$$P_2(s) = \frac{d_n}{\theta} = \frac{0,15 \cdot \left(1 + \frac{0,06}{0,15}s\right)}{s}$$

Ecuación 12. Función de transferencia de la planta $P_2(s)$.

5.2 DISEÑO DE LOS CONTROLES PID

Una vez obtenidos los modelos de las plantas descritas en el apartado anterior, se van a diseñar ahora los controles de forma teórica.

En vez de implementar un control con el modelo de realimentación habitual, que sería complejo de diseñar ya que habría dos integraciones en la planta, se va a diseñar un control en cascada. Este control cuenta con dos lazos, como se muestra en la Figura 45. El lazo externo tiene como referencia la distancia a la pared deseada y como salida la referencia de ángulo para el lazo interno. Este lazo intenta conseguir esa referencia dada aplicando la tensión diferencial (u_d) necesaria. En la figura se observa también que hay dos perturbaciones. Por un lado, está d_1 que se corresponde con la asimetría de los motores, que está relacionada con la diferencia entre la tensión aplicada a ambos motores. Por otro lado, está d_2 , que representa el cambio en el ángulo medido que se produce cuando el robot llega a una curva. Los bloques $Cr(s)$ y $C(s)$ son las funciones de transferencia de los controles que se van a diseñar a continuación (control de dos grados de libertad).

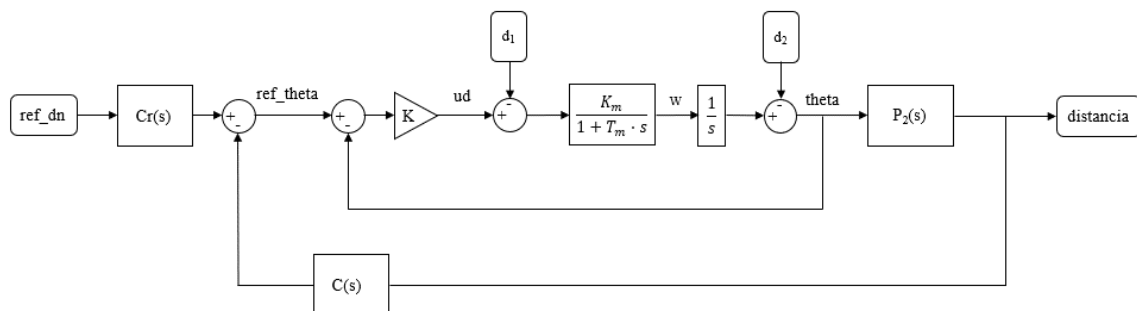


Figura 45. Diagrama del sistema a controlar.

DISEÑO DEL CONTROL DEL LAZO INTERNO

En primer lugar, se va a diseñar el control para el lazo interno. En la Figura 46 se muestra el diagrama de Black de la planta $P_1(s)$. Se observa que, a diferencia de la planta del control de velocidad de avance, este gráfico no es especialmente “horizontal”, por lo que el margen limitante y por el que se va a diseñar es el de fase.

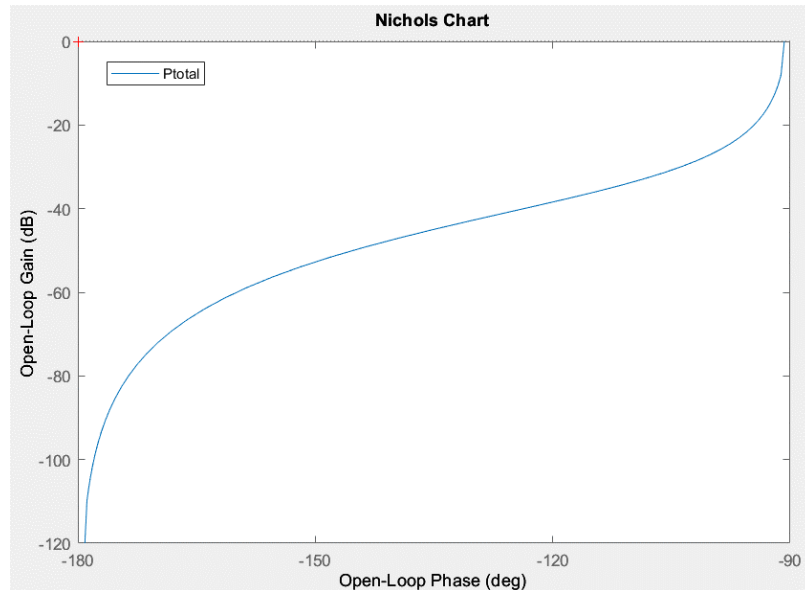


Figura 46. Diagrama de Black de la planta del lazo interno.

5.2.1 CONTROL PROPORCIONAL, P

En la Ecuación 13 se muestra la función de transferencia del control proporcional, que cuenta con la ganancia como único grado de libertad. La elección de este parámetro conlleva un compromiso entre el amortiguamiento (se ve afectado si la ganancia tiene valores demasiado elevados) y la rapidez. En este caso se quiere diseñar un control con un sobrepaso no muy elevado, ya que oscilaciones muy grandes afectan a la respuesta del control externo, que puede llegar a convertirse en un sistema oscilante.

$$C(s) = K_p$$

Ecuación 13. Control proporcional (formato serie).

Para el diseño analítico por margen de fase es necesario seguir estos pasos:

1. Establecer el margen de fase (Φ_m). Valores típicos son entre 45° y 60° .
2. Calcular la pulsación de cruce (w_o).

$$\angle G(jw_o) = \angle P(jw_o) = -180^\circ + \Phi_m$$

3. Calcular la ganancia K_p .
 - a. En reales

$$|G(jw_o)| = K_p \cdot |P(jw_o)| = 1 \rightarrow K_p = \frac{1}{|P(jw_o)|}$$

- b. En dB

$$|G(jw_o)|_{dB} = K_p \text{ dB} + |P(jw_o)|_{dB} = 0 \rightarrow K_p \text{ dB} = -|P(jw_o)|_{dB}$$

Para el diseño se ha escogido finalmente un margen de fase de 60° obteniéndose una pulsación de cruce de 2,88 rad/s (ver Figura 47) y una ganancia $K = 83,33$. En la Figura 48 se muestra la respuesta teórica de este control frente a un escalón unitario en referencia y perturbación (d_I). Se comprueba que la respuesta tiene un sobrepaso razonable (8,7%), alcanzando el pico a los 0,97 segundos y con un tiempo de establecimiento al 5% de 1,27 segundos. Además, se observa que el error en seguimiento a referencia es cero gracias a la integración de la planta, por lo que finalmente se decide escoger este control para el lazo interno, ya que el error en rechazo a perturbación (d_I) es pequeño y puede encargarse de ello el diseño del lazo externo.

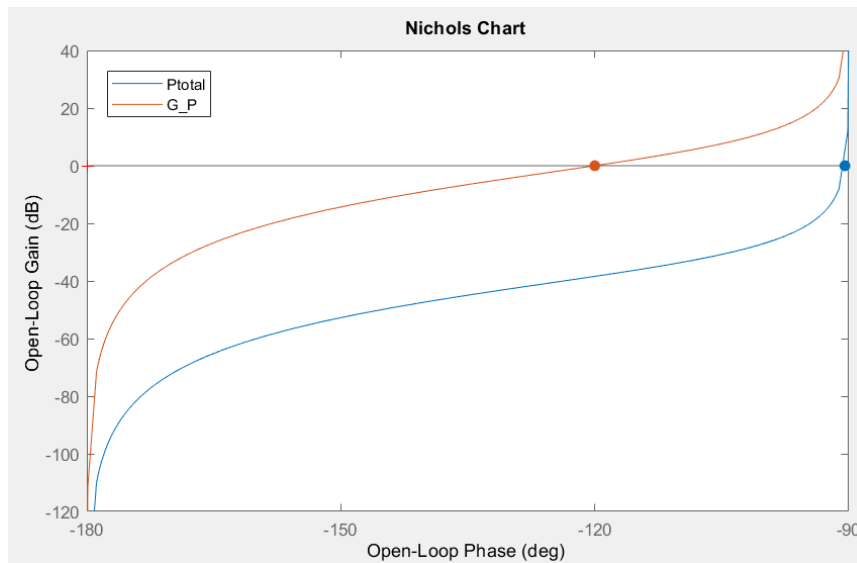


Figura 47. Gráfico de Black para la planta $P(s)$ y el lazo abierto $G(s)$ para el control P del lazo interno.

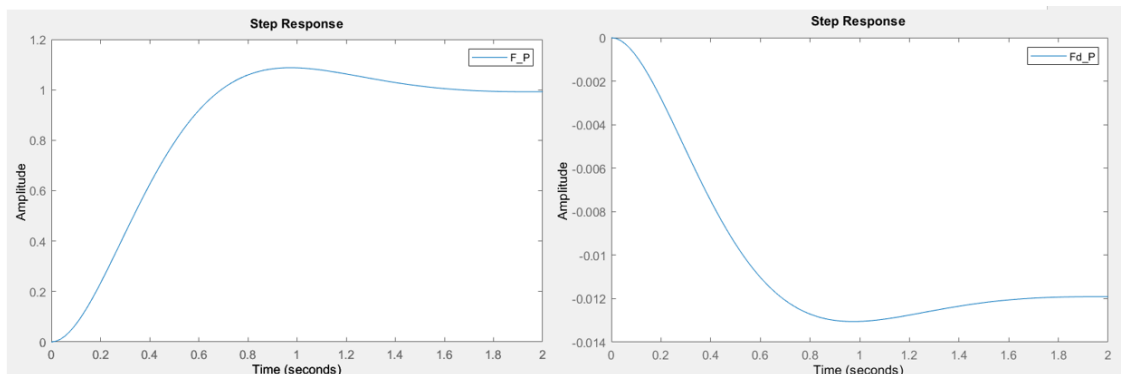


Figura 48. Respuesta a un escalón unitario en referencia y perturbación para el control P del lazo interno.

DISEÑO DEL CONTROL DEL LAZO EXTERNO

La función de transferencia del lazo cerrado anterior es parte de la planta del lazo externo, cuya gráfica de Black se muestra en la Figura 49. De nuevo se puede comprobar que el margen limitante y por el que se va a diseñar, es el de fase.

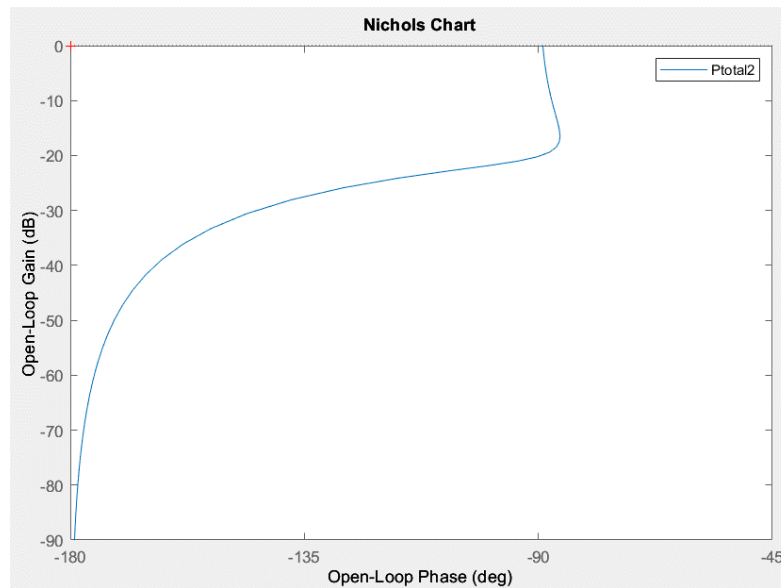


Figura 49. Diagrama de Black de la planta del lazo externo.

5.2.2 CONTROL PROPORCIONAL, P

Siguiendo los pasos de diseño descritos en el apartado anterior para un control proporcional con un margen de fase de 65° (ver Figura 50) se obtiene una ganancia de 15,55 y una pulsación de cruce ω_o de 3,70 rad/s. En la Figura 51 se muestra la respuesta del control para un escalón unitario en referencia y perturbación (d_2). Se observa que, debido a la integración de la planta, no hay error de seguimiento a referencia. En este caso la respuesta no tiene sobrepaso, siendo 0,99 su valor máximo y alcanzándolo a los 0,74 segundos. En cuanto al tiempo de establecimiento al 5% es de 1,67 segundos.

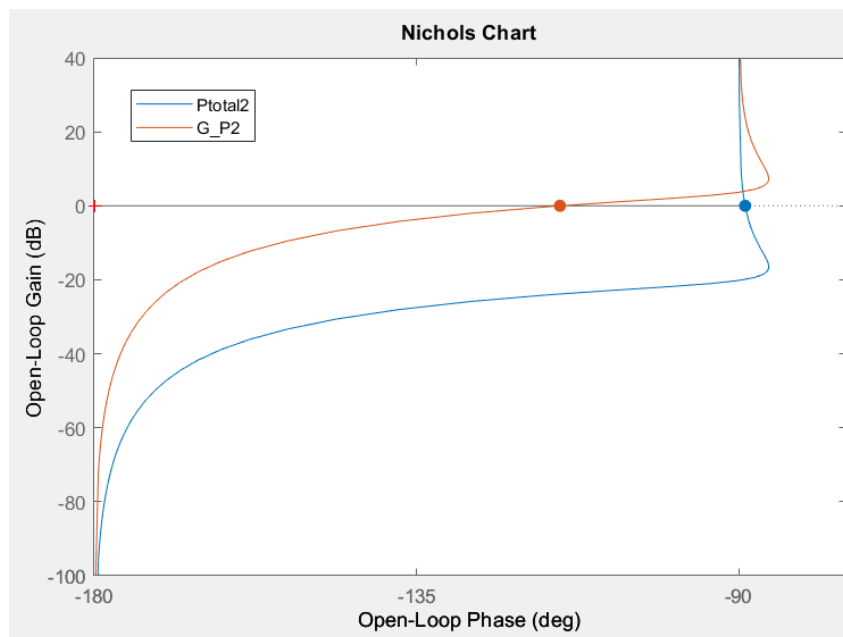


Figura 50. Gráfico de Black para la planta $P(s)$ y el lazo abierto $G(s)$ para el control P del lazo externo.

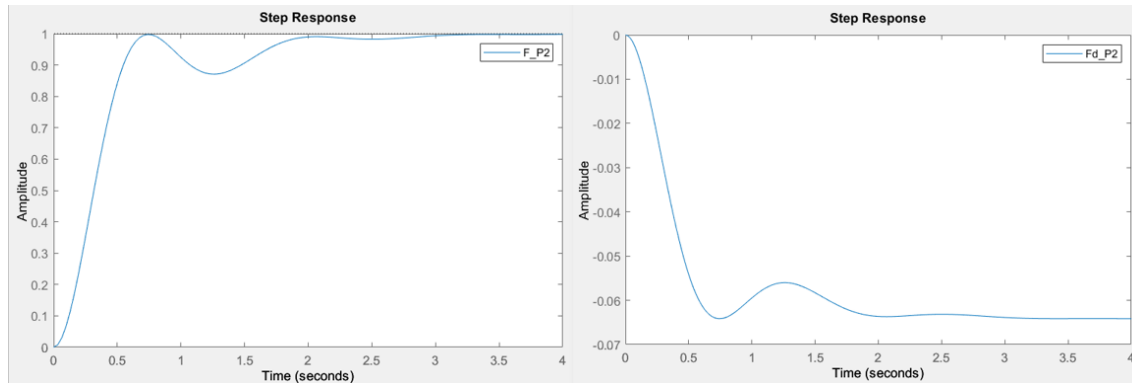


Figura 51. Respuesta a un escalón unitario en referencia y perturbación para el control P del lazo externo.

5.2.3 CONTROL PROPORCIONAL INTEGRAL, PI

El objetivo principal de este control es mejorar la precisión del sistema, anulando el error en régimen permanente tanto en seguimiento de referencia como rechazo a una perturbación constante. La función de transferencia para el control PI se muestra en la Ecuación 14, donde se observa que existe un desplazamiento del gráfico de Black hacia la izquierda, debido al retraso de fase que introduce el control.

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

Ecuación 14. Función de transferencia para el control PI (formato serie).

Para diseñar este tipo de controles lo habitual es fijar la velocidad, menor que la del control P para acomodar el retraso de fase, y el amortiguamiento, como se indica a continuación:

1. Establecer el margen de fase (Φ_m), entre 45° y 60° típicamente.
2. Definir el valor de la pulsación de cruce ($w_{oPI} < w_{oP}$). Aunque también se podría fijar el retraso de fase aportado por la parte integral del control.
3. Calcular la fase del control.

$$\angle G(jw_o) = \angle C(jw_o) + \angle P(jw_o) = -180^\circ + \phi_m$$

$$\varphi_c = -180^\circ + \phi_m - \angle P(jw_o)$$

4. Calcular la ganancia del control (A_C).
 - a. En reales.

$$|G(jw_o)| = |C(jw_o)| \cdot |P(jw_o)| = 1 \rightarrow A_c = \frac{1}{|P(jw_o)|}$$

- b. En dB.

$$|G(jw_o)|_{dB} = |C(jw_o)|_{dB} + |P(jw_o)|_{dB} = 0 \rightarrow A_{c\ dB} = -|P(jw_o)|_{dB}$$

5. Calcular los parámetros del control.

$$|C(j\omega_o)| \cdot e^{\angle C(j\omega_o)} = A_C \cdot e^{j\varphi_C} = K_P \cdot \frac{1 + j\omega_o I}{j\omega_o I}$$

a. Cálculo de I .

$$\varphi_C = -90^\circ + \text{atan}(I\omega_o) \rightarrow I = \frac{\tan(90^\circ + \varphi_C)}{\omega_o}$$

b. Cálculo de K_P .

$$A_C = K_P \cdot \frac{\sqrt{1 + (I\omega_o)^2}}{I\omega_o} \rightarrow K_P = \frac{A_C I \omega_o}{\sqrt{1 + (I\omega_o)^2}}$$

Seleccionando una $\omega_{oPI} = 0,8 \cdot \omega_{oP}$ y un margen de fase de 65° se obtiene un retraso de fase de $-12,85^\circ$ (ver Figura 52). Los parámetros obtenidos del control son una ganancia K de 12,51 y un T_i de 1,47 segundos. En la Figura 53 se muestra la respuesta a un escalón unitario en referencia y perturbación. Se observa que el sobrepaso es aceptable (13,85%) por lo que se decide no usar ponderación a la referencia. El pico se alcanza a los 2,17 segundos. Por último, el tiempo de establecimiento al 5% obtenido es de 3,92 segundos, mayor que para el control P.

En este caso el error en seguimiento a referencia en régimen permanente ya era cero en simulación gracias a la integración de la planta. Sin embargo, añadir la acción integral sí lo elimina en rechazo a perturbación.

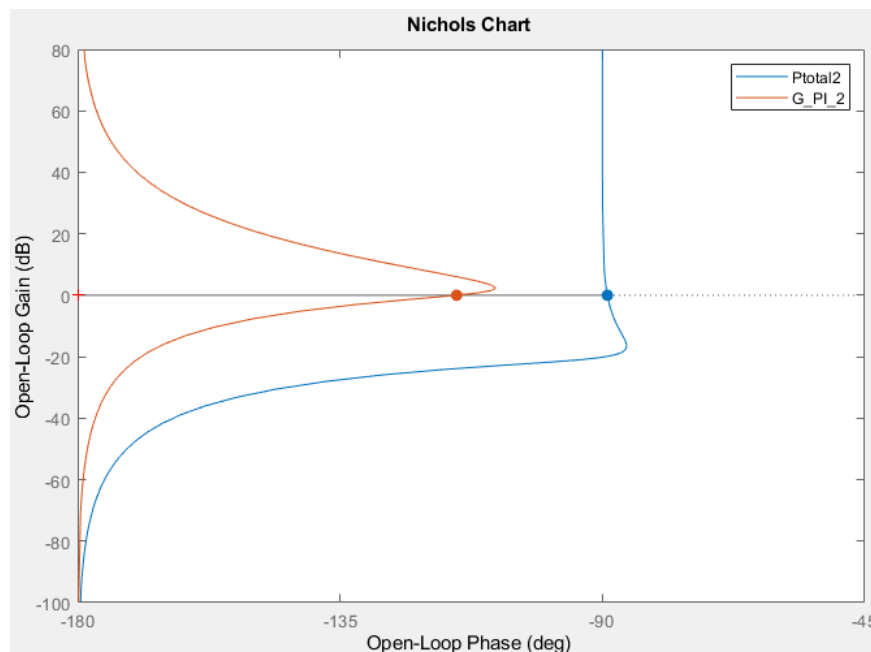


Figura 52. Gráfico de Black para la planta $P(s)$ y el lazo abierto $G(s)$ para el control PI del lazo externo

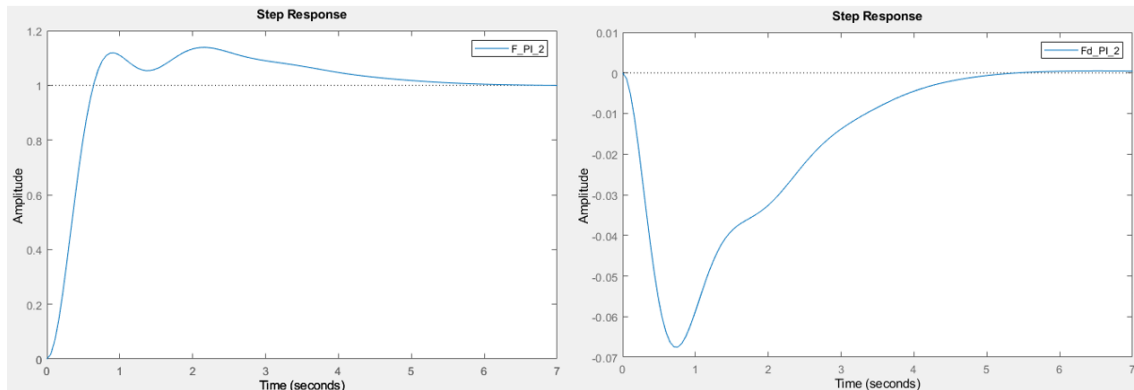


Figura 53. Respuesta a un escalón unitario en referencia y perturbación para el control PI del lazo externo.

5.2.4 CONTROL PROPORCIONAL DIFERENCIAL, PD

El objetivo que se pretende conseguir con este control es mejorar la respuesta transitoria (la rapidez, el amortiguamiento o ambas). Además, también es frecuente una mejora en la precisión con respecto al proporcional debido a que se suelen obtener valores mayores de K_P . En la Ecuación 15 se muestra la función de transferencia del control, siendo f el factor de filtrado. En este caso, se aprecia que el control presenta una fase positiva ($\pi/2 > \varphi_C > 0$), que provoca un desplazamiento hacia la derecha en el gráfico de Black.

$$C(s) = K_P \cdot \frac{1 + Ds}{1 + fDs}$$

Ecuación 15. Función de transferencia para el control PD (formato serie).

Los pasos a seguir para diseñar el control por respuesta en frecuencia son:

1. Establecer el margen de fase (Φ_m), entre 45° y 60° típicamente.
2. Establecer la pulsación de cruce ω_o y fijar el factor de filtrado como $0 < f < f_M$. Este factor de filtrado máximo se calcula como:

$$f_M = \frac{1 - \text{sen}(\varphi_C)}{1 + \text{sen}(\varphi_C)}$$

3. Calcular la fase del control.

$$\angle G(j\omega_o) = \angle C(j\omega_o) + \angle P(j\omega_o) = 180^\circ + \phi_m$$

$$\varphi_C = -180^\circ + \phi_m - \angle P(j\omega_o)$$

4. Calcular la ganancia del control (A_C).
 - a. En reales.

$$|G(j\omega_o)| = |C(j\omega_o)| \cdot |P(j\omega_o)| = 1 \rightarrow A_C = \frac{1}{|P(j\omega_o)|}$$

b. En dB.

$$|G(jw_o)|_{dB} = |C(jw_o)|_{dB} + |P(jw_o)|_{dB} = 0 \text{ dB}$$

$$A_C \text{ dB} = -|P(jw_o)|_{dB}$$

5. Calcular los parámetros del control.

a. Cálculo de D .

$$w_o D = \frac{\frac{1}{f} - 1}{2tg(\varphi_c)} - \sqrt{\left(\frac{\frac{1}{f} - 1}{2tg(\varphi_c)}\right)^2 - \frac{1}{f}}$$

b. Cálculo de K_P .

$$K_P = A_C \frac{\sqrt{1 + (fDw_o)^2}}{\sqrt{1 + (w_o D)^2}}$$

Escogiendo finalmente una $w_{oPD} = 1,5 \cdot w_{oP}$, con un factor de filtrado de 0,1 y un margen de fase de 65° se obtiene un adelanto de fase de $26,32^\circ$ (ver Figura 54). Los parámetros del control obtenidos son una ganancia K de 24,8, un T_d de 0,09 segundos y una N de 9. En la Figura 55 se muestra la respuesta a un escalón unitario en referencia y perturbación para este control. El pico de la primera oscilación se alcanza a los 0,533 segundos, un tiempo menor que para el control P gracias a la acción diferencial. La salida no tiene sobrepaso, siendo su valor máximo 0,99. El tiempo de establecimiento al 5 % es de 1,31 segundos, de nuevo más rápido que los controles anteriores. Al igual que pasaba con el control P no hay error de seguimiento a referencia en régimen permanente en el diseño teórico, gracias a la integración de la planta. Sin embargo, sí que está presente en la respuesta a perturbación.

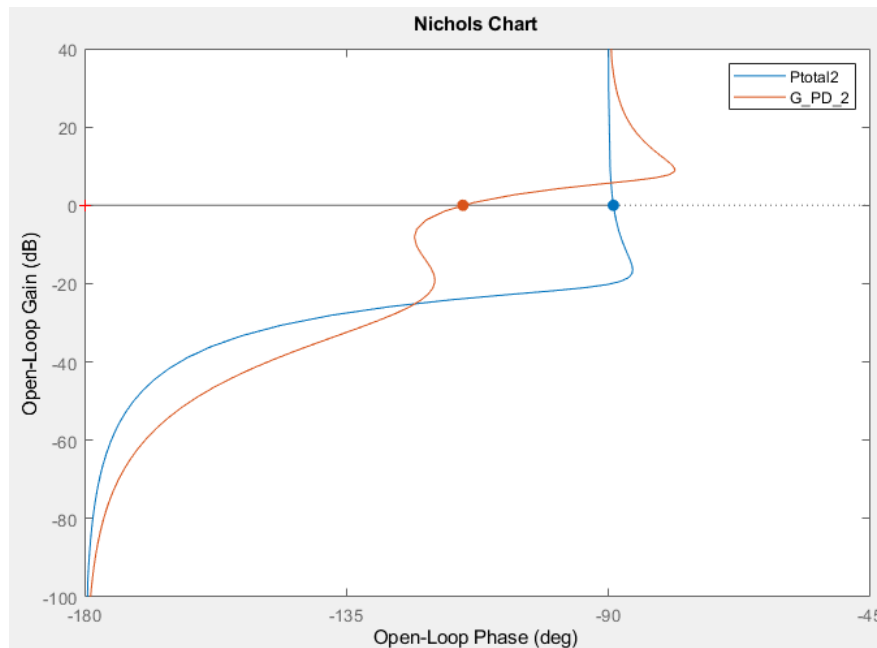


Figura 54. Gráfico de Black para la planta $P(s)$ y el lazo abierto $G(s)$ para el control PD del lazo externo.

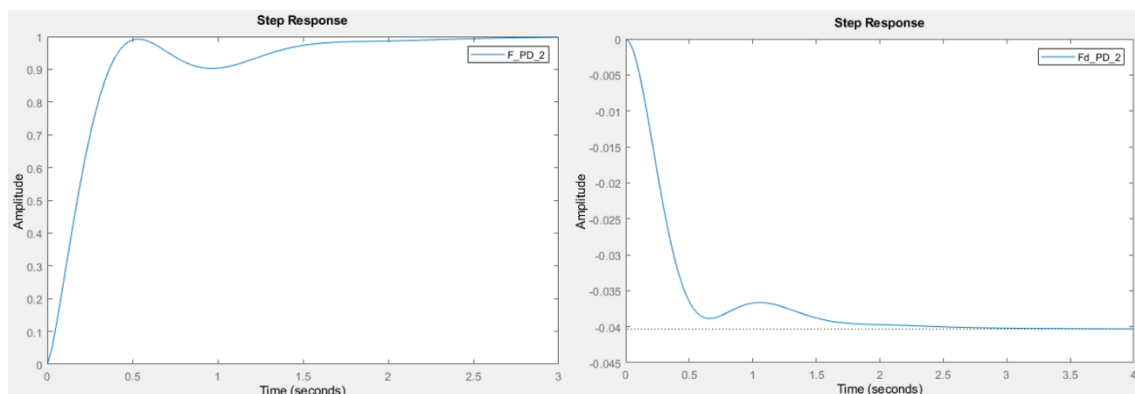


Figura 55. Respuesta a un escalón unitario en referencia y perturbación para el control PD del lazo externo.

5.2.5 CONTROL PROPORCIONAL INTEGRAL DIFERENCIAL, PID

El objetivo principal de este control es combinar las ventajas de la acción diferencial, en cuanto a mejora de la rapidez de la respuesta, y de la acción integral, en cuanto a la eliminación del error en régimen permanente. En la Ecuación 16 se muestra la función de transferencia de este control. Al combinar ambas acciones se produce un adelanto de fase neto, ya que el adelanto de la parte diferencial compensa el retraso de la parte integral.

$$C(s) = K_P \cdot \frac{1 + Is}{Is} \cdot \frac{1 + Ds}{1 + fDs}$$

Ecuación 16. Función de transferencia para el control PID (formato serie).

Los pasos a seguir para diseñar por margen de fase son:

1. Fijar el margen de fase (Φ_m).
2. Fijar el retraso de la parte integral del control. Valores típicos están entre -5° y -10° .
3. Establecer el valor de la pulsación de cruce (w_o) y el factor de filtrado. Este último toma valores entre 0 y f_m .
4. Calcular la fase del control (φ_C) y el adelanto de fase de la parte diferencial (φ_A).

$$\angle G(jw_o) = \angle C(jw_o) + \angle P(jw_o) = -180^\circ + \Phi_m$$

$$\varphi_C = -180^\circ + \Phi_m - \angle P(jw_o)$$

$$\varphi_A = \varphi_C - \varphi_R$$

5. Calcular la ganancia del control (A_C).

$$|G(jw_o)| = |C(jw_o)||P(jw_o)| = 1 \rightarrow A_C = \frac{1}{|P(jw_o)|}$$

6. Calcular los parámetros del control:

- a. Cálculo de I

$$\varphi_R = -90^\circ + \text{atan}(Iw_o) \rightarrow I = \frac{\tan(90^\circ + \varphi_R)}{w_o}$$

- b. Cálculo de D

$$w_o D = \frac{\frac{1}{f} - 1}{2 \tan(\varphi_A)} - \sqrt{\left(\frac{\frac{1}{f} - 1}{2 \tan(\varphi_A)}\right)^2 - \frac{1}{f}}$$

- c. Cálculo de K_P

$$K_P = A_C \frac{Iw_o}{\sqrt{1 + (Iw_o)^2}} \frac{\sqrt{1 + (fDw_o)^2}}{\sqrt{1 + (Dw_o)^2}}$$

Se escoge finalmente una w_{oPID} 1,9 veces mayor que la del control P, un margen de fase de 65° , un retraso de la parte integral de -10° y un factor de filtrado $f = 0,1$. Con esas características se obtiene un adelanto de fase neto de $37,61^\circ$ (ver Figura 56), una ganancia $K = 31,18$, $T_i = 0,99$, $T_d = 0,15$ y $N = 7$. Además, se ha ajustado la respuesta aplicando una ponderación a la referencia de 0,9. Con dichos valores se obtiene la respuesta a referencia y perturbación que se muestra en la Figura 57. La salida tiene un sobrepaso aceptable del 8,76%, alcanzando el pico a los 1,54 segundos y con un tiempo de establecimiento al 5% de 2,47 segundos. Con ello se comprueba que se cumplen los objetivos del control de aumentar la rapidez de la respuesta.

Por otro lado, el control PID tiene otro objetivo de eliminar el error en régimen permanente, tanto de seguimiento como de respuesta a perturbación. En este caso el primero ya es cero

en el diseño teórico, gracias a la integración de la planta, por lo que la mejoría de la parte integral del control se puede apreciar en la respuesta a perturbación.

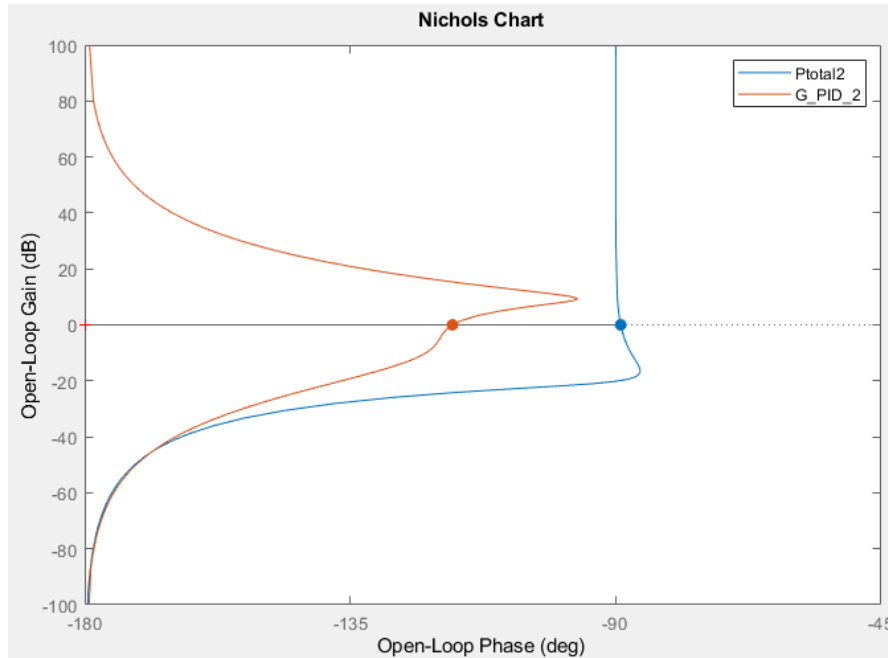


Figura 56. Gráfico de Black para la planta $P(s)$ y el lazo abierto $G(s)$ para el control PID del lazo externo.

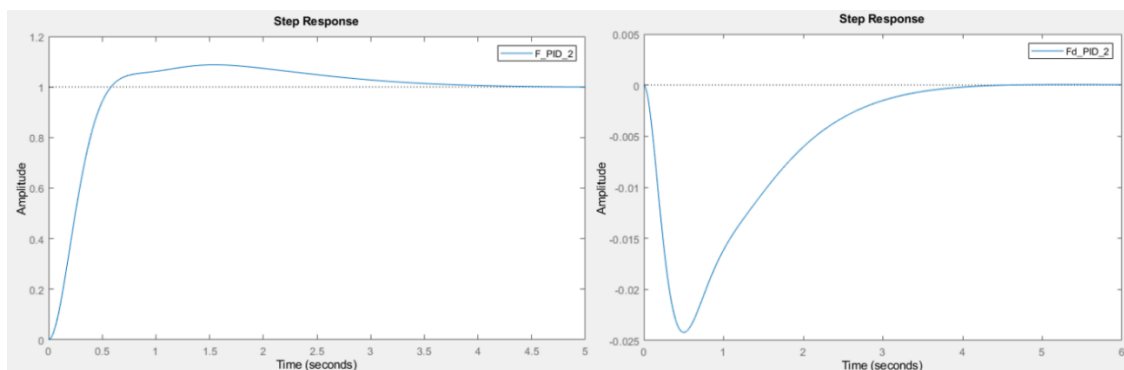


Figura 57. Respuesta a un escalón unitario en referencia y perturbación para el control PID del lazo externo.

Una vez diseñados todos los controles se presenta en la Tabla 10 una comparación de los parámetros obtenidos con un margen de fase de 65° . En la Figura 58 y en la Tabla 11 se muestra una comparación de la respuesta a un escalón unitario en referencia. Se puede observar que todos ellos tienen error nulo de seguimiento a referencia, gracias a la integración de la planta. Sin embargo, esto no tiene por qué ocurrir también en el ensayo para los que no tengan parte integral (P y PD), ya que debido a la perturbación que introduce la asimetría de los motores la distancia medida puede no alcanzar la referencia. Si se compara el tiempo de establecimiento (t_s) de las respuestas se puede concluir que el control PD es el más rápido.

Por otro lado, en la Figura 59 se muestra la respuesta a un escalón unitario en perturbación para los cuatro controles. En este caso sí se puede observar como los que tienen acción integral (PI y PID) no tienen error en régimen permanente, a diferencia de los otros dos (P y PD) que no consiguen eliminar el error. Se puede observar también que gracias al aumento de la rapidez de la parte diferencial el control PD tiene menos pico en la respuesta a perturbación, ya que se compensa más rápidamente. Esta ventaja combinada con la acción integral hace que el control PID tenga menos pico y consiga eliminar el error en régimen permanente en rechazo a perturbación.

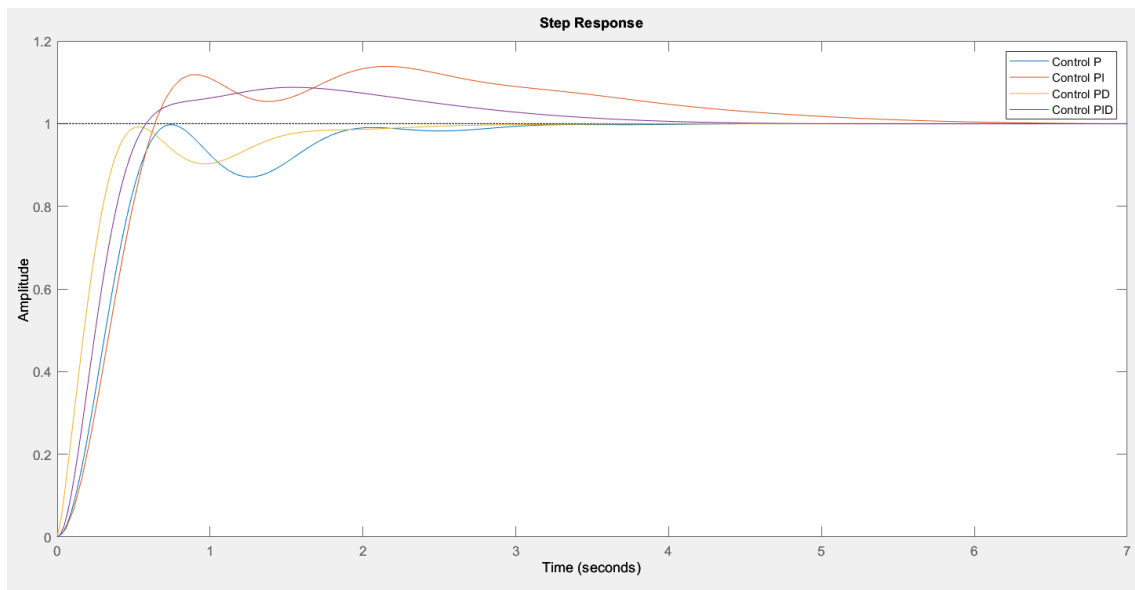


Figura 58. Comparativa de la respuesta a un escalón unitario en referencia para los controles diseñados.

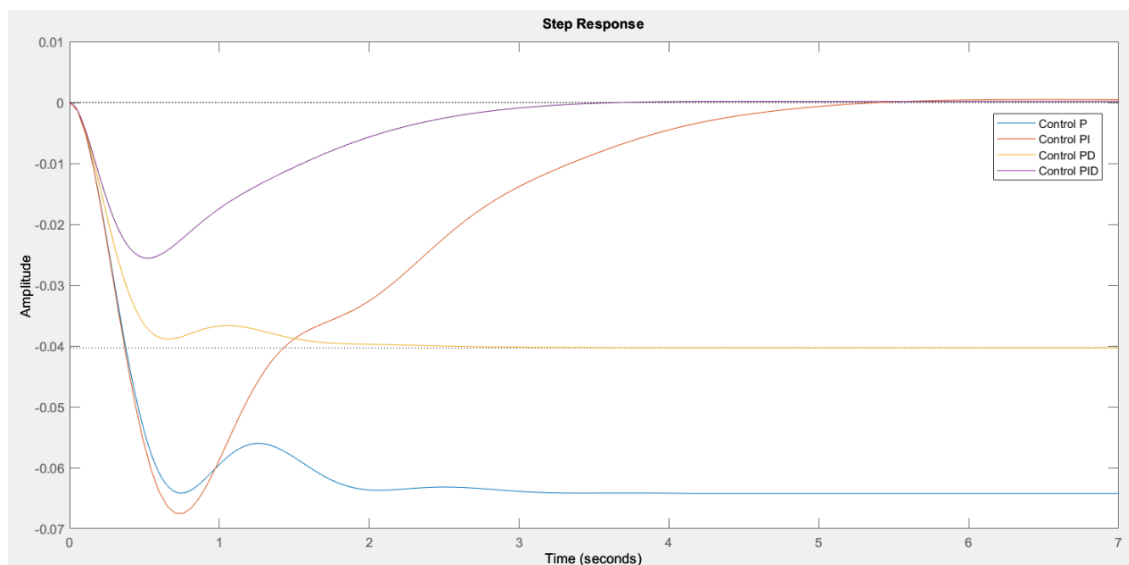


Figura 59. Comparativa a un escalón unitario en perturbación para los controles diseñados.

$\Phi_m = 65^\circ$	w_o	K	T_i	T_d	N	f	φ_{PI}	φ_{PD}
P	3,70	15,55	∞	0	-	1	-	-
PI	2,96	12,52	1,48	0	-	1	-12,85°	-
PD	5,55	24,80	∞	0,09	9,0	0,1	-	26,32°
PID	7,04	31,18	0,99	0,15	7,0	0,1	-10,00°	46,61°

Tabla 10. Resumen del diseño de los controles de seguimiento de pared.

$\Phi_m = 65^\circ$	$M_p(\%)$	$t_p(seg)$	$t_s(seg)$	E_s	E_d
P	-	-	1,67	0	0,040
PI	13,85	2,17	3,92	0	0
PD	-	-	1,31	0	0,064
PID	8,76	1,54	2,47	0	0

Tabla 11. Comparativa de la respuesta temporal para los controles diseñados.

5.3 IMPLEMENTACIÓN DEL CONTROL DE SEGUIMIENTO DE PARED

La implementación del control de seguimiento de pared se ha llevado a cabo de la misma forma que en el control de velocidad de avance. En la Figura 60 se muestra el archivo de Simulink que se ha desarrollado para implementar el control, equivalente al simplificado que se muestra en la Figura 45. Este archivo se despliega en la Raspberry Pi del robot utilizando la generación de código C++ (Simulink CODER) y la ROS *Toolbox* de MATLAB. Al igual que ocurría con el control de velocidad, se ha implementado el control de forma que todos los parámetros y consignas que se necesitan se envíen desde otros archivos mediante *topics*, de forma que no haya que esperar para desplegar cada vez que se necesite modificar algo. Con ello en este control se tienen:

- *Topic* /control_reference_dn: recibe un mensaje de tipo std_msgs/Float32 con la consigna de distancia a la pared deseada.
- *Topic* /control_on: está dentro del subsistema 'U(s)' y recibe un mensaje de tipo std_msgs/Bool con la variable de activación del control. Al igual que ocurría con el

control de velocidad, se trata de una variable que permite evitar que el robot continúe moviéndose si hay un problema con la parada del nodo del control tras finalizar el ensayo. En este caso se ha añadido una condición adicional al arranque del robot y es que, aparte de que `/control_on` esté activo, la referencia tiene que ser mayor que el umbral indicado. Esto se ha hecho así ya que la tensión común aplicada a los motores se recibe como un parámetro de ROS junto con el resto de los parámetros del control. Sin embargo, la referencia se envía a través del *topic* `/control_reference_dn`, que llegaba ligeramente más tarde que la tensión común. Esto provocaba que se empezase a mover el robot con una referencia de cero metros, causando un comienzo muy brusco. Como se muestra en la Figura 61 no se manda tensión a los motores hasta que no se cumplen ambos requisitos. Además, estas condiciones se utilizan también para controlar la variable ‘Activación’. Esta se encarga de eliminar el error acumulado que puedan tener los bloques de integración del control previo al arranque.

- *Topic* `/cmd_vel`: se encuentra también dentro del subsistema ‘U(s)’ y publica un mensaje de tipo `geometry_msgs/Twist` con la componente común y la diferencial del mando. Previamente al envío de las mismas se hace una comprobación para asegurar que la suma de ambas no es superior a los límites de $\pm 100\%$. En caso de superarlos se limita la tensión diferencial y se activa la variable ‘AW_interno’ para evitar que la parte integral del control acumule error. Se ha escogido limitar el giro en caso de saturación ya que la planta del control externo (ver Ecuación 12) depende del punto de operación y, por lo tanto, de la velocidad de avance. Por ello si se limita la tensión común, dando prioridad al giro, se está cambiando la planta con la que se han diseñado los controles. Es posible que a pesar de ello sea necesario dar prioridad al giro si en una situación práctica se choca con la pared de no hacerlo.
- *Topic* `/lidar`: recibe un mensaje de tipo `geometry_msgs/TwistStamped` con la medida de distancia y de ángulo. Estas se calculan en el nodo `/tof_node`, que recibe las medidas de los sensores ToF y realiza las operaciones necesarias (ver Ecuación 5) para obtener la distancia y el ángulo.
- *Topic* `/measurements`: publica un mensaje de tipo `geometry_msgs/TwistStamped` con las medidas de ángulo, distancia, sus respectivas referencias y el mando diferencial para su posterior monitorización.
- Por último, existe también la variable ‘AW externo’ que permite evitar que la parte integral del control externo acumule error en caso de que la referencia de ángulo saturase. Los límites de esta saturación se han establecido en $\pm 45^\circ$, aunque se podría estudiar la conveniencia de utilizar otro valor.

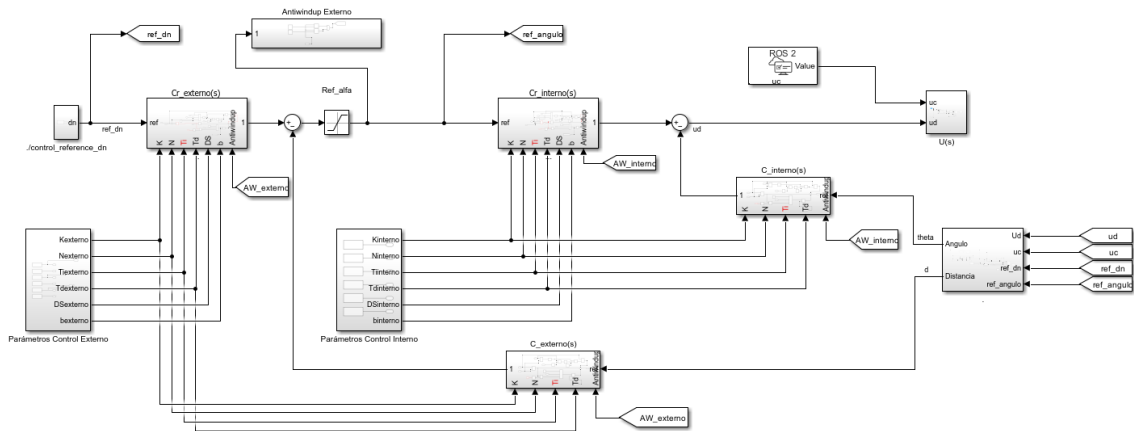


Figura 60. Diagrama del control de seguimiento de pared desplegado.

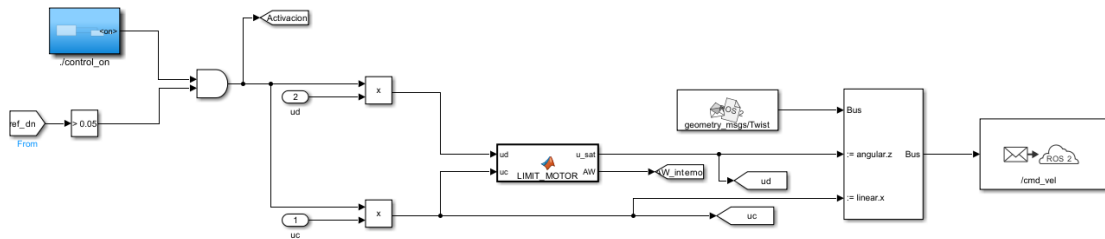


Figura 61. Subsistema $U(s)$.

Por último, en la Figura 62 se muestra el otro archivo de Simulink necesario. Este archivo se encarga de enviar las consignas necesarias mediante *topics*, por lo que se puede modificar sin necesidad de volver a compilar el código. En este archivo se define el tiempo del ensayo y la consigna de distancia a la pared (tipo de onda, amplitud, periodo...). Además, también se publica `/control_on` para gestionar la activación del control.

Por último, se reciben las medidas del ensayo (publicadas mediante `/measurements`) y se guardan en el *workspace* de MATLAB para su monitorización.

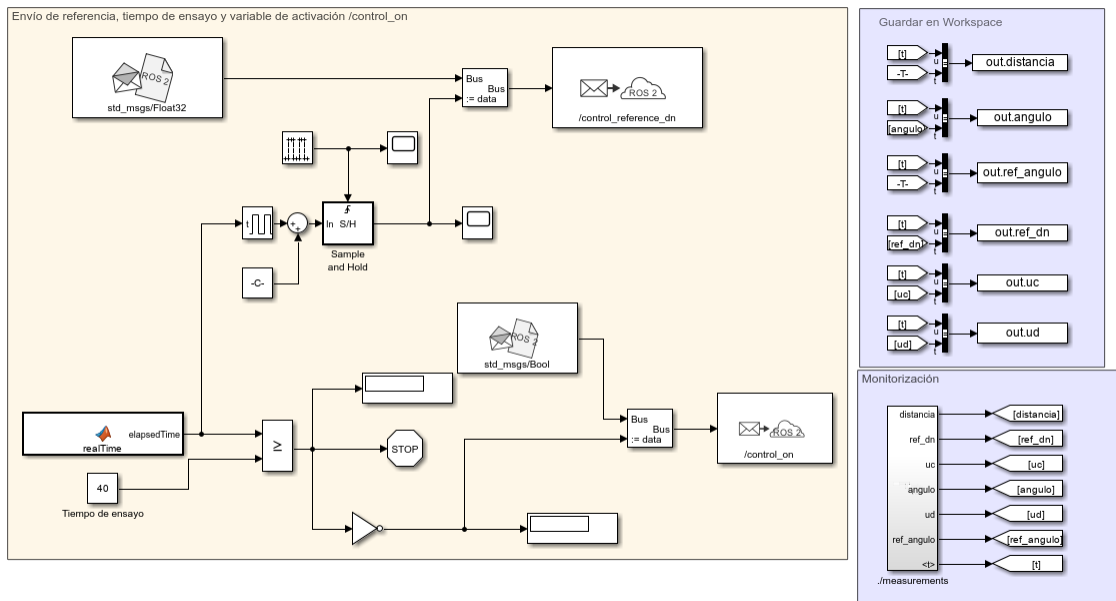


Figura 62. Archivo de monitorización y envío de referencia para el control de seguimiento de pared.

En la Figura 63 se muestra el gráfico de nodos del control de seguimiento de pared. Se puede observar que hay siete nodos (elipses) y siete *topics* (rectángulos). Los nodos `/WallFollowerNode` y `/Envio_referencia90580` corresponden a los dos archivos de Simulink mencionados anteriormente, el resto de los nodos son creados por el robot. El nodo `/Envio_referencia90580` publica `/control_refernce_dn` y `/control_on`, mientras que el nodo `/WallFollowerNode` tiene los suscriptores asociados para poder recibir los mensajes. Este último nodo se encarga de publicar la consigna de mando común y diferencial a través de `/cmd_vel`. Además, también publica `/measurements` con las medidas del ensayo. Por último, el nodo `/tof_node` se encarga de recibir las medidas de los sensores ToF, procesarlas según la Ecuación 5 y publicarlas a través de `/lidar`, para que puedan ser usadas por el nodo del control.

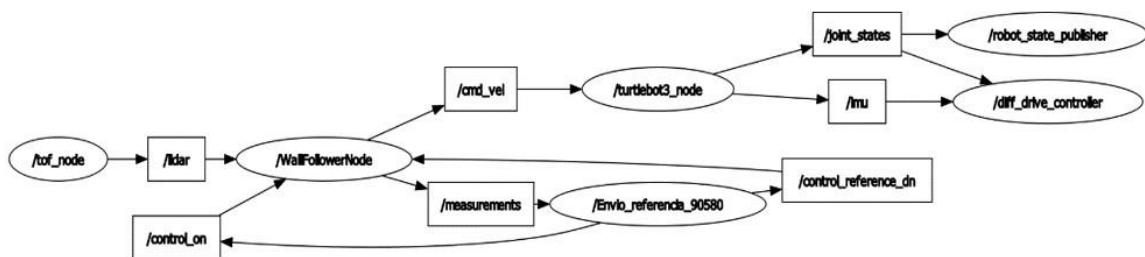


Figura 63. Gráfico de nodos del control de seguimiento de pared.

5.4 COMPARACIÓN ENTRE SIMULACIÓN Y ENSAYO PARA EL CONTROL DE SEGUIMIENTO DE PARED

Una vez implementados los controles de seguimiento de pared, en este apartado se van a comparar los resultados obtenidos de los ensayos frente a los obtenidos por simulación. En

este caso hay tres gráficas. Por un lado, se va a comparar la respuesta de la salida del control (la distancia medida) frente a la referencia dada. Además, se estudiará el ángulo del robot con respecto a la pared y la referencia de este y, por último, el mando diferencial calculado por el control.

La comparación entre el ensayo y la simulación para el control proporcional diseñado se muestra en la Figura 64. Se puede observar que en la medida de distancia existe un retardo de 0,2 segundos, debido a los retrasos en las comunicaciones de los diferentes elementos. Se ha decidido no modelar dicho retardo finalmente ya que incluirlo no mejoraba de forma significativa el comportamiento de la simulación frente al ensayo.

Por otro lado, se puede observar que la simulación es más rápida que el ensayo si se miden los tiempos en los que se produce el pico de la primera oscilación en la distancia (0,77 segundos en la simulación y 1,06 segundos aproximadamente en el ensayo). En ambos casos se observa que no hay sobrepaso, aunque el ensayo alcanza valores menores. Además, se puede apreciar que la simulación no tiene error en régimen permanente en seguimiento a referencia, al igual que el diseño teórico del apartado “Diseño del control del lazo externo”. Sin embargo, en el ensayo sí se aprecia que la distancia medida no alcanza el valor de referencia. Esto es causado por la perturbación d_1 (ver Figura 45) que representa la asimetría de los motores. Esta perturbación no está incluida en la simulación, por eso existe esa diferencia con el ensayo.

Con respecto al ángulo, se comprueba que la simulación es más rápida que el ensayo, aunque este alcanza valores superiores. Esto podría estar causado por el retardo no modelado en la distancia, sin embargo, debido a la realimentación es complejo determinar cuál es la causa y cuál la consecuencia. El ángulo simulado tiene su valor máximo en 10° a los 0,56 segundos mientras que el medido tiene un valor máximo de $12,57^\circ$, alcanzándolo a los 0,76 segundos. La referencia de ángulo del ensayo se mantiene con valores mayores durante más tiempo que la simulada. Esto concuerda con lo que se observa en la gráfica de la distancia ya que la medida está por debajo de la simulada y es más lenta, por lo que la referencia de ángulo va reduciendo su valor más despacio.

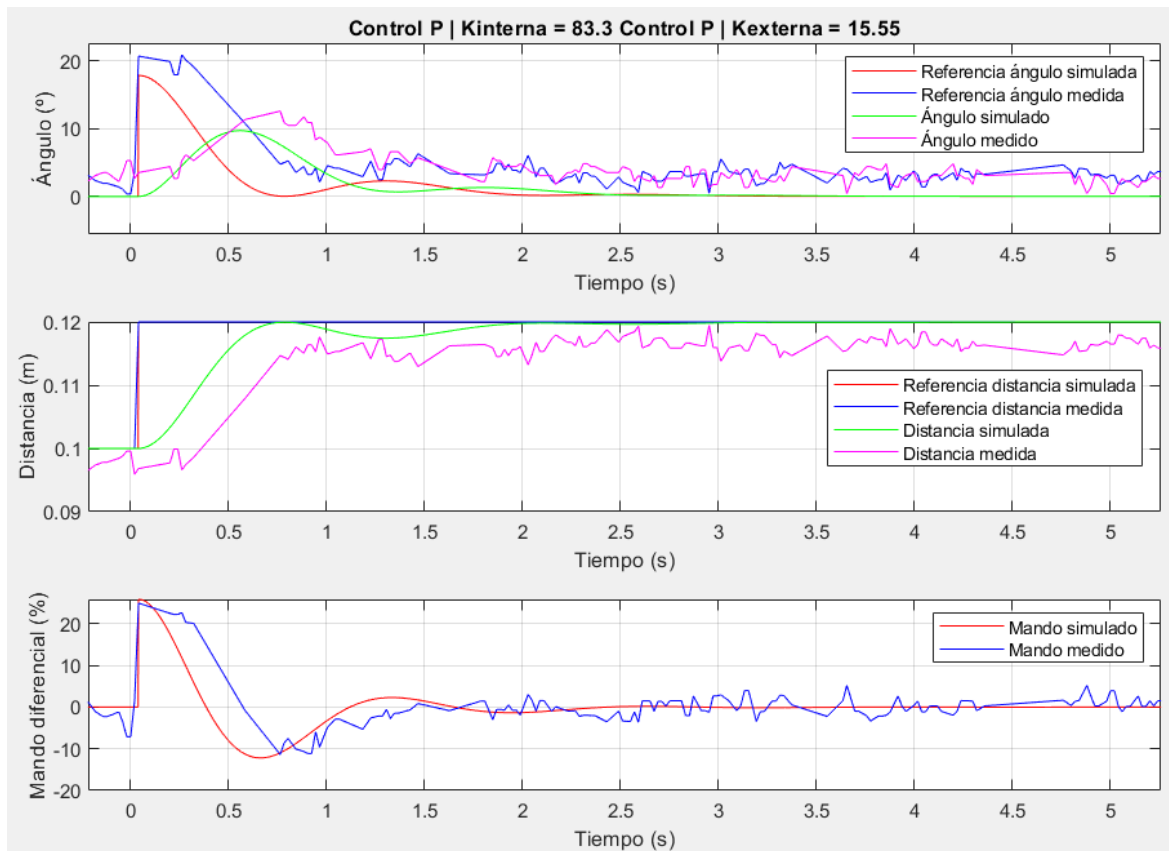


Figura 64. Comparación entre ensayo y simulación para el control P.

En la Figura 65 se muestra la comparación entre ensayo y simulación para el control PI, con los parámetros que se indican en la figura. Para la medida de distancia se observa que el ensayo tiene un sobrepaso mayor (7,16% frente al 2,31% de la simulación), aunque sigue estando dentro de los límites aceptables. Con respecto al tiempo donde se produce ese pico, en simulación es antes (0,82 segundos) que en el ensayo (0,98 segundos). En ambos casos se puede observar que se alcanza la referencia de distancia en régimen permanente. En el ensayo esto es gracias a la integración del control, que elimina el error causado por la asimetría de los motores.

Con respecto a la medida de ángulo, se puede observar que el ensayo tiene diferencias con la simulación, alcanzando valores máximos de aproximadamente 15° frente al pico de 10° en simulación. Además, se comprueba que la simulación es más rápida que el ensayo, si se mide como el instante donde se produce el pico (0,64 segundos frente a 0,86 segundos respectivamente). La referencia de ángulo del ensayo alcanza valores mayores que la simulada, además de que va reduciendo su valor más lentamente. Esto es coherente con la respuesta de la distancia medida, que es más lenta que la simulada, por lo que la referencia del ángulo se mantiene con valores mayores durante más tiempo. Tanto en simulación como en ensayo el valor en régimen permanente del ángulo coincide con el de su referencia, sin embargo, en el caso del ensayo no es exactamente cero si no ligeramente superior (aproximadamente 3°). Esto se debe a que las medidas de los sensores ToF no son perfectas y pueden verse afectadas por el ruido de medida.

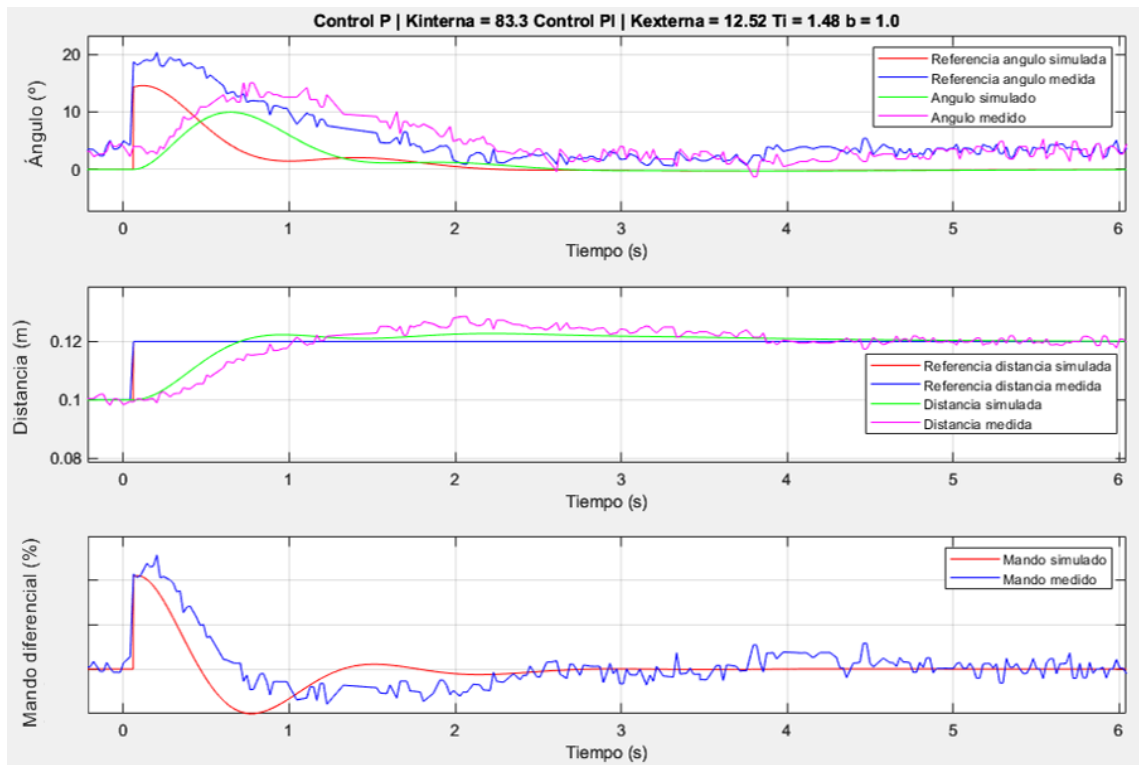


Figura 65. Comparación entre ensayo y simulación para el control PI.

En la Figura 66 se muestra la comparación entre ensayo y simulación para un control PD con acción diferencial a la salida, para los parámetros indicados en la misma. En este caso, al igual que en el control anterior, el ángulo medido alcanza valores superiores al simulado (17° del ensayo frente a los 10° de la simulación) y es más lento, alcanzando el pico a los 0,64 segundos frente a los 0,50 segundos de la simulación. La referencia de ángulo se mantiene con valores mayores a los simulados, lo que concuerda con las medidas de distancia a la pared del ensayo, más lentas también que la simulación. En ambos casos la distancia no tiene sobrepaso, manteniéndose por debajo del valor de referencia durante el transitorio. Se observa que el ensayo es más lento que la simulación también en la gráfica de distancia. Al igual que ocurría con el control P, en la simulación no hay error en régimen permanente para la medida de distancia ya que no se ha incluido la asimetría de los motores. Pero en el ensayo sí se observa que la salida se queda cerca pero no alcanza el valor de referencia.

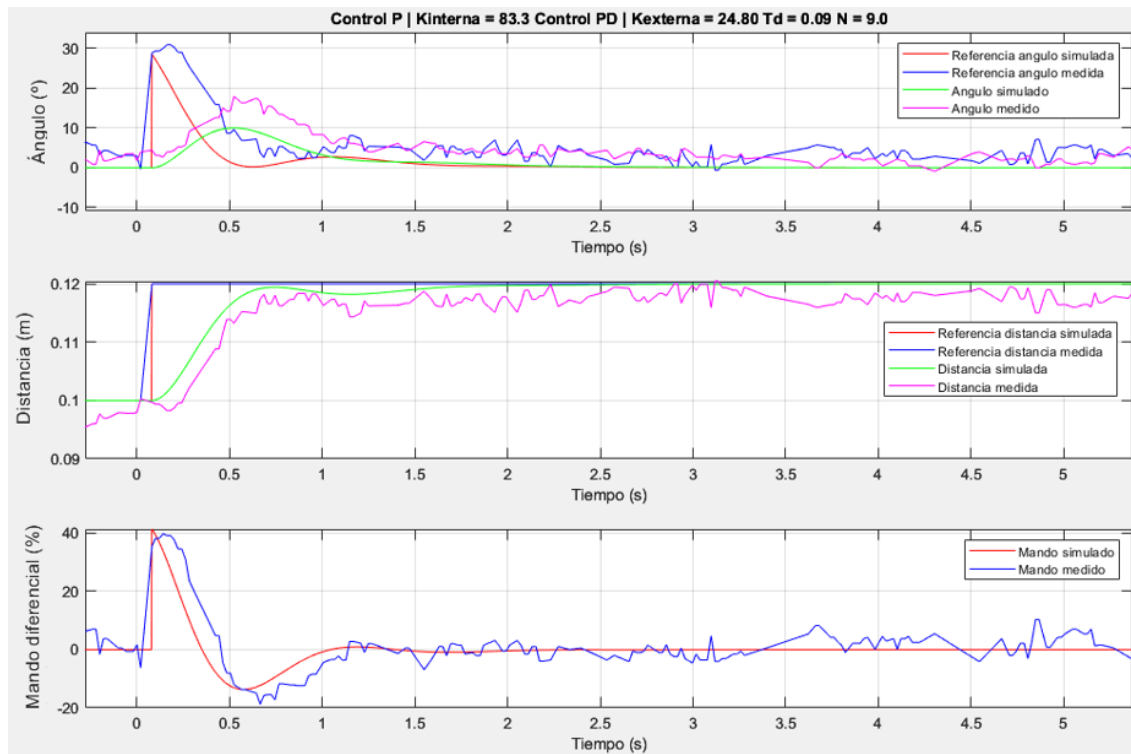


Figura 66. Comparación entre ensayo y simulación para el control PD con acción diferencial a la salida.

Por último, en la Figura 67 se muestra la comparación entre ensayo y simulación para un control PID con acción diferencial a la salida. Se observa que la respuesta de la distancia simulada tiene un sobrepaso del 1,46%, alcanzando el pico a los 1,54 segundos. Con respecto a la medida de distancia obtenida del ensayo, no se puede determinar el tiempo de pico ni el valor del sobrepaso con exactitud debido al ruido de medida, sin embargo, se puede apreciar que es similar a la simulación. En este caso, gracias a la parte integral del control, la distancia medida alcanza el valor de referencia en el ensayo, así como en simulación. Por otro lado, el ángulo medido alcanza valores superiores que los de la simulación (20° frente a 10°), además de ser ligeramente más lento que la simulación si se comparan los tiempos del pico en el ángulo (0,55 segundos frente a los 0,50 de la simulación).

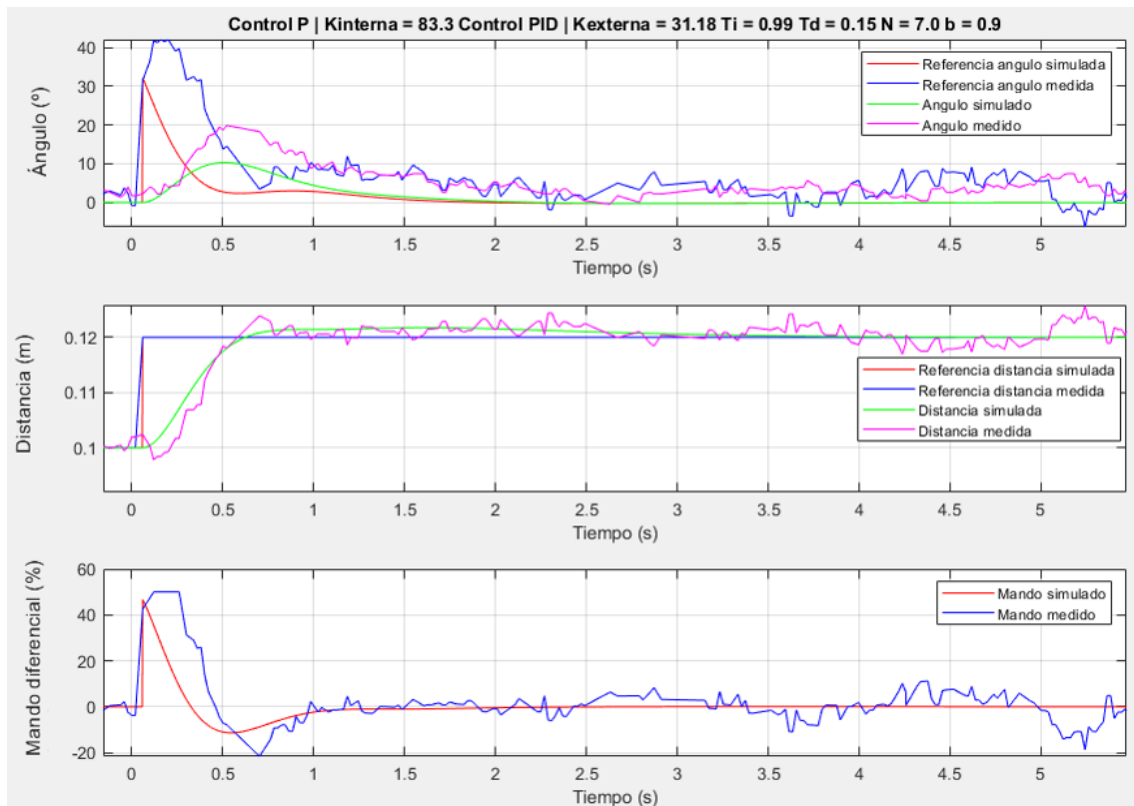


Figura 67. Comparación entre ensayo y simulación para el control PID con acción diferencial a la salida.

5.5 RESPUESTA A PERTURBACIÓN DE LA CURVA DEL CONTROL DE SEGUIMIENTO DE PARED

En este apartado se va a estudiar cómo se comporta el robot frente a una perturbación causada por la curva del circuito. Esta perturbación se corresponde con el cambio que se produce en el ángulo cuando llega a la misma, en el diagrama de la Figura 45 se indicaba con d_2 . Esta curva se puede modelar como si se tratase de otro cuerpo girando con una velocidad de rotación como se indica en la Ecuación 17.

$$T_p = \frac{\pi \cdot r_{curva}}{v}$$

$$w_{pared} = \frac{\pi}{T_p} = \frac{v}{r_{curva}}$$

Ecuación 17. Velocidad de rotación de la pared.

Una característica que destacar de los ensayos realizados es que la curva del circuito no es un semicírculo perfecto, si no que tiene un pequeño tramo recto en medio. Esto hace que en todas las medidas de las gráficas se observe como parece que hay dos perturbaciones seguidas.

En la Figura 68 se muestra la respuesta del control P frente a la curva, que comienza aproximadamente a los 9 segundos hasta los 21 segundos. Se observa en la gráfica cómo la distancia se reduce hasta los 0,105 metros aproximadamente, sin conseguir recuperar el valor que tenía antes de la perturbación hasta que no termina la curva.

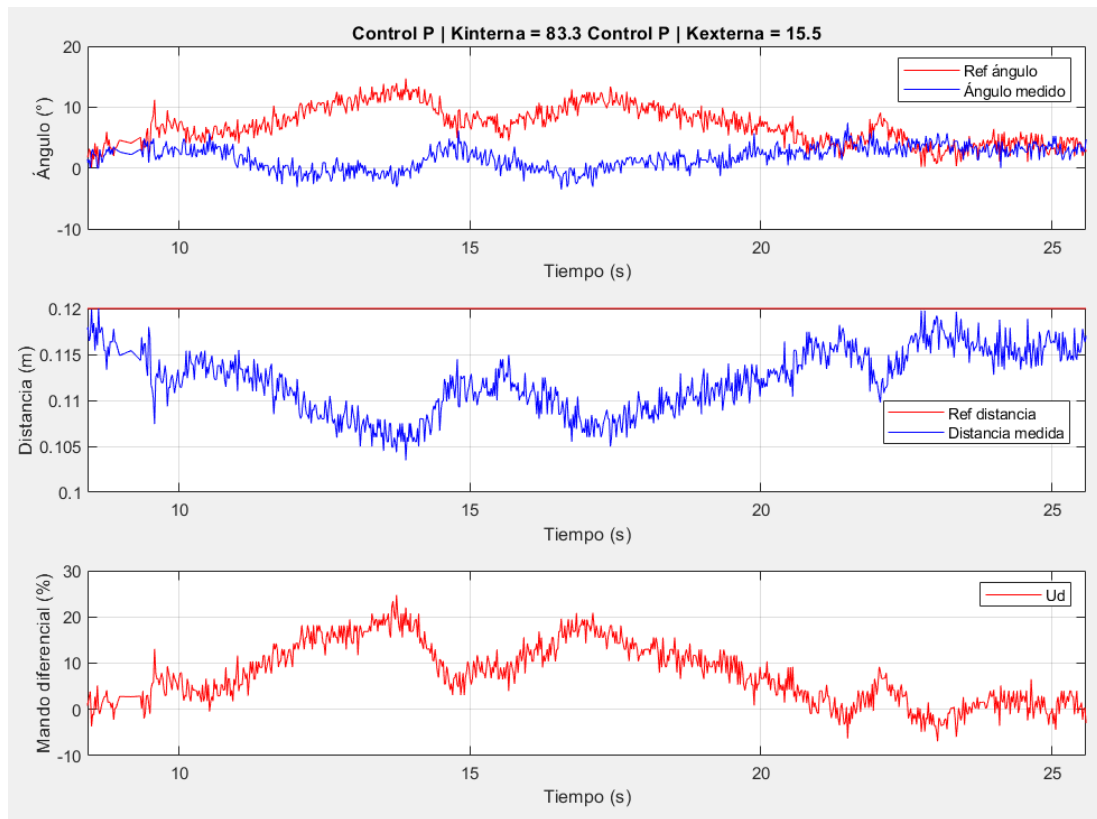


Figura 68. Respuesta a perturbación para el control P.

Con respecto a la respuesta a perturbación, que dura desde los 12 hasta los 25 segundos aproximadamente, del control PI (ver Figura 69) se puede observar cómo en este caso la distancia se reduce menos, el robot se acerca menos a la pared. Además, gracias a la acción integral el control intenta recuperar el valor de referencia, sin embargo, corrige en exceso y se aleja demasiado de la pared antes de conseguir finalmente recuperar el valor de referencia.

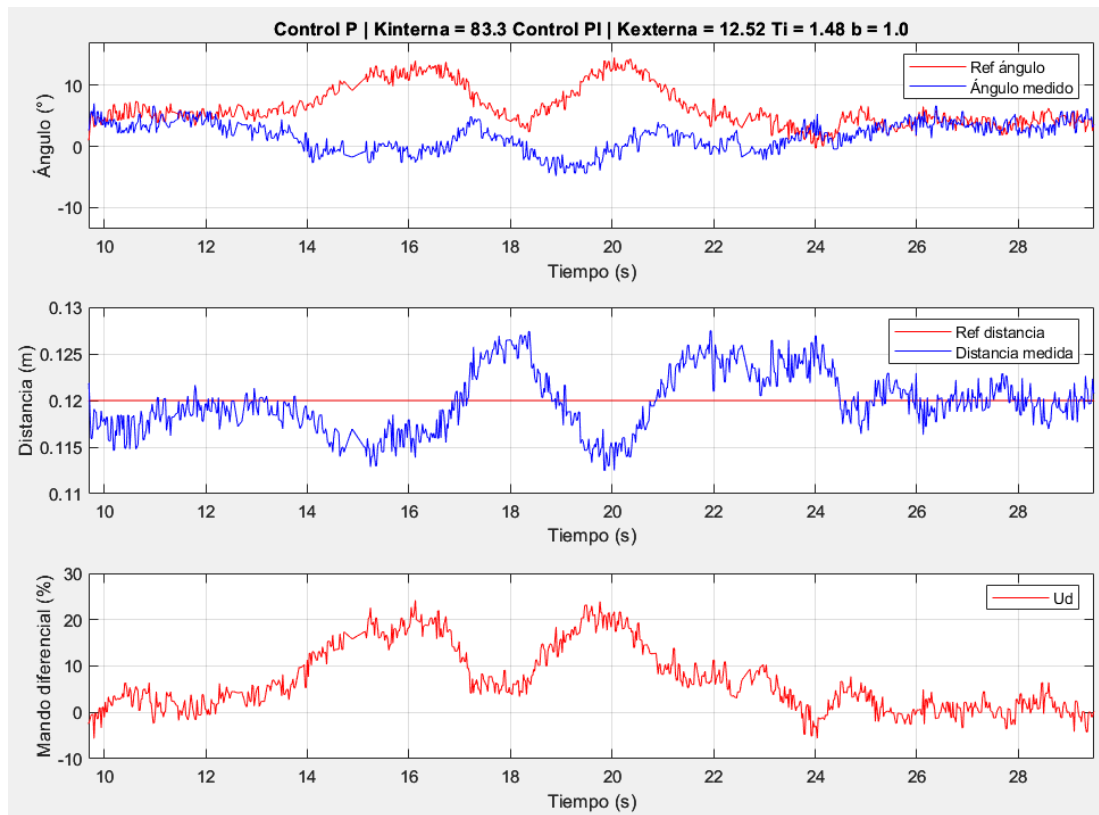


Figura 69. Respuesta a perturbación para el control PI.

En la Figura 70 se muestra la respuesta a perturbación del control PD con acción diferencial a la salida. Esta perturbación comienza a los 12 segundos y termina a los 22 aproximadamente. En este caso se puede comprobar que, al igual que con el control P, el PD no quita el error y solo recupera el valor de la medida de distancia anterior a la perturbación una vez que se termina la curva. La diferencia de este control con el P es que este tiene menos pico en la respuesta debido a la parte diferencial, que compensa antes la perturbación gracias al incremento en la rapidez.

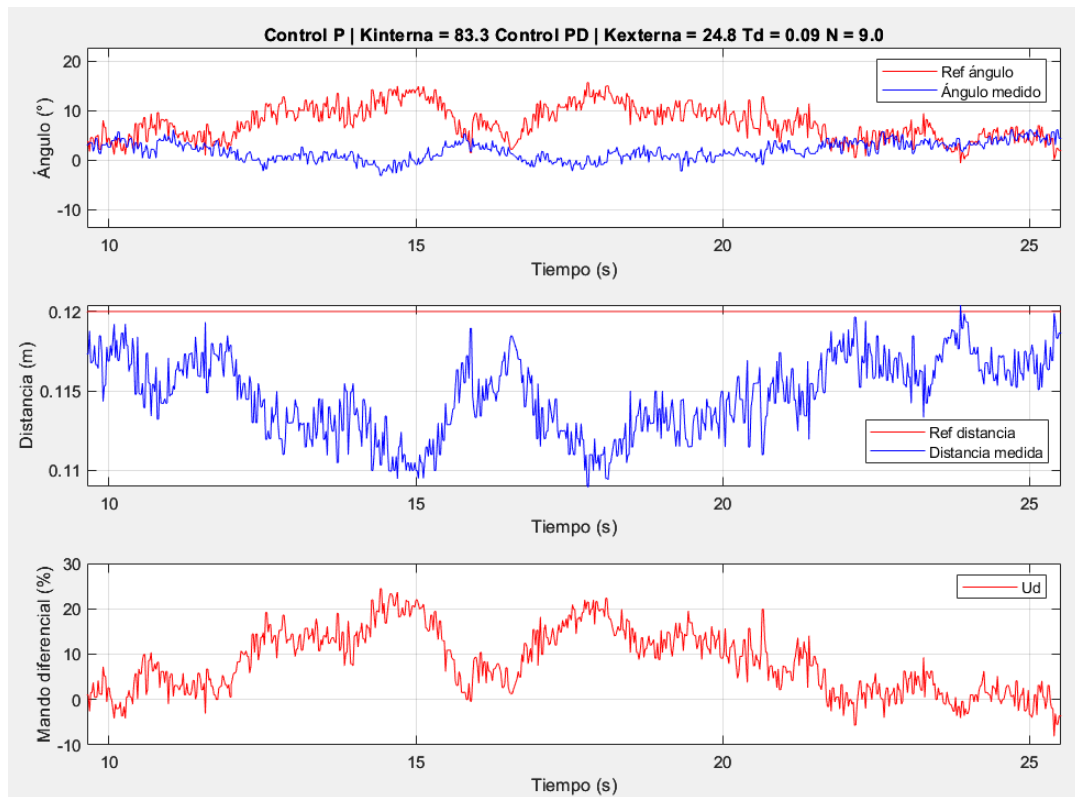


Figura 70. Respuesta a perturbación para el control PD con acción diferencial a la salida.

Por último, la respuesta frente a la perturbación de la curva del control PID se puede observar en la Figura 71. Esta empieza a los 12 segundos y termina aproximadamente a los 21 segundos. En este caso, al igual que ocurría con el PI, la parte integral consigue anular el error de perturbación. Además, gracias a la parte diferencial la perturbación se compensa más rápidamente y se consigue un menor pico en la respuesta, haciendo que este sea el control que hace que el robot se acerque menos a la pared durante la curva.

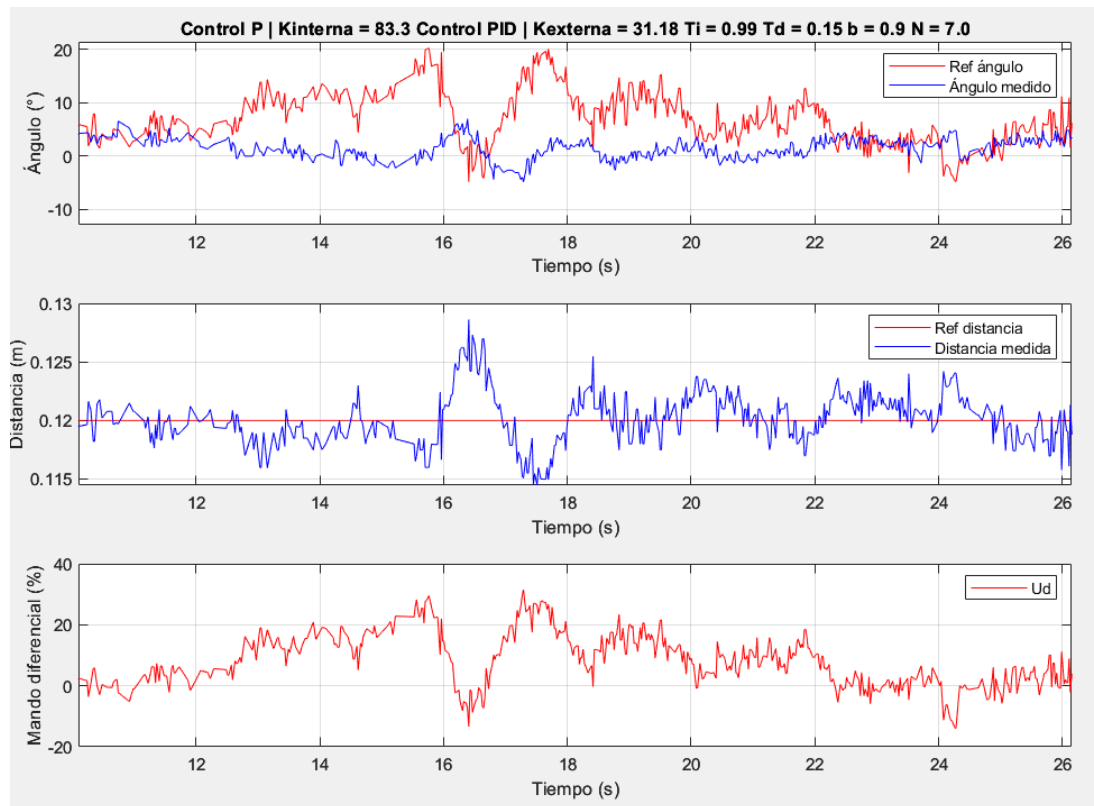


Figura 71. Respuesta a perturbación para el control PID con acción diferencial a la salida.

5.6 CONCLUSIONES DEL CONTROL DE SEGUIMIENTO DE DISTANCIA A LA PARED

Tras analizar la respuesta de todos los controles diseñados, tanto en simulación como en ensayo, y de comparar la respuesta a perturbación en el ensayo se explican en este apartado las conclusiones obtenidas.

Con el control proporcional no se ha obtenido sobrepaso y el tiempo donde se alcanza el pico de la primera oscilación es a los 1,06 segundos. El control PD tampoco tiene sobrepaso y alcanza el pico de la primera oscilación a los 0,85 segundos, más rápido que el proporcional. Con el control PI se obtiene un sobrepaso razonable de 7,16% aproximadamente a los 0,98 segundos. El control PID consigue un sobrepaso menor, del 1,46% a los 1,54 segundos.

Como se ha comentado con anterioridad gracias a la integral de la planta el error de seguimiento a referencia es nulo en la simulación, por lo que aplicar la parte integral afectará a la respuesta a perturbación. Sin embargo, se ha observado que esto no es así en el ensayo donde, debido a la asimetría de los motores, solo el control PI y PID consiguen alcanzar la referencia en régimen permanente. Tras comparar las respuestas de los distintos controles en seguimiento a referencia y rechazo a la perturbación de la curva se escoge finalmente un control PID. Con él se consigue más rapidez que con los controles P y PI y se consigue obtener error nulo en seguimiento a referencia. Además, la respuesta a la perturbación tiene

menos pico gracias a la parte diferencial que permite compensar con más rapidez la perturbación y que el robot se acerque lo menos posible a la pared durante la curva.

5.7 ESTUDIOS COMPLEMENTARIOS

Como se ha mencionado en el apartado de “Identificación de la planta”, $P_2(s)$ depende del punto de operación. En concreto de la coordenada en el eje x del punto A y de la velocidad v_o (ver Ecuación 12). Por ello, en este apartado se ha estudiado cual sería el efecto de modificar la consigna de velocidad, es decir, la tensión común aplicada a los motores, en el comportamiento del robot. Para ello se ha diseñado otro control PID, esta vez para una consigna de velocidad de 0,15 m/s en lugar de 0,1 m/s (ver Figura 73). Si se compara dicho control con el PID diseñado para una velocidad menor (ver Figura 72) se observa que el primero es más oscilante en el ensayo, además de tener un sobrepaso mayor. Este comportamiento también está presente en el resto de las medidas (ángulo y tensión diferencial). Sin embargo, esto no ocurre en la simulación. La causa más probable de esto es el retardo no modelado, que parece afectar más cuanto mayor es la velocidad del punto de trabajo.

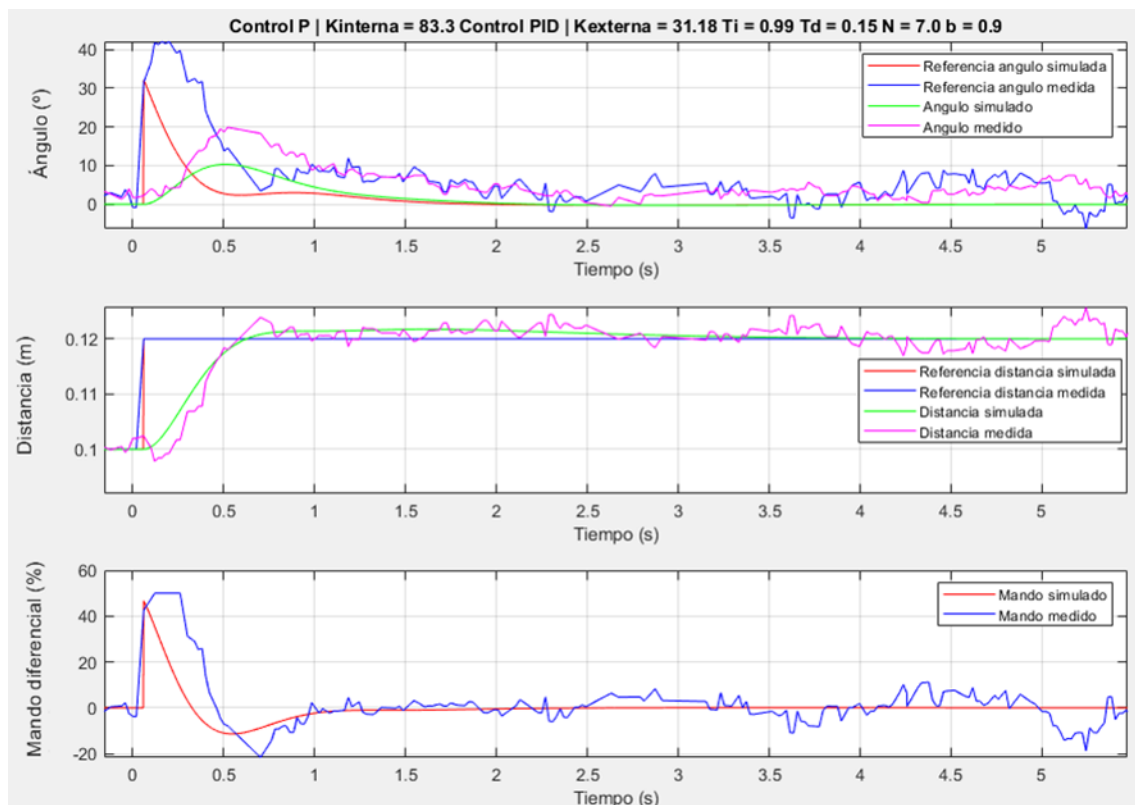


Figura 72. Control PID diseñado y ensayado con una velocidad de 0,1 m/s.

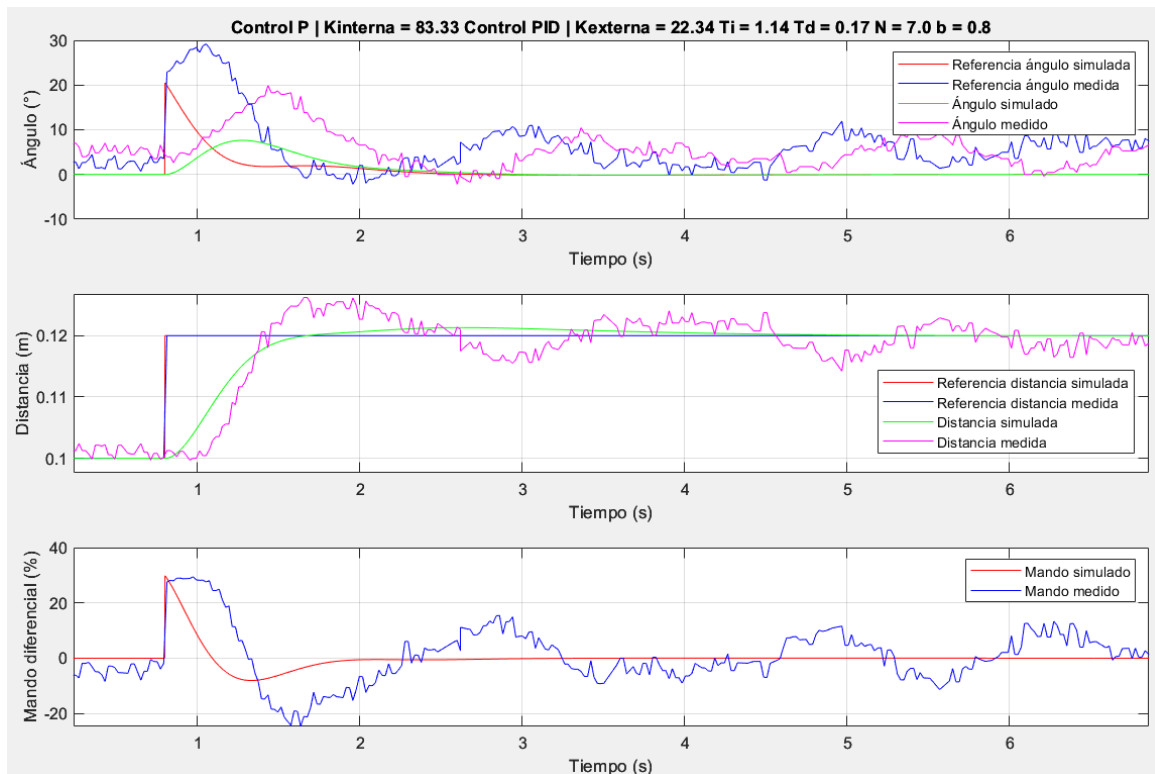


Figura 73. Control PID diseñado y ensayado con una velocidad de 0,15 m/s.

A continuación, se han intercambiado los controles para ver cómo se comportaría el robot con un diseño para una velocidad distinta a la ensayada. En la Figura 75 se muestra cómo se comporta un control PID diseñado para una velocidad de 0,1 m/s si se hace funcionar al robot a una velocidad de 0,15 m/s. Lo contrario se muestra en la Figura 74, donde el robot tiene una consigna de velocidad menor que con la que se ha diseñado. Se puede observar como el comportamiento en el ensayo si se trabaja a una velocidad menor que la diseñada no afecta tanto a los resultados. El control no es oscilante y el sobrepaso obtenido es aceptable, incluso parecido al simulado. Esto no ocurre si la velocidad de trabajo es mayor que la usada para el diseño, donde se comprueba que el control es mucho más oscilante y presenta un sobrepaso excesivo en el ensayo.

Por lo tanto, se puede concluir que la elección del punto de trabajo a la hora de diseñar controles sí afecta al comportamiento del robot en los ensayos, existiendo una diferencia más importante si se utiliza una velocidad mayor en el ensayo que la escogida para el diseño. Para tener en cuenta esta característica de la planta se podría diseñar por otras técnicas de control más avanzadas, como puede ser el control adaptativo.

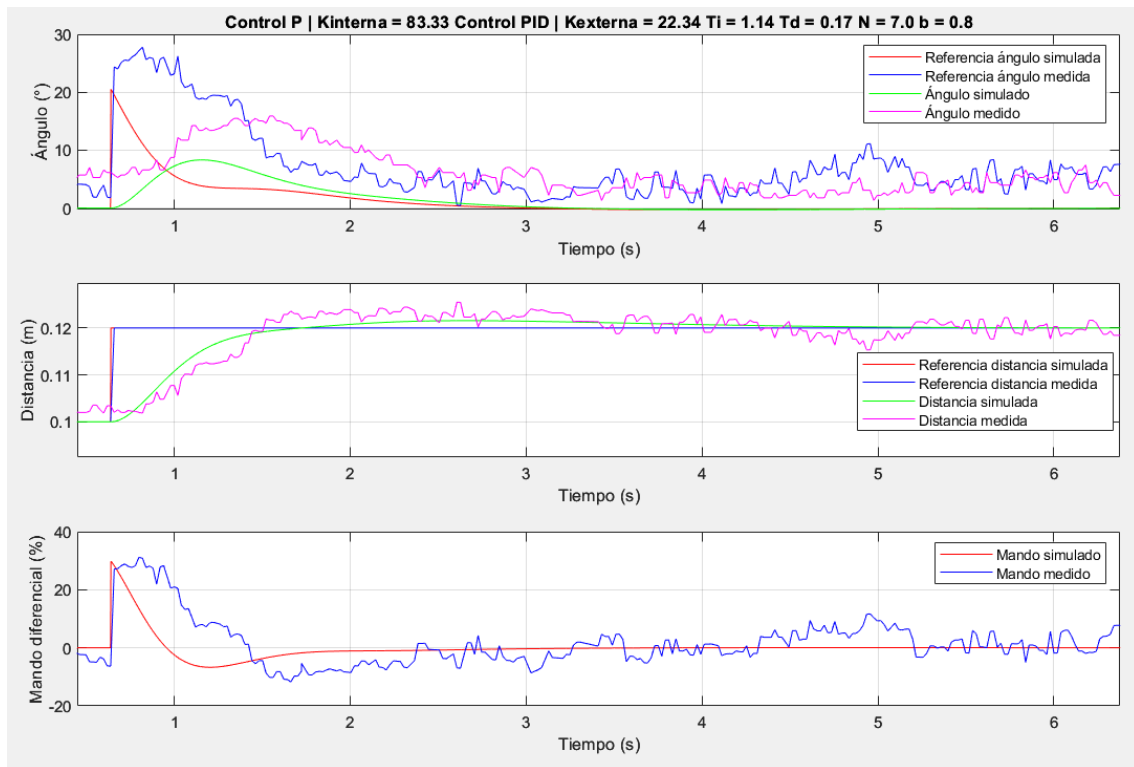


Figura 74. Control PID diseñado para una velocidad de 0,15 m/s y ensayado con una velocidad de 0,1 m/s.

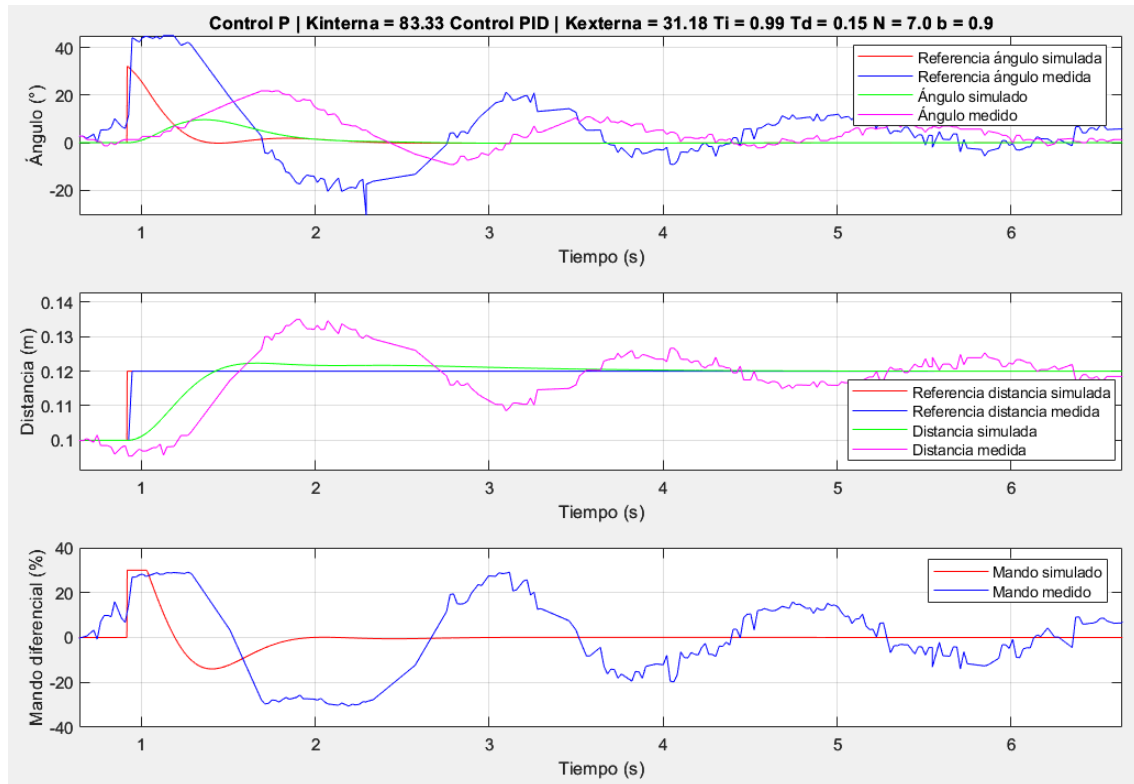


Figura 75. Control PID diseñado para una velocidad de 0,1 m/s y ensayado con una velocidad de 0,15 m/s.

5.8 CONTROL DE SEGUIMIENTO DE PARED CON EL LiDAR

En este apartado se va a estudiar el comportamiento del robot si se usa el LiDAR en lugar del sensor ToF para medir la distancia a la pared. Para ello se muestra en la Figura 76 que el LiDAR proporciona dos medidas, d_1 y d_2 , separadas un ángulo Φ . Este valor se puede modificar en función de lo adelantada que se desee que esté la medida. Para este proyecto se ha seleccionado un ángulo de 14° , pero se podría estudiar la elección de otros valores.

Si se define el punto A como el centro del LiDAR se puede obtener la distancia mínima desde el mismo a la pared (d_n) y el ángulo con respecto a la misma (θ) como:

$$\theta = \text{atan} \left(\frac{d_1 - d_2 \cdot \cos(\Phi)}{d_2 \cdot \sin(\Phi)} \right)$$

$$d_n = d_1 \cdot \cos(\theta)$$

Ecuación 18. Medidas de ángulo y distancia a la pared con el LiDAR.

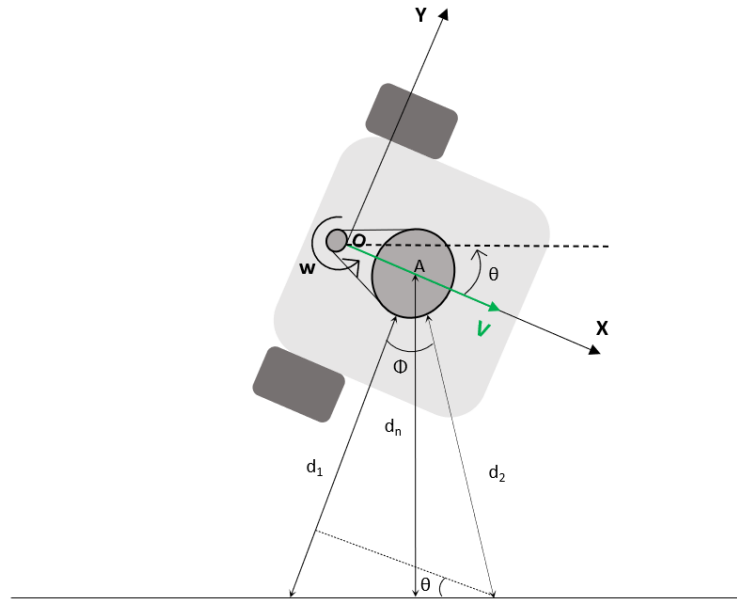


Figura 76. Medidas de ángulo y distancia a la pared con el LiDAR.

Siguiendo el mismo método de identificación de la planta que con el sensor ToF [30], se define V como la velocidad de avance del punto medio de las ruedas, x_A e y_A como las coordenadas del punto A en el sistema de referencia solidario al robot y w como la velocidad angular de rotación, se puede definir la velocidad del punto A, como si este perteneciese al vehículo, como:

$$\begin{bmatrix} v_{Ax} \\ v_{Ay} \end{bmatrix} = \begin{bmatrix} v \\ 0 \end{bmatrix} + w \cdot \begin{bmatrix} y_A \\ x_A \end{bmatrix} = \begin{bmatrix} v \\ w \cdot x_A \end{bmatrix}$$

Ecuación 19. Velocidad del punto A con el LiDAR.

Utilizando la ecuación anterior y la Ecuación 7, la Ecuación 8 y la Ecuación 9 y linealizando para un punto de operación $v = v_o$ se obtiene el modelo de la planta que se muestra en la Ecuación 20. Al igual que ocurría con el sensor ToF, esta planta se puede expresar como el producto de dos, $P_1(s)$ ($\theta(s)/U_d(s)$) y $P_2(s)$ ($D_n(s)/\theta(s)$).

$$P(s) = \frac{D_n(s)}{U_d(s)} = P_1(s) \cdot P_2(s) = \frac{K_m}{s \cdot (T_m s + 1)} \cdot \frac{v_o \cdot \left(1 + \frac{x_A}{v_o} s\right)}{s}$$

Ecuación 20. Función de transferencia de la planta del control de seguimiento de pared con el LiDAR.

Una vez obtenido el modelo de la planta es necesario determinar los valores numéricos de la misma. La identificación se ha realizado con un control P en ambos lazos, el lazo externo con una ganancia de 20,0 y el lazo interno de 90,0 (ver Figura 77). Para la identificación de la planta $P_1(s)$ se ha utilizado el método de mínimos cuadrados, obteniéndose el resultado que se muestra en la Figura 78 y los parámetros que se indican en la Ecuación 21. Se puede observar que la constante de tiempo obtenida es la misma que se obtuvo para la planta identificada con el sensor ToF, sin embargo, la ganancia estática de esta planta es menor.

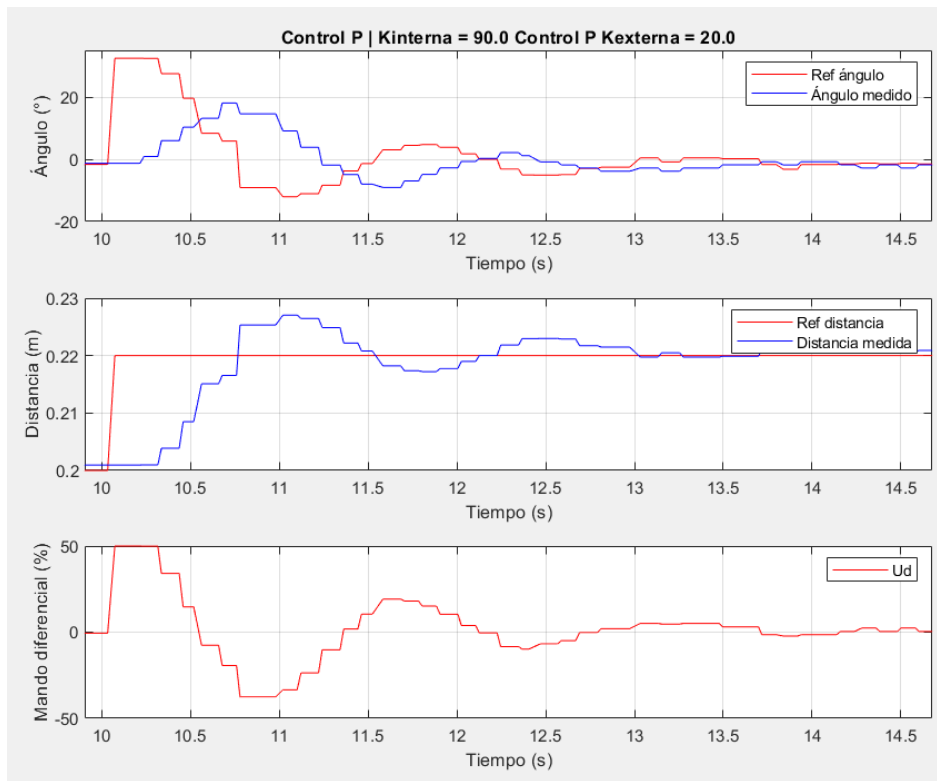


Figura 77. Ensayo utilizado para la identificación de la planta $P_1(s)$ con el LiDAR.

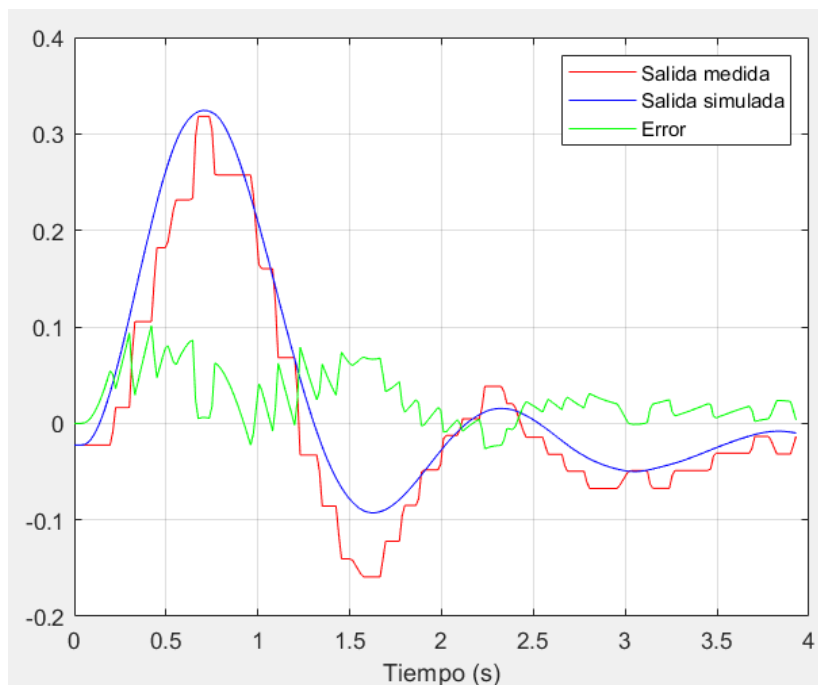


Figura 78. Resultado de identificación de la planta $P_1(s)$ con el LiDAR.

$$P_1(s) = \frac{\theta}{u_d} = \frac{0,023}{(0,2 \cdot s + 1) \cdot s}$$

Ecuación 21. Función de transferencia de la planta $P_1(s)$ con el LiDAR.

La planta $P_2(s)$ no se va a identificar mediante el método de mínimos cuadrados, ya que son todo parámetros que se pueden obtener del punto de operación y de la geometría del robot. La planta resultante es:

$$P_2(s) = \frac{d_n}{\theta} = \frac{0,11 \cdot \left(1 + \frac{0,04}{0,11} s\right)}{s}$$

Ecuación 22. Función de transferencia de la planta $P_2(s)$.

El último detalle que destacar del diseño del sistema con el LiDAR es la presencia de un retardo en la medida de distancia de 0,2 segundos que ha de tenerse en cuenta a la hora de diseñar los controles e incluirlo en la simulación. Se ha decidido incluir esta vez el retardo ya que en este caso sí modificaba de forma significativa el comportamiento de la simulación. De esta forma el diagrama del sistema a controlar se muestra en la Figura 79.

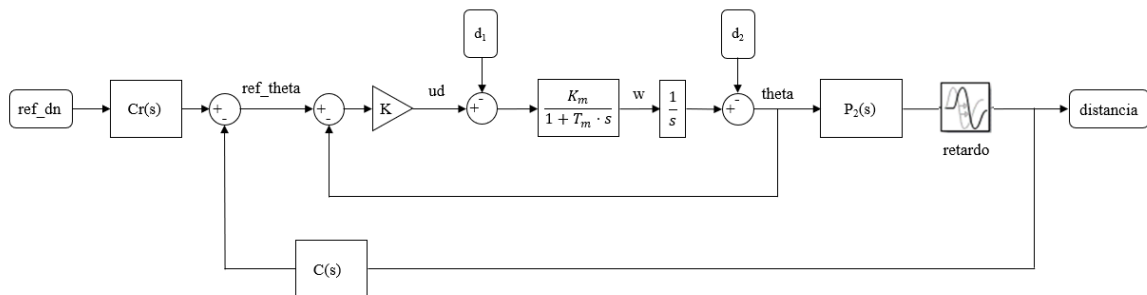


Figura 79. Diagrama de bloques del sistema a controlar con el LiDAR.

Una vez obtenido el modelo de la planta se va a diseñar ahora los controles para ambos lazos de forma teórica, siguiendo los procedimientos descritos en el apartado “Diseño de los controles PID”. Para el lazo interno se obtiene un control proporcional con una ganancia $K = 84,20$ para un margen de fase de 70° .

Para el lazo externo se ha diseñado un control PID con un margen de fase de 60° , una $w_{oPID} = 1,8 \cdot w_{oP}$, un retraso de la parte integral de -10° y un factor de filtrado f de 0,1. Con esas características se obtienen los siguientes parámetros del control: $K = 17,69$, $T_i = 2,70$, $T_d = 0,31$, $N = 8$ y $b = 0,8$.

En la Figura 80 se muestra el gráfico de Black para la función del lazo abierto $G(s)$ y para la planta $P(s)$, en ella se observa que el margen limitante y por el que se ha diseñado es el de fase. Con los parámetros mencionados se obtiene la respuesta a referencia y perturbación (d_2) que se muestra en la Figura 81. La salida tiene un sobrepaso del 7,27%, alcanzando el pico a los 3,18 segundos y con un tiempo de establecimiento al 5% de 4,78 segundos.

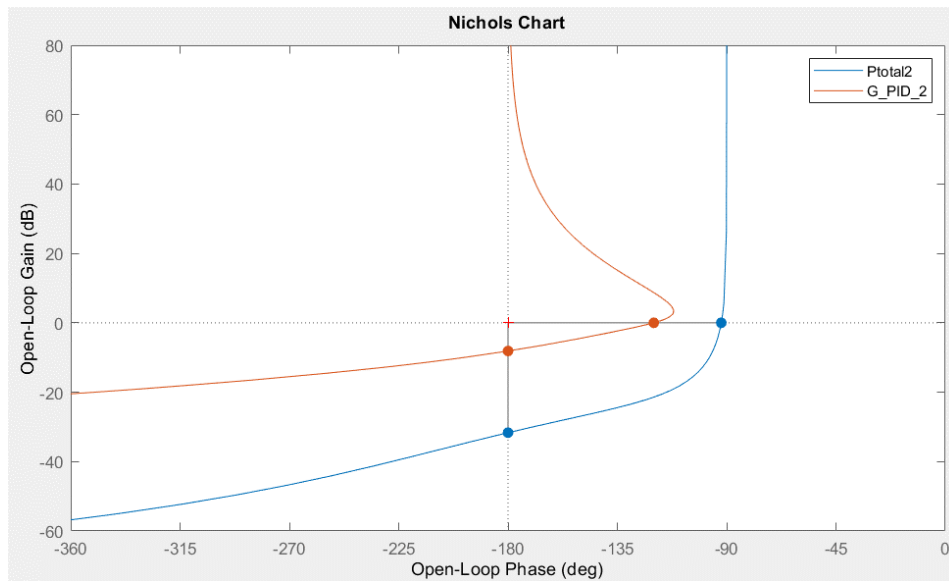


Figura 80. Gráfico de Black para la planta $P(s)$ y el lazo abierto $G(s)$ para el control PID del lazo externo con el LiDAR.

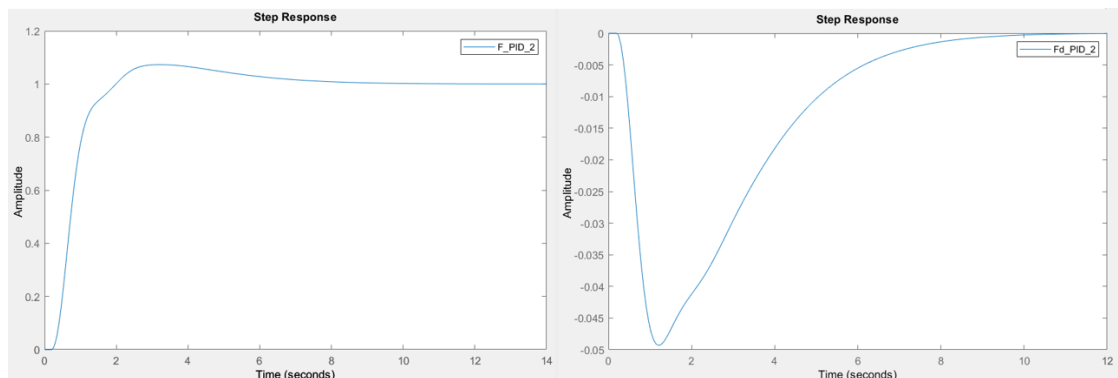


Figura 81. Respuesta a un escalón unitario en referencia y perturbación para el control PID del lazo externo con el LiDAR.

La implementación del control de seguimiento de pared con el LiDAR se realiza de la misma forma que la descrita en el apartado “Implementación del control de seguimiento de pared” con el sensor ToF. La única diferencia es que el *topic* /lidar recibe el mensaje de tipo *geometry_msgs/TwistStamped* con las medidas de distancia y ángulo que son calculadas en el nodo /lidar_node. Este nodo recibe la información del entorno publicada por /ld08_driver, a través del *topic* /scan (ver Figura 82), y realiza los cálculos necesarios para obtener la distancia y el ángulo, según lo indicado en la Ecuación 18.

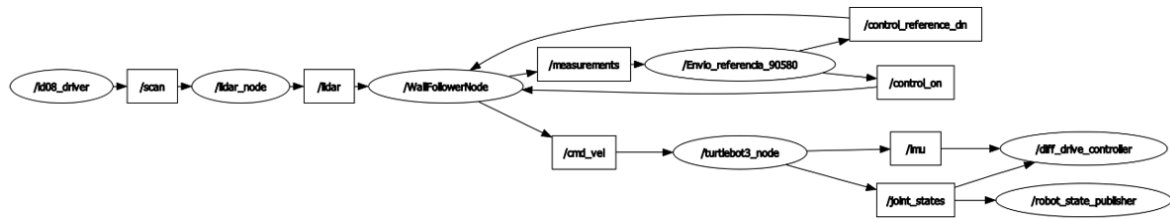


Figura 82. Gráfico de nodos del control de seguimiento de pared con el LiDAR.

En la Figura 83 se muestra la comparación entre ensayo y simulación para el control PID diseñado, con los parámetros que se indican en la figura. En la medida de distancia se observa que la simulación parece tener un sobrepaso mayor, sin embargo, puede deberse al ruido de la medida. El tiempo de pico es difícil de determinar con exactitud en el ensayo, mientras que en simulación se produce a los 3,16 segundos. En ambos casos sí se aprecia que alcanza la referencia en régimen permanente. Además, también se puede comprobar cómo está presente el retardo mencionado anteriormente en la medida de distancia.

Con respecto al ángulo se observa que también se alcanza la referencia en régimen permanente. Sin embargo, en el ensayo este valor no es cero como en simulación, sino que está cerca de los 5° . Esto se debe probablemente a que las medidas tomadas por el sensor no son perfectas y se ven afectadas por el ruido presente.

Por último, con respecto al mando diferencial se puede comprobar que, en régimen permanente (una vez que la distancia medida alcanza la referencia), es cero tanto en simulación como en ensayo. Además, alcanza su valor máximo en 23,84% para simulación y 27,56% para ensayo.

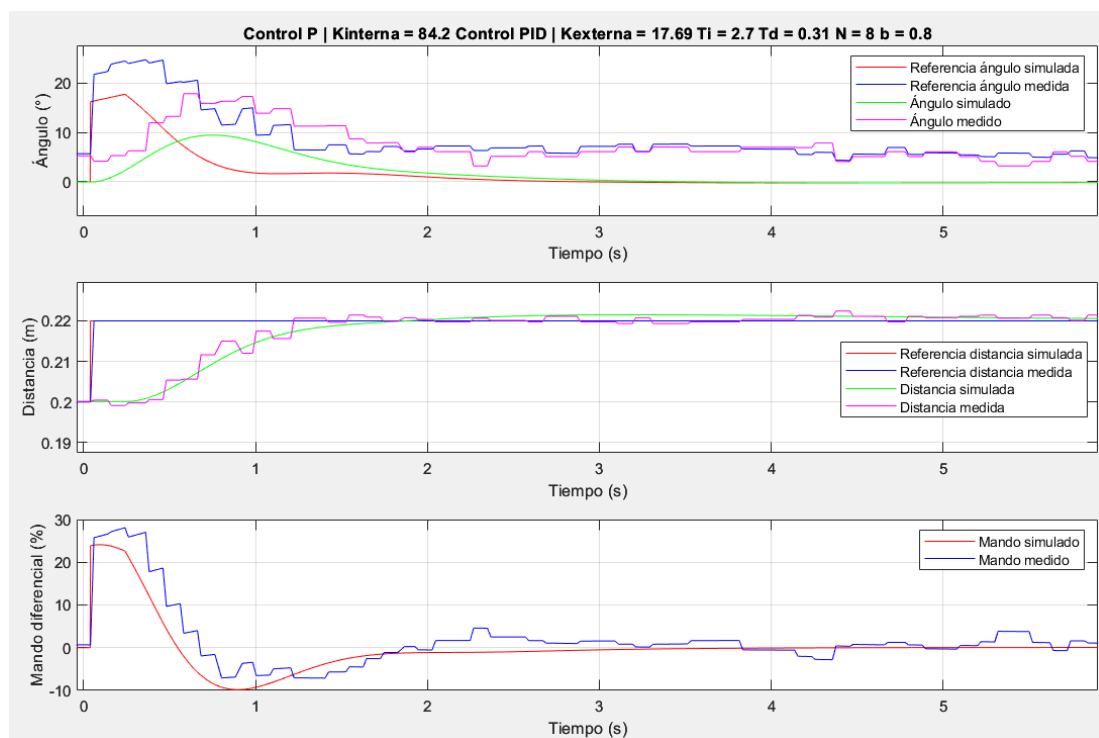


Figura 83. Comparación entre ensayo y simulación para el control PID con el LiDAR.

Capítulo 6. CONCLUSIONES Y TRABAJOS FUTUROS

6.1 CONCLUSIONES

Este trabajo fin de máster tiene como objetivo general la adquisición de experiencia y dominio en el uso del TurtleBot3 Burger para prácticas de control y robótica móvil. Para ello se han desarrollado e implementado un control de velocidad de avance y un control de seguimiento de pared. La metodología seguida ha consistido, en primer lugar, en el modelado de las plantas de ambos controles mediante una identificación por mínimos cuadrados. Posteriormente, se han diseñado los controles de forma teórica, mediante técnicas de respuesta en frecuencia. Una vez obtenidos los parámetros de los controles se ha realizado una comparación entre ensayo y simulación. Para ello se han implementado los controles en el Turtlebot utilizando una combinación de dos de las formas de conexión posibles entre el entorno de MATLAB/Simulink y el robot. Por un lado, se ha utilizado el método de generación de nodos de ROS para desplegar el control PID completo en el robot. Por otro lado, se ha usado el método de conexión con ROS para el envío de parámetros y consignas, así como para recoger los resultados de los ensayos para su posterior monitorización. A continuación, se presentan las principales conclusiones de cada control.

6.1.1 CONTROL DE VELOCIDAD DE AVANCE

Una de las características importantes del modelado de la planta del control de velocidad es la presencia de un retardo de 120ms, causado por los retrasos en las comunicaciones entre los distintos elementos. La elección de los controles estaba entre un PI o un PID, ya que se requería de un control con parte integral para poder eliminar el error en régimen permanente tanto a seguimiento de referencia como rechazo a perturbación, causada esta última por el par de fricción debido a la rampa del circuito. Finalmente se ha escogido un control PI ya que, debido al retardo, la rapidez obtenida con el PID no era demasiado significativa con respecto al PI. Los parámetros del control escogido son: $K = 8,0$, $T_i = 0,09$ y $b = 1,5$.

6.1.2 CONTROL DE SEGUIMIENTO DE PARED

Uno de los problemas encontrados a la hora de desarrollar el control de seguimiento de pared fue la presencia de picos en las medidas que tomaba el LiDAR, causados por los brillos del material de la pared del circuito del laboratorio. Mientras se buscaba una solución a este problema se decidió continuar el proyecto utilizando otro método de medida de distancia y ángulo, un sensor VL6180X ToF.

Para la implementación se ha utilizado un control en cascada con dos lazos. El lazo externo tiene la distancia a la pared como referencia y como salida la consigna de ángulo para el lazo interno. Este último intenta conseguir dicho ángulo aplicando la tensión diferencial necesaria a los motores. En este control también se ha detectado la presencia de un retardo de 0,2 segundos. Sin embargo, se ha decidido no incluirlo en el modelado de los controles ya que su presencia no mejoraba de forma significativa el comportamiento de la simulación frente al ensayo.

Finalmente se ha escogido un control proporcional con una ganancia de 83,33 para el lazo interno, ya que con él se conseguía error nulo en seguimiento a referencia gracias a la integración de la planta y el error de rechazo a perturbación era pequeño y podía encargarse de ello el lazo externo. Para este último se ha escogido un control PID con los siguientes parámetros: $K = 31,18$, $T_i = 0,99$, $T_d = 0,15$, $b = 0,9$ y $N = 7$. Este control permite obtener un error nulo en seguimiento a referencia, además de conseguir mayor rapidez que con controles P y PI. Por último, en la respuesta a la perturbación causada por la curva del circuito, este control tiene un pico menor gracias a la parte diferencial, que compensa más rápidamente la perturbación y hace que el robot se acerque lo menos posible a la pared durante la curva.

Una de las características a destacar de la planta del lazo externo del control es que depende del punto de operación, en concreto de la velocidad v_0 . Para analizar más en profundidad esto se diseñaron dos controles PID, uno para una velocidad de 0,1 m/s y otro para 0,15 m/s y se realizaron ensayos intercambiándolos. Los resultados obtenidos indican que la elección de la velocidad del punto de trabajo sí afecta al comportamiento del robot en los ensayos. Es más significativo si se trabaja con una velocidad superior a la usada para diseñar.

Por último, una vez que se solucionó el problema de las medidas en el circuito con el LiDAR se estudió el comportamiento del robot con el mismo. La planta identificada tenía la misma constante de tiempo que la modelada con el sensor ToF. Sin embargo, presentaba una ganancia estática menor. En este caso sí se ha incluido el retardo de 0,2 segundos existente, causado por los retrasos en las comunicaciones entre los elementos, ya que contribuía de forma significativa a que simulación y ensayo fuesen más parecidos. El cambio en la ganancia estática de la planta, la diferencia en la geometría del robot a la hora de obtener las medidas de distancia y ángulo y el retardo hacen que ambos sensores no sean intercambiables para el mismo control, sino que es necesario diseñar controles específicos para cada uno de ellos.

Con todo esto se puede concluir que se han alcanzado todos los objetivos del proyecto, ya que se han diseñado e implementado los controles de velocidad y seguimiento de pared, identificando el modelo del robot en ambos casos.

6.2 TRABAJOS FUTUROS

Tras la realización de este proyecto se han identificado tres posibles puntos que se podrían explorar como trabajos futuros:

- **Posición del LiDAR.** Actualmente este sensor se encuentra situado en el medio del robot, pero podría estudiarse la posibilidad de adelantarlo para ver si se pueden obtener mejores respuestas, en especial a la hora de tratar con la perturbación causada por la curva del circuito.
- **Elección de las medidas del LiDAR.** En línea con lo mencionado en el punto anterior se podría estudiar también la elección de distintos valores del ángulo que separa las medidas que se usan en el LiDAR. En este proyecto se ha utilizado un ángulo de 14° , pero se podría tratar de aumentar el ángulo para adelantar dicha medida y estudiar cómo afecta al comportamiento del robot en los ensayos.

- Por último, como se ha mencionado anteriormente la planta del lazo externo del control de seguimiento de pared depende de la velocidad del punto de operación. Por ello como futuro desarrollo se podría estudiar el uso de técnicas de diseño de controles más avanzadas, como puede ser el control adaptativo.

Capítulo 7. BIBLIOGRAFÍA

- [1] F. Barragán Castro, «Implementación de algoritmos de navegación en un TurtleBot3,» Madrid, 2022.
- [2] Robotis, «ROBOTIS e-Manual,» [En línea]. Available: <https://emanual.robotis.com/docs/en/platform/turtlebot3/overview/#overview>. [Último acceso: 11 Julio 2023].
- [3] MathWorks, «Get Started with ROS,» [En línea]. Available: <https://www.mathworks.com/help/ros/ug/get-started-with-ros.html>. [Último acceso: 11 Julio 2023].
- [4] MathWorks, «Exchange Data with ROS Publishers and Subscribers,» [En línea]. Available: <https://www.mathworks.com/help/ros/ug/exchange-data-with-ros-publishers-and-subscribers.html>. [Último acceso: 29 Junio 2023].
- [5] MathWorks, «ROS 2 Node Generation and Deployment,» [En línea]. Available: <https://es.mathworks.com/help/ros/ros2-node-generation-and-deployment.html>. [Último acceso: 26 Junio 2023].
- [6] «ROS: Why ROS?,» [En línea]. Available: <https://www.ros.org/blog/why-ros/>. [Último acceso: 28 Octubre 2022].
- [7] Quanser, «QBot 3,» 2022. [En línea]. Available: <https://www.quanser.com/products/qbot-3/>. [Último acceso: 12 Octubre 2022].
- [8] Waveshare, «AlphaBot2-Pi,» [En línea]. Available: <https://www.waveshare.com/wiki/AlphaBot2-Pi#AlphaBot2-Base>. [Último acceso: 12 Octubre 2022].
- [9] ROBOTIS e-Manual, «TurtleBot3,» [En línea]. Available: <https://emanual.robotis.com/docs/en/platform/turtlebot3/features/#features>. [Último acceso: 12 Octubre 2022].
- [10] Thymio, «Thymio,» [En línea]. Available: <https://www.thymio.org/products/extend-learning-with-thymio/thymio/>. [Último acceso: 12 Octubre 2022].
- [11] Agilex, «LIMO,» [En línea]. Available: <https://www.agilex.ai/product/12?lang=en-us>. [Último acceso: 12 Octubre 2022].
- [12] Bizbot Technology, «Bveeta Mini Type R007,» [En línea]. Available: <https://bzonemy101.wixsite.com/mysite>. [Último acceso: 26 Octubre 2022].

- [13] Lab. Regulación Automática, «Proyecto 2 – Control PID diseñado por respuesta en frecuencia para navegación mediante seguimiento de pared,» 2022.
- [14] Robotis, «ROBOTIS e-Manual,» [En línea]. Available: <https://emanual.robotis.com/docs/en/platform/turtlebot3/overview/#overview>. [Último acceso: 20 Junio 2023].
- [15] Robotis, «OpenCR1.0 ROBOTIS e-Manual,» [En línea]. Available: https://emanual.robotis.com/docs/en/platform/turtlebot3/appendix_opencr1_0/. [Último acceso: 20 Junio 2023].
- [16] Robotis, «ROBOTIS e-Manual OpenCR,» [En línea]. Available: <https://emanual.robotis.com/docs/en/parts/controller/opencr10/>. [Último acceso: 20 Junio 2023].
- [17] Raspberry Pi, «Raspberry Pi 4 Tech Specs,» [En línea]. Available: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/>. [Último acceso: 1 Julio 2023].
- [18] Robotis, «XL430-W250-T ROBOTIS e-manual,» [En línea]. Available: <https://emanual.robotis.com/docs/en/dxl/x/xl430-w250/>. [Último acceso: 24 Junio 2023].
- [19] Robotis, «LDS-02 ROBOTIS e-Manual,» [En línea]. Available: https://emanual.robotis.com/docs/en/platform/turtlebot3/appendix_lds_02/. [Último acceso: 24 Junio 2023].
- [20] Adafruit, «VL6180X Time of Flight Distance Ranging Sensor,» [En línea]. Available: <https://www.adafruit.com/product/3316>. [Último acceso: 25 Junio 2023].
- [21] Texas Instruments, «TCA9548A,» [En línea]. Available: https://www.ti.com/product/TCA9548A?utm_source=google&utm_medium=cpc&utm_campaign=asc-null-null-gpn_en-cpc-pf-google-ww&utm_content=tca9548a&ds_k=TCA9548A+Datasheet&dcm=yes&gclid=Cj0KCQjwy9-kBhCHARIsAHpBjHg0q5PVLMwhlaeX3mBqk2mwLTPHxEw5xsFEuKTQDOBzb_gEdYX. [Último acceso: 25 Junio 2023].
- [22] ROS, «ROS 2 Documentation,» [En línea]. Available: <https://docs.ros.org/en/foxy/index.html>. [Último acceso: 25 Junio 2023].
- [23] ROS, «ROS2 Documentation: Concepts,» [En línea]. Available: <https://docs.ros.org/en/foxy/Concepts.html>. [Último acceso: 25 Junio 2023].
- [24] ROS, «ROS2 Documentation: Understanding nodes,» [En línea]. Available: <https://docs.ros.org/en/foxy/Tutorials/Beginner-CLI-Tools/Understanding-ROS2-Topics/Understanding-ROS2-Topics.html>. [Último acceso: 25 Junio 2023].

- [25] MathWorks, «ROS Toolbox,» [En línea]. Available: <https://es.mathworks.com/help/ros/>. [Último acceso: 26 Junio 2023].
- [26] MathWorks, «ROS Log Files (rosbags),» [En línea]. Available: <https://es.mathworks.com/help/ros/ug/ros-log-files-rosbags.html>. [Último acceso: 26 Junio 2023].
- [27] MathWorks, «Get Started with ROS,» [En línea]. Available: <https://www.mathworks.com/help/ros/ug/get-started-with-ros.html>. [Último acceso: 29 Junio 2023].
- [28] MathWorks, «Get Started with ROS in Simulink,» [En línea]. Available: <https://www.mathworks.com/help/ros/ug/get-started-with-ros-in-simulink.html>. [Último acceso: 29 Junio 2023].
- [29] F. L. Pagola, Regulación automática, 2006.
- [30] Lab. Regulación Automática , «Proyecto 2 - Control PID diseñado por respuesta en frecuencia para navegación mediante seguimiento de pared,» 2022.
- [31] ONU, «Objetivos de Desarrollo Sostenible,» 2015. [En línea]. Available: <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>. [Último acceso: 11 Octubre 2022].
- [32] GLOBE NEWSWIRE, «Global Mobile Robotics Market to Generate USD 77.98 Billion by 2030, Outlines a New Report by Next Move Strategy Consulting,» 2022. [En línea]. Available: <https://www.globenewswire.com/en/news-release/2022/08/22/2502286/0/en/Global-Mobile-Robotics-Market-to-Generate-USD-77-98-Billion-by-2030-Outlines-a-New-Report-by-Next-Move-Strategy-Consulting.html>. [Último acceso: 26 Octubre 2022].
- [33] PwC, «Claves e inversiones estratégicas para una España 5.0,» 2021.

ANEXO I.

ALINEACIÓN CON LOS OBJETIVOS DE DESARROLLO

SOSTENIBLE

Los objetivos de desarrollo sostenible (ODS) consisten en un total de 17 objetivos presentados en 2015 por la Organización de las Naciones Unidas como parte de la Agenda 2030, para un desarrollo sostenible en el ámbito económico, social y ambiental. A estos 17 objetivos de desarrollo sostenible están ligados 169 metas, ambos de carácter integrado e indivisible [31].

Este proyecto fin de máster está principalmente relacionado con el Objetivo 4: Garantizar una educación inclusiva, equitativa y de calidad y promover oportunidades de aprendizaje durante toda la vida para todos [31]. Este objetivo tiene diez metas específicas, de las cuales las más aplicables a este proyecto son:

- La meta 4.3 De aquí a 2030, asegurar el acceso igualitario de todos los hombres y las mujeres a una formación técnica, profesional y superior de calidad, incluida la enseñanza universitaria.
- La meta 4.4 De aquí a 2030, aumentar considerablemente el número de jóvenes y adultos que tienen las competencias necesarias, en particular técnicas y profesionales, para acceder al empleo, el trabajo decente y el emprendimiento.

Con este proyecto se pretende diseñar un robot móvil para ser utilizado en ambientes educativos universitarios, con ello se contribuiría a dichas metas proporcionando una herramienta que ayude a los estudiantes a alcanzar las competencias necesarias para una educación de calidad.

Por otro lado, este trabajo también permite contribuir al Objetivo 9: Construir infraestructuras resilientes, promover la industrialización sostenible y fomentar la innovación [31]. Ya que con él se pretende diseñar los controles del robot para que pueda circular adecuadamente, ayudando a formar a nuevas generaciones de profesionales que contribuyan con sus conocimientos a la innovación y construcción de infraestructuras sostenibles y resilientes.

Contribuir a la consecución de estos objetivos es de gran importancia si se tiene en cuenta el impacto que va a tener este sector en el futuro. Según indica [32], en 2021 el mercado global de la robótica móvil generó 13'56 mil millones de dólares, esta cifra se espera que aumente hasta los 77'98 mil millones para 2030. Por otro lado, PwC [33] realizó un informe en el que estima que este cambio en la industria requerirá que se fomente la capacitación de nuevos

profesionales y poder atraer talento, se estima que esto podría implicar la creación de 200.000 nuevos puestos de trabajo para 2030.