
ÍNDICE GENERAL

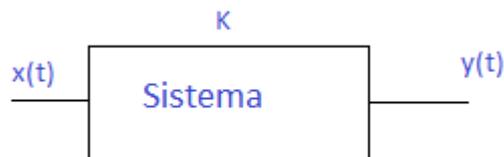
1 INTRODUCCIÓN.....	2
1.1 SISTEMAS EN LAZO ABIERTO Y EN LAZO CERRADO.....	2
1.2 FUNCIÓN DE TRANSFERENCIA.....	3
1.3 ANÁLISIS DEL MOTOR.....	4
2 ESTUDIO DEL CONTROLADOR PROPORCIONAL.....	8
2.1 ESTUDIO TEÓRICO.....	8
2.2 ESTUDIO PRÁCTICO.....	9
2.3 LUGAR DE RAÍCES.....	11
3 ESTUDIO DEL CONTROLADOR PROPORCIONAL-INTEGRAL.....	15
3.1 ESTUDIO TEÓRICO.....	15
3.2 ESTUDIO PRÁCTICO.....	17
3.3 LUGAR DE RAÍCES.....	19
4 ESTUDIO DEL CONTROLADOR PID.....	21
4.1 ESTUDIO TEÓRICO.....	21
4.1.1 <i>Estudio básico</i>	21
4.1.2 <i>Estudio de las perturbaciones</i>	23
4.2 SIMULACIONES CON MATLAB.....	24
4.3 ESTUDIO PRÁCTICO.....	27
4.4 LUGAR DE RAÍCES.....	29
4.5 CALCULADORA ANALÓGICA.....	30
4.5.1 <i>Saturación</i>	30
4.5.2 <i>Offset</i>	31
5 BIBLIOGRAFÍA.....	32
6 ANEXO: CÓDIGO DE MATLAB.....	33

1 Introducción

1.1 Sistemas en lazo abierto y en lazo cerrado

Para comenzar nuestro estudio, hemos de saber que existen dos tipos de sistemas: Sistemas en lazo abierto y sistemas en lazo cerrado. A continuación detallaremos brevemente las diferencias entre ambos.

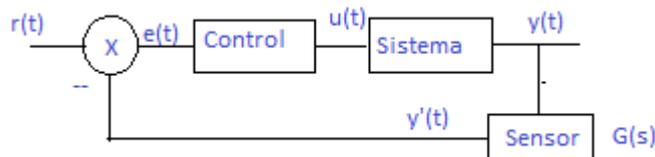
Un sistema en lazo abierto presenta la siguiente estructura:



Siendo $x(t)$ la entrada dada e $y(t)$ la salida deseada. Por lo tanto la función de transferencia de este sistema será $k = \frac{y(t)}{x(t)}$. Si realizamos esta operación en el dominio de la frecuencia mediante el uso de la transformada de Laplace, obtendremos como resultado que $G(s) = \frac{Y(s)}{X(s)}$ de donde obtendremos que la entrada es $X(s) = \frac{Y(s)}{G(s)}$. Realizando la transformada inversa de Laplace obtendremos la función $x(t)$ o la función k según nos convenga.

Cabe resaltar que este tipo de sistemas son inestables por lo que no los desarrollaremos durante nuestro estudio

Un sistema en lazo cerrado (solución retroalimentada) presenta la siguiente estructura:



En nuestros casos idealizaremos el sensor suponiendo que su $G(s) = 1$, lo cual nos va a simplificar mucho los cálculos de esta forma.:

$$e(t) = r(t) - y(t)$$

$$e'(t) = r'(t) - y'(t) . \text{ Con lo que si } e' \text{ tiende a cero podemos afirmar que } y'(t) \text{ tiende a } r'(t).$$

El controlador es un sistema a diseñar, en el cual buscaremos cumplir una serie de especificaciones dadas por unos parámetros tales como el tiempo de subida o el tiempo de establecimiento. En todo controlador intentaremos que el error tienda a cero.

Si denominamos G_c a la ganancia del controlador y G a la ganancia del sistema, podemos establecer la función de transferencia del sistema total, en el dominio de Laplace de la siguiente manera:

$$H(s) = \frac{Y(s)}{R(s)} = \frac{G * G_c}{1 + G * G_c}$$

1.2 Función de transferencia

Ya hemos hablado brevemente de la función de transferencia en el apartado anterior, pero ahora daremos una definición más formal. Llamaremos función de transferencia a la relación existente entre la salida y la entrada de un sistema en el dominio de Laplace tomando siempre condiciones iniciales nulas. Este último aspecto es el realmente importante dentro de nuestra definición.

Dentro de este apartado aparecen ciertos conceptos que serán necesarios para explicar el resto de nuestro trabajo.

Un cero es una entrada que no genera salida. Hablando en términos matemáticos un cero es un valor que anula el numerador.

Un polo es una salida con entrada nula. Implica condiciones iniciales no nulas por lo que dejaría de existir una función de transferencia, ya que como hemos indicado solo se define para condiciones iniciales nulas. En términos matemáticos un polo es un valor que anula el denominador.

Para calcular los polos de la función de transferencia igualamos el denominador a cero. Como nuestro trabajo se va a realizar como máximo con ecuaciones de segundo grado, obtenemos como máximo dos polos. Definiendo el denominador como $s^2 + a*s + b$ e igualando a cero, obtenemos que podemos tener 3 posibles casos.

Si $a^2 > 4*b$ entonces tenemos dos polos reales y distintos

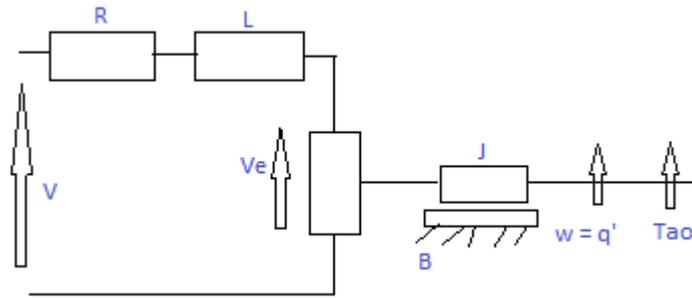
Si $a^2 = 4*b$ entonces tenemos un único polo real doble

Si $a^2 < 4*b$ entonces tenemos dos polos complejos conjugados. La parte real nos dará la amortiguación y la parte imaginaria la oscilación

Controlabilidad: A veces nos aparecerá el problema de que el sistema no puede ser controlado. Este problema suele ir asociado a la cancelación de polos. Si esto sucede habría que revisar el modelo, ya que la cancelación significa que un sistema de orden superior puede expresarse como uno de menor orden cosa que no es cierta. Las cancelaciones únicamente son válidas si no se cancelan los polos inestables.

Estabilidad: Diremos que nuestro sistema es estable cuando para una entrada acotada obtengamos una salida acotada.

1.3 Análisis del motor



Arriba hemos representado gráficamente nuestro motor. A continuación definiremos los parámetros:

V= Voltaje inicial con el que se alimenta el motor

R= Resistencia que incluye la bobina al tratarse de una bobina no ideal

L= Bobina

Ve= Fuerza electromotriz en sentido inverso al de la alimentación. Puede expresarse como $Ke * q'$

J= Momento de inercia

B= Rozamiento

Tao = Par mecánico. Puede expresarse de la forma $Km * i$

q= Posición del motor.

Wm= velocidad del motor. Puede definirse también como la derivada de la posición respecto al tiempo.

Taof= Par de fricción

En este punto es importante resaltar, que esta es una visión muy simplificada de nuestro motor.

A continuación planteamos el sistema de ecuaciones del cual saldrá nuestra función de transferencia:

$$\text{Ec. eléctrica: } V = R * i + L * \left(\frac{di}{dt}\right) + Ve$$

$$\text{Ec. mecánica: } Tao = J * q'' + B * q' + Taof$$

$$\text{Ec. que relaciona las dos anteriores: } Tao = Km * i$$

Si sustituimos:

$$V = \frac{R}{Km} * (J * q'' + B * q' + Taof) + \left(\frac{L}{Km}\right) * \left(\frac{d}{dt}\right) * (J * q'' + B * q' + Taof) + Ve$$

$$V = \frac{L * J}{Km} * q'''' + \left(\frac{R * J + L * B}{Km * q''}\right) + \frac{R * B}{Km * q'} + R * Taof + \left(\frac{L}{Km}\right) * \left(\frac{d}{dt}\right) * (Taof)$$

En este primer modelo será una simplificación ya que tomaremos que no existe el par mecánico. De él, hemos obtenido una ecuación de tercer orden respecto a la posición, que se convierte en una de segundo orden respecto a la velocidad.

Si calculamos la función de transferencia de la función anterior obtenemos que para la posición

$$G(s) = \frac{Km}{s[(R + L * s) * (B + J * s) + Ke * Km]}$$

Ahora bien, sabemos que la ganancia en velocidad es $Gv(s) = s * G(s)$. De aquí obtenemos que la ganancia en velocidad es $Gv = H(s) / [1 + H(s)]$ con $H(s) = Km * Ke / [(R + L * s) * (B + J * s)]$. De aquí se deduce fácilmente por el orden del denominador que la función únicamente tiene dos polos, uno

relacionado con la parte eléctrica (primer término) y otro relacionado con la parte mecánica (segundo término).

En este punto de la explicación haremos un breve paréntesis para introducir un pequeño desarrollo matemático. Si tenemos una función de transferencia de segundo orden de la forma $G(S) = \frac{K}{s^2} + a*s + b$ para cualquier k esta expresión se puede descomponer en fracciones simples como $\frac{\alpha 1}{s - p1} + \frac{\alpha 2}{s - p2}$, de donde las expresiones resultantes son fácilmente reconocibles para el cálculo de su transformada inversa ya que son simples exponenciales en las cuales p1 y p2 serán sus respectivos coeficientes de amortiguamiento. En esta situación de suma de dos exponenciales amortiguadas aparece el concepto de polo dominante. Denominamos polo dominante a aquel de los dos polos que predomina en la suma total de las dos señales, ya que una de ellas disminuirá muy lentamente respecto a la otra, haciendo que la influencia de la otra sea mucho menor.

Una vez aclarado este punto y dados los valores de los diferentes elementos, podemos observar que el polo dominante pertenece a la parte mecánica ya que $\frac{R}{L} \gg \frac{B}{J}$ con lo que $\frac{R}{L}$ caerá mucho más rápido que el otro término.

Ahora bien, si queremos calcular la salida Y(s) teniendo como entrada el escalón, entonces tenemos $Y(s) = \frac{K}{s+p} * U(s) = \frac{\alpha 1}{s+p} + \frac{\alpha 2}{s}$. Como tenemos condiciones iniciales nulas podemos concluir que $\alpha 1 = -\alpha 2$.

Si realizamos la transformada inversa de Laplace obtendremos que la respuesta al escalón es $y(t) = \alpha 2 * (1 - \exp(-p*t))$. Podemos calcular el valor de $\alpha 2$ de dos formas equivalentes.

La primera sería calcular el límite cuando t tiende a infinito de la expresión anterior con lo que obtendremos que y(t) es igual a $\alpha 2$. Midiendo este resultado directamente del motor obtendremos que $\alpha 2$ son 5.18 V para una entrada a = 5V. La alternativa es aplicar el teorema del valor final que nos dice que el límite anterior es igual al límite de $S*Y(s)$, obteniendo de esta manera el mismo resultado.

Para calcular el valor del polo, hemos utilizado Matlab. Si despejamos p de la expresión anterior

obtenemos que $p = \frac{-\ln(1 - \frac{y(t)}{\alpha 2})}{t}$, tal y como se observa en la grafica anterior. De aquí deducimos que nuestro polo se encuentra en $p = 42.11$.

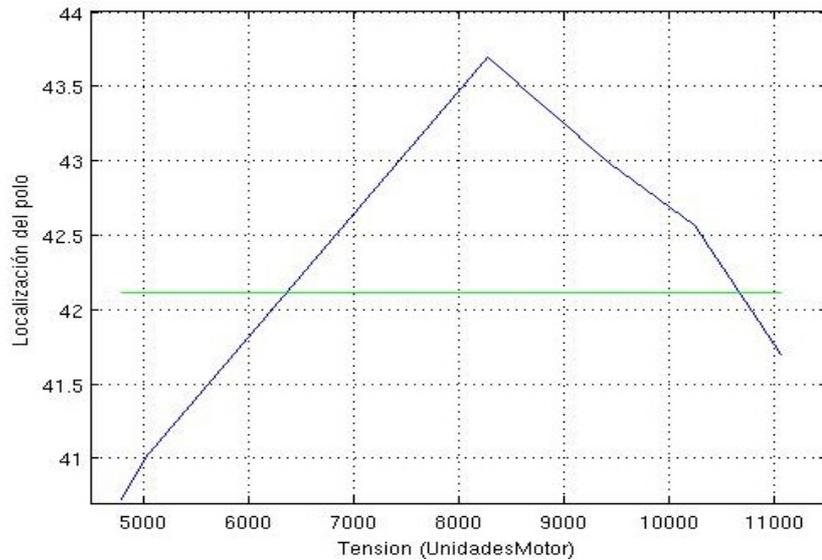


Figura 1: Búsqueda de la posición del polo

En el eje X tenemos la tensión medida en las unidades que nos proporciona el sistema (según los cálculos realizados $1V = 2184.4$) y en el eje Y el valor en el que estará el polo. Con línea azul representamos los valores generales y en verde la media que es la que nos lleva al establecimiento al valor del polo. El programa que realiza esta función se puede observar en los archivos adjuntos.

De aquí podemos obtener el valor de $K = \frac{a2 * p}{a} = 43.64$. Con este resultado obtenemos que la ganancia es $G(0) = \frac{K}{p} = 1.036$, cercano a 1 que es el resultado que buscábamos.

Además para estos valores hemos realizado una simulación de su respuesta al impulso tanto para velocidad como para posición cuyos resultados aquí reflejamos.

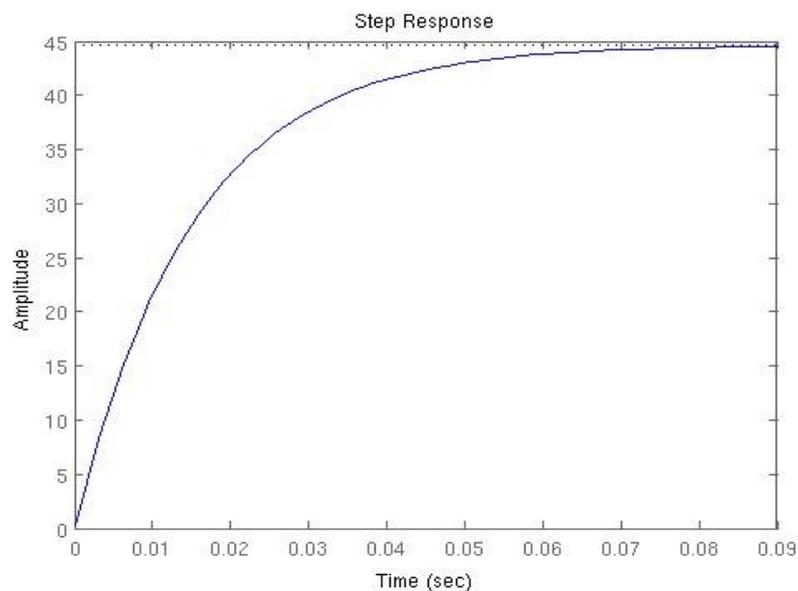


Figura 2: Respuesta al escalón unitario para la función de transferencia de la velocidad

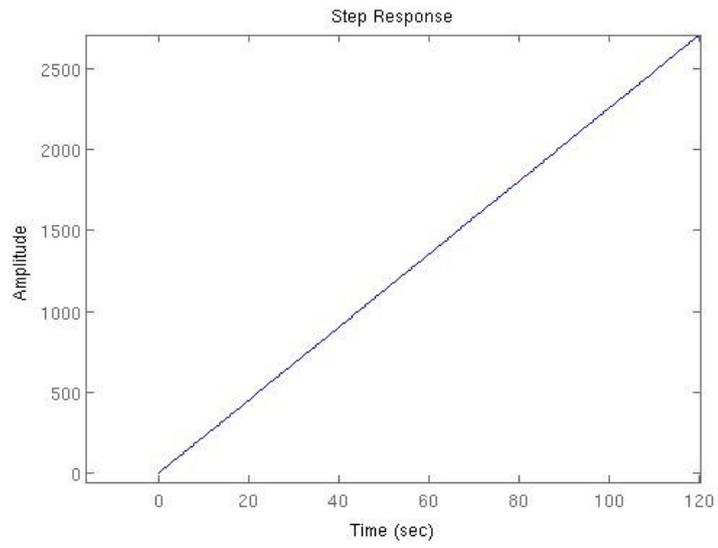


Figura 3: Respuesta al escalón para la función de transferencia para la posición

2 Estudio del controlador proporcional

2.1 Estudio teórico

Si volvemos al esquema en lazo cerrado y denominamos K_p a la ganancia del controlador y $\frac{K}{s*(s+p)}$ a la ganancia del sistema obtenemos que la función de transferencia del sistema es

$$H(s) = \frac{K * K_p}{s(s+p) + K * K_p}$$

Su respuesta al escalón tendrá la misma forma que la que hemos definido en el apartado anterior ya que únicamente estamos multiplicando por una constante.

Como ya ha sido estudiado en asignaturas de circuitos, la respuesta será diferente según el valor de los términos de orden cero y uno del denominador. El término de orden cero es identificado con la pulsación de resonancia, y el término de orden uno es identificado con 2 veces la pulsación de resonancia por el coeficiente de amortiguamiento. Dependiendo del valor del coeficiente la respuesta toma diversa forma:

1. Si el coeficiente es menor que 1 se genera una señal exponencial amortiguada.
2. Si el coeficiente es igual a 1 se genera una señal que es la suma de una constante más una exponencial.
3. Si el coeficiente es mayor que 1 se generan dos señales exponenciales.

La salida de nuestro sistema para la entrada escalón será $Y(s) = H(s) * R(s) = \frac{K * K_p * V_{in}}{(s^2 + s * p + K * K_p) * s}$. Si identificamos términos deducimos que la pulsación de resonancia al cuadrado es $\omega_n^2 = K * K_p$ y que $p = 2 * \epsilon * \omega_n$.

De aquí podemos sacar que el valor del polo al cuadrado es igual a $p^2 = 4 * \epsilon^2 * K * K_p$. Como buscamos el límite para $\epsilon = 1 = \frac{p^2}{2 * \sqrt{K * K_p}}$. Si realizamos los cálculos obtenemos que $K_p = 16.25$.

Además hemos realizado un estudio en Matlab para el cálculo de este parámetro y el resultado obtenido se puede reflejar en la siguiente gráfica

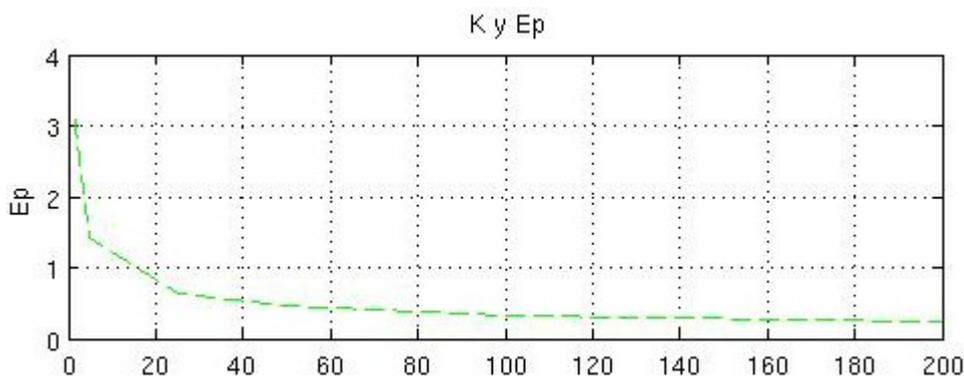


Figura 4: Variación del coeficiente de amortiguamiento en función de la ganancia

Utilizando la función `ginput` del Matlab obtenemos que el valor de K para el cual se cumple que $Ep=1$ es de 16.35, muy aproximado al cálculo teórico que habíamos realizado.

También hemos calculado la elongación máxima a la que hemos denominado M_p y que se define como la exponencial de $\frac{\pi * eps}{\sqrt{1-eps}}$. Teóricamente hemos obtenido el resultado siguiente, realizando la simulación en Matlab.

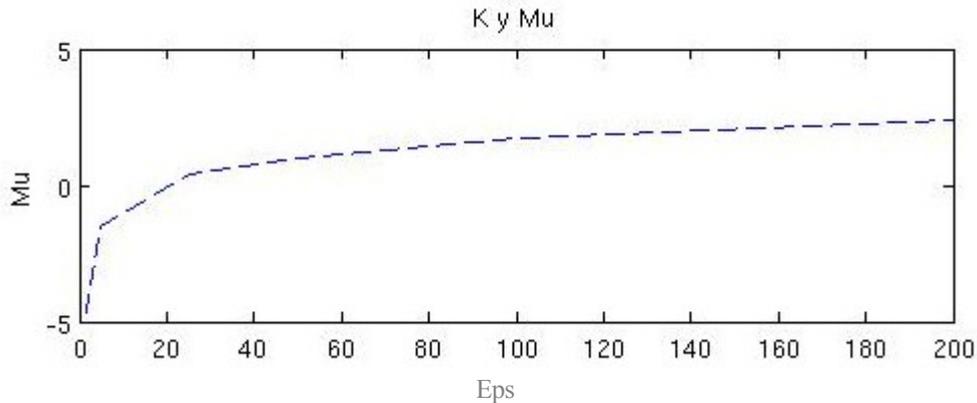


Figura 5: Variación del coeficiente de amortiguamiento máximo

El que los primeros valores sean negativos, es entendible ya que es para los momentos en los que la señal no se amortigua. Realmente no son valores negativos reales, sino imaginarios.

2.2 Estudio práctico

A continuación, reflejaremos una serie de resultados obtenidos con las medidas del motor y unos sencillos programas que hemos programado y simulado en Matlab. En ellos ofrecemos la variación de la posición, la velocidad, y el error variando el parámetro K .

Para ello vamos a diferenciar dos casos, la entrada a un escalón constante y la entrada a una rampa, variando la constante K o variando la pendiente de la rampa. Cabe destacar que en lo que se dibujará, la entrada no será exactamente una rampa sino la entrada que capta el motor como entrada.

Para el escalón podremos observar que a partir de $K=15$ se empieza a observar una pequeña oscilación correspondiente al punto que habíamos predicho. Esto se debe, al amortiguamiento subcrítico que se daba a partir de $K=16$. Pasado un pequeño tiempo en el cual la función de salida se encuentra en régimen transitorio, se establece el régimen permanente.

A continuación, se pueden ver unas cuantas gráficas resultado de los experimentos realizados en las que todas ellas están hechas sobre un controlador de velocidad en lo cual variamos la K . El resultado obtenido es el siguiente. Algunas de ellas estarán presentando una saturación a partir de la cual, no nos es interesante el estudio en esa parte.

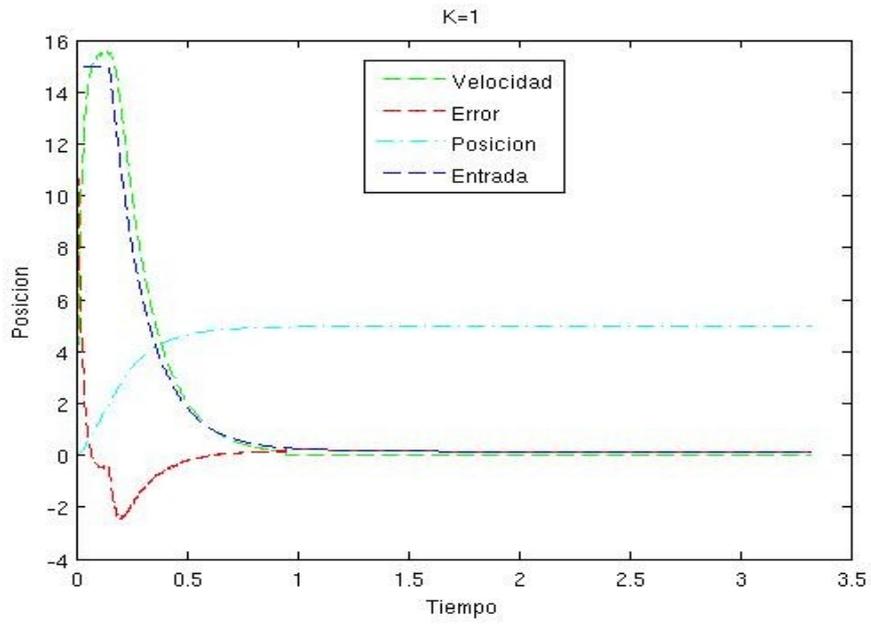


Figura 6: Simulación del motor para un controlador proporcional con $K=1$.

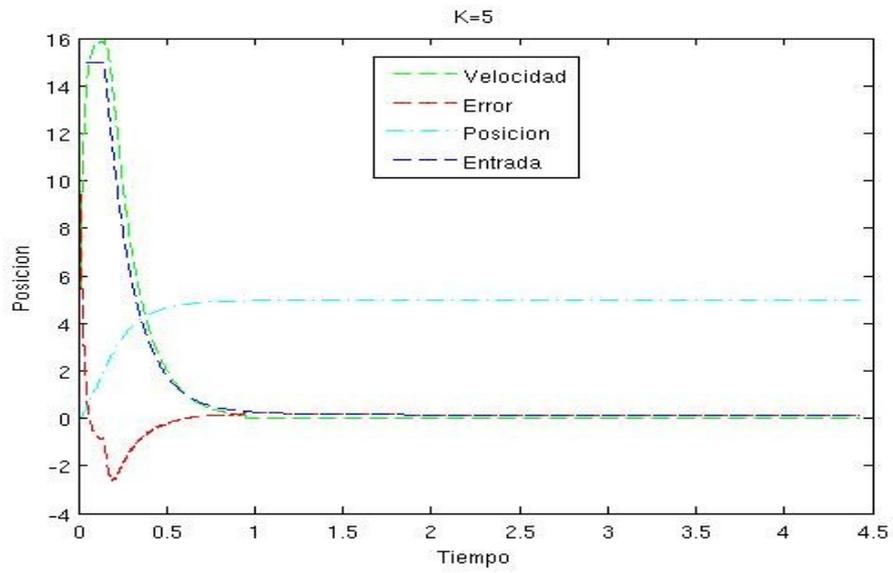


Figura 7: Simulación del motor para un controlador proporcional con $K=5$.

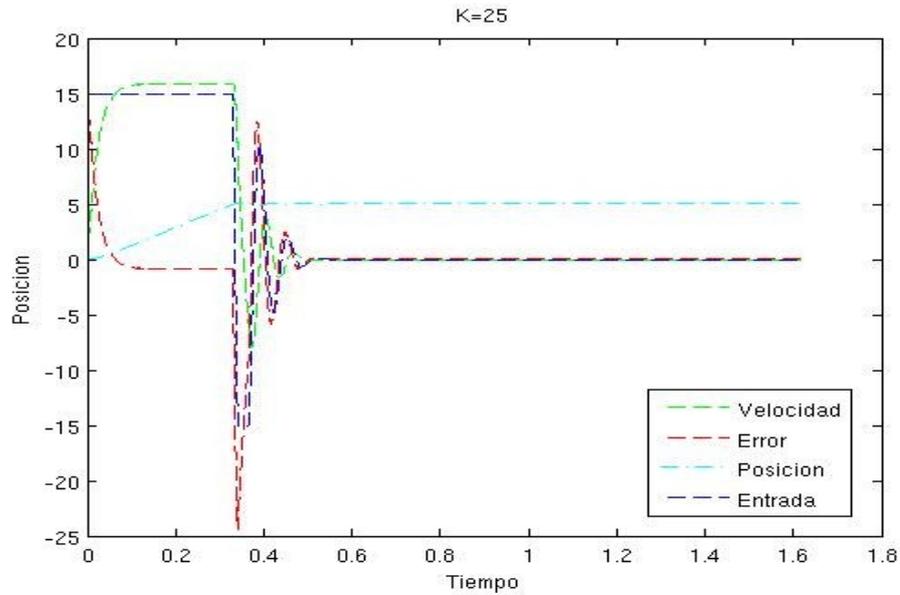


Figura 8: Simulación de un controlador proporcional con K=25.

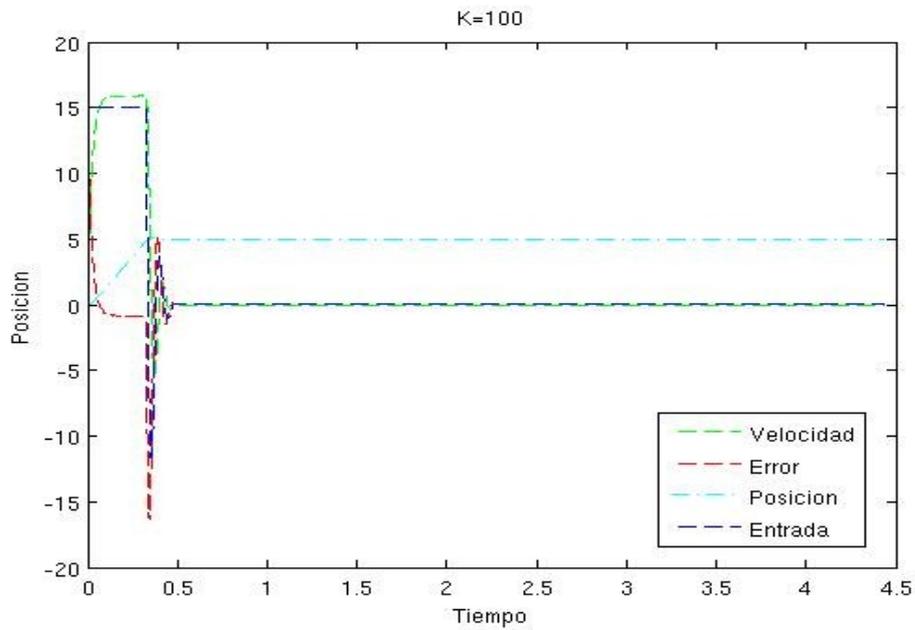


Figura 9: Simulación de un controlador proporcional con K=100.

2.3 Lugar de raíces

Ahora estudiaremos la variación de la posición de los polos según el método del lugar de raíces. La idea básica es que los valores de s que hacen la función de transferencia alrededor del lazo sea igual a 1 deben satisfacer la ecuación característica del sistema. Sea la función de transferencia

$$H(s) = \frac{Y(s)}{X(s)} = \frac{G(s)}{1 + G(s) * G_c(s)}$$

La ecuación característica para este sistema en lazo cerrado se obtiene igualando el denominador a cero. Luego $G(s) * G_c(s) = -1$. Aquí habría que satisfacer la condición de ángulo y la condición de magnitud, pero el lugar de raíces solo nos une los puntos que cumplen la condición de ángulo. Dicha condición es que la fase de $G(s) * G_c(s)$ es igual a un múltiplo impar de veces de 180° tanto en sentido horario como en sentido antihorario. Lo primero será localizar los polos en el eje real en lazo abierto. Además sabemos que el número de lugares de raíces. Ahora para

calcular las asíntotas tomamos un punto muy lejano y realizamos el límite. Luego el ángulo será $\frac{180 * (2k + 1)}{N^{\circ} \text{ de polos}}$.

Como ya hemos ido viendo, sabemos que nuestra función de transferencia para la velocidad en el modelo simplificado en lazo abierto es $H(s) = \frac{K_m}{s + p}$ en lazo abierto. Además sabemos que la función de transferencia para la posición es $G(s) = \frac{H(s)}{s}$. Aplicando este desarrollo y la función rlocus de Matlab hemos obtenido estos resultados para el modelo simplificado.

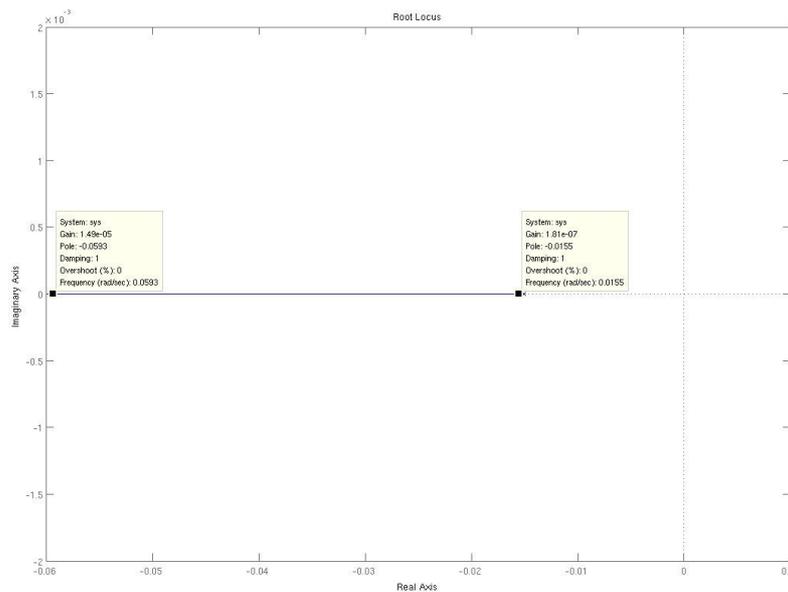


Figura 10: Lugar de raíces para el controlador proporcional para la función velocidad.

Debido a la baja resolución y contraste de la imagen comentaremos los aspectos más significativos de la imagen.

Para la función de transferencia de la posición la gráfica del lugar de raíces correspondiente es la siguiente. Como vemos presenta dos polos. Uno en el origen incluido por la por el factor $1/s$ que introduce la posición por ser la integral de la velocidad. El otro que aparece es el mismo que ya teníamos para la función velocidad.

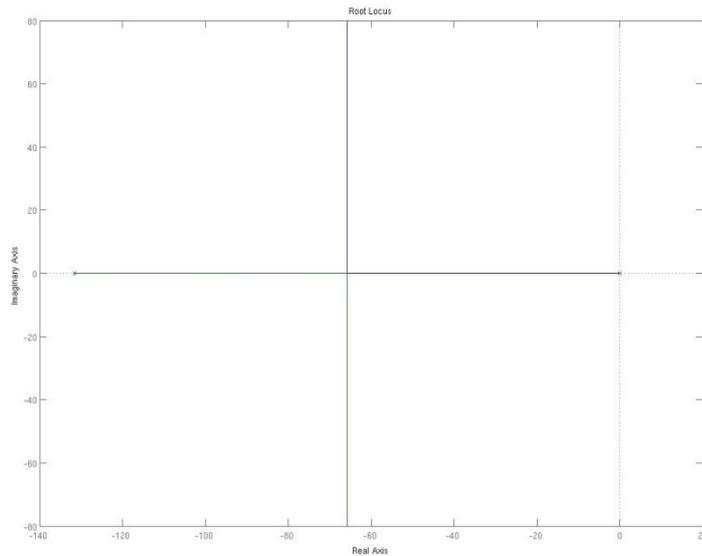


Figura 11: Gráfica del lugar de raíces para la posición en el modelo simplificado.

Si no utilizamos el modelo simplificado encontraremos que la función velocidad es de orden 2 por lo que su modelo se parecerá bastante al modelo anterior

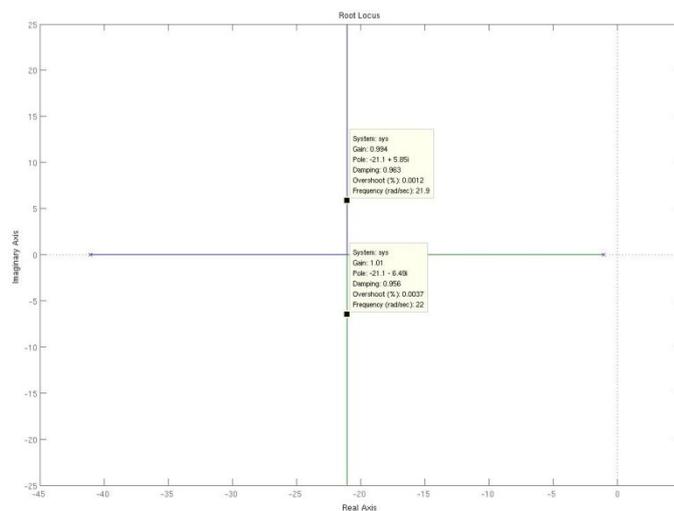


Figura 12: Gráfica del lugar de raíces para la velocidad en el modelo no simplificado.

En el eje X tenemos la parte real y en el eje Y la parte imaginaria. El punto más a la derecha representa el polo eléctrico, que como ya dijimos anteriormente está muy próximo al cero. El punto más a la izquierda representa el otro polo, aproximadamente localizado en 42. En estos puntos la ganancia es cero. Siguiendo únicamente esta rama, según vayamos aumentando la K_p del controlador, esta se irá deslizando a lo largo del eje X hasta llegar al punto de ruptura, donde a partir de la cual el valor de la K_p no nos da dos raíces reales, sino dos raíces complejas conjugadas. En esta gráfica hemos señalado los puntos en los que el parámetro “gain” se aproxima a 1. Si aumentáramos la K_p , la parte imaginaria de los polos continuaría aumentando.

Para la posición en el modelo no simplificado la cosa se complica ya que en esta ocasión tenemos tres polos.

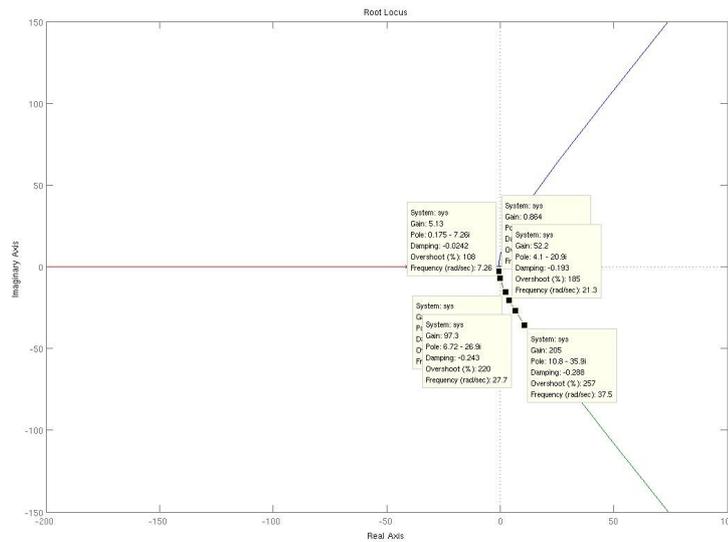


Figura 13: Gráfica del lugar de raíces para la posición en el modelo no simplificado.

Cabe destacar que con la poca resolución que ofrece el rlocus y más aun reduciéndolo a esta pequeña resolución. Un polo se encuentra en el cero y el siguiente se encuentra en un lugar muy próximo al cero.

Como ventajas de este tipo de controlador frente a otro tipo de controladores podemos destacar que obtenemos el resultado de forma instantánea. Como principales desventajas obtenemos la poca protección frente al ruido y la imposibilidad de corregir algunos errores en régimen permanente.

3 Estudio del controlador Proporcional-Integral

3.1 Estudio teórico

Como continuación del último párrafo del anterior epígrafe debemos destacar que el término integral tiene la función de corregir el error que obteníamos. Ahora nuestra función va a depender de dos parámetros:

K_p , la ganancia de nuestro controlador y T_i , el tiempo en el que se integra.

La función de transferencia para el controlador será $G_c(s) = K_p \left(1 + \frac{1}{T_i s} \right)$

Si escribimos la función de transferencia en lazo cerrado para el sistema en conjunto obtenemos

que para la velocidad es $H(s) = \frac{K * K_p * s + \frac{K * K_p}{T_i}}{s^2 + (p + K * K_p) * s + \frac{K * K_p}{T_i}}$ y para la posición

$$H(s) = \frac{K * K_p * s + \frac{K * K_p}{T_i}}{s^3 + p * s^2 + K * K_p * s + \frac{K * K_p}{T_i}}$$

Las respuestas al escalón para el controlador de velocidad y el controlador de posición obtenidas mediante simulación en Matlab son las siguientes

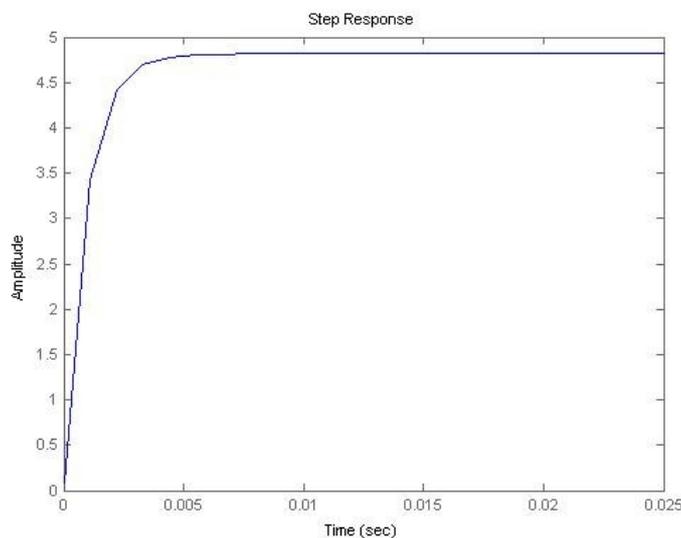


Figura 14: Respuesta al escalón para el controlador de velocidad

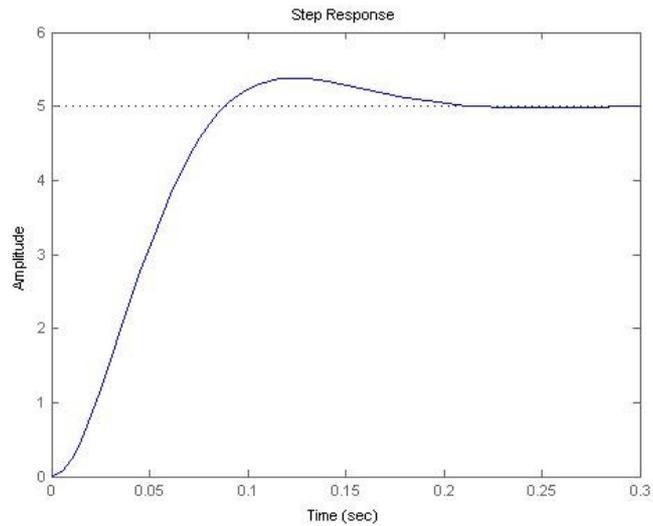


Figura 15: Respuesta al escalón para el controlador de posición

El montaje que realizamos para su implantación en la calculadora analógica es:

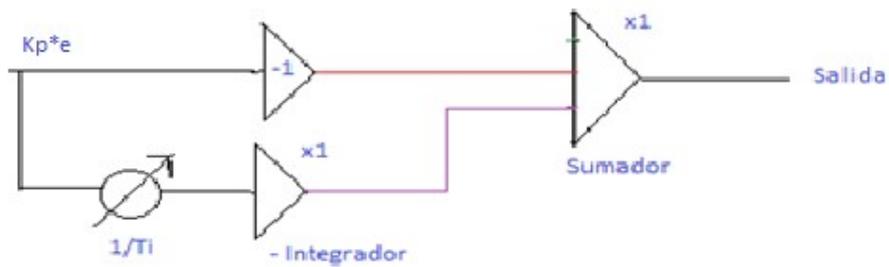


Figura 16: Montaje realizado en la calculadora analógica para simular el PI

Con este montaje nos queda que la salida $y(t) = Kp \cdot e + \frac{1}{Ti} \int Kp \cdot e dt$ que es el esquema que queríamos obtener.

3.2 Estudio práctico

A continuación reflejaremos algunas simulaciones que hemos realizado para el controlador proporcional integral. Hemos analizado el caso de que la entrada sea un escalón. Para cada imagen se adjunta el valor de K_p y T_i .

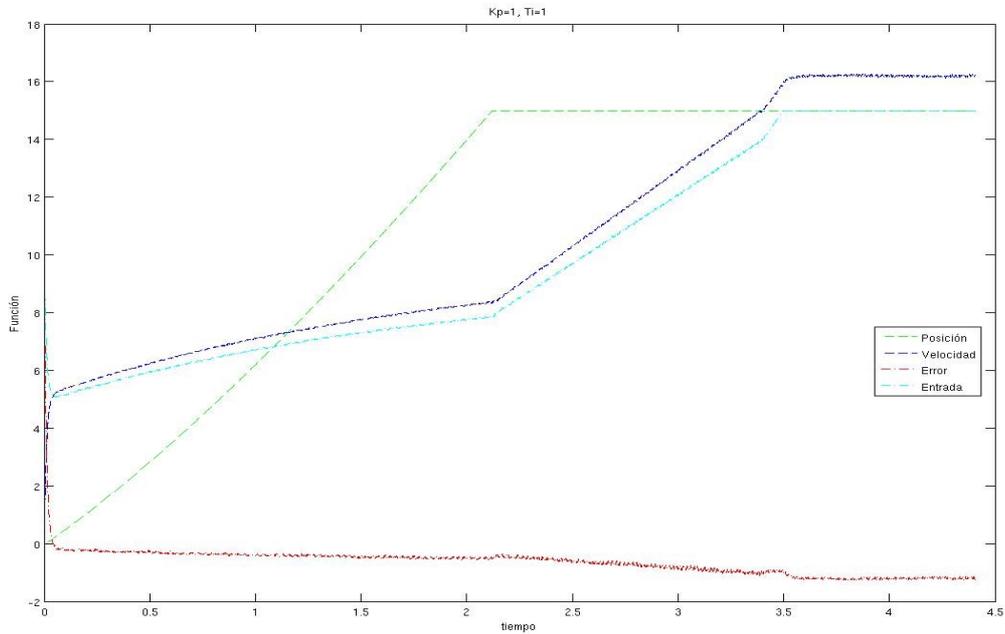


Figura 17: Respuesta del motor al escalón con $K_p = 1$ y $T_i = 1$

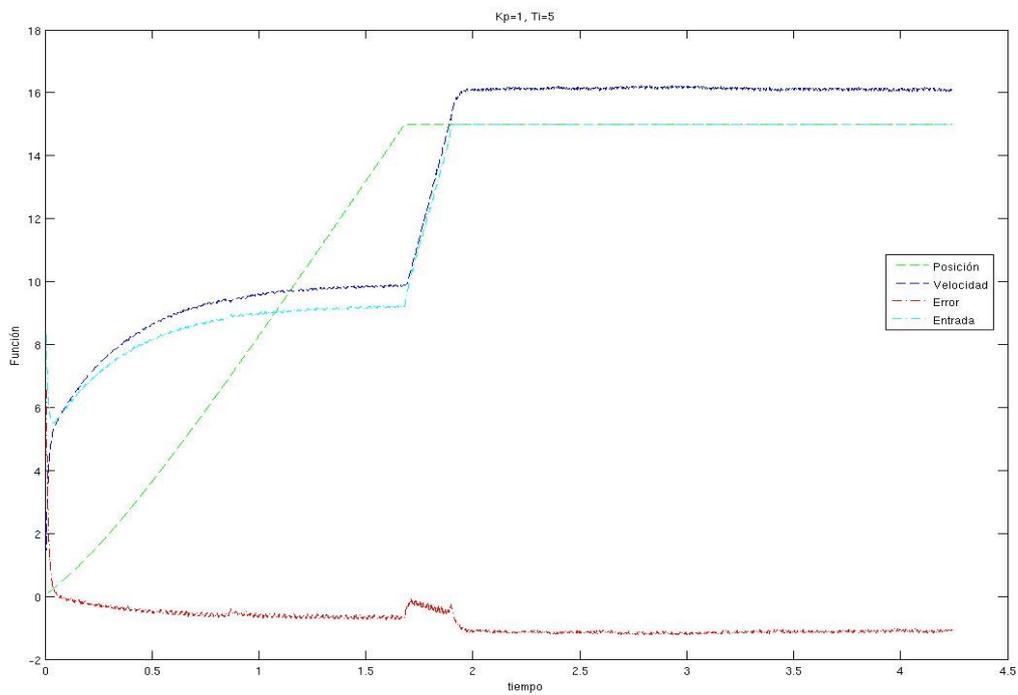


Figura 18: Respuesta del motor al escalón con $K_p = 1$ y $T_i = 5$

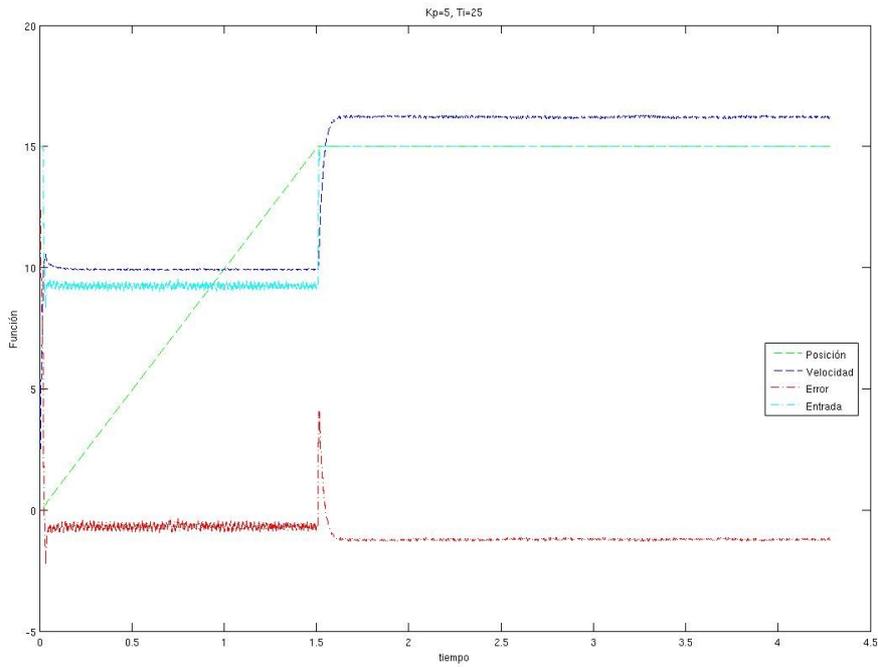


Figura 19: Respuesta del motor al escalón con $K_p = 5$ y $T_i = 25$

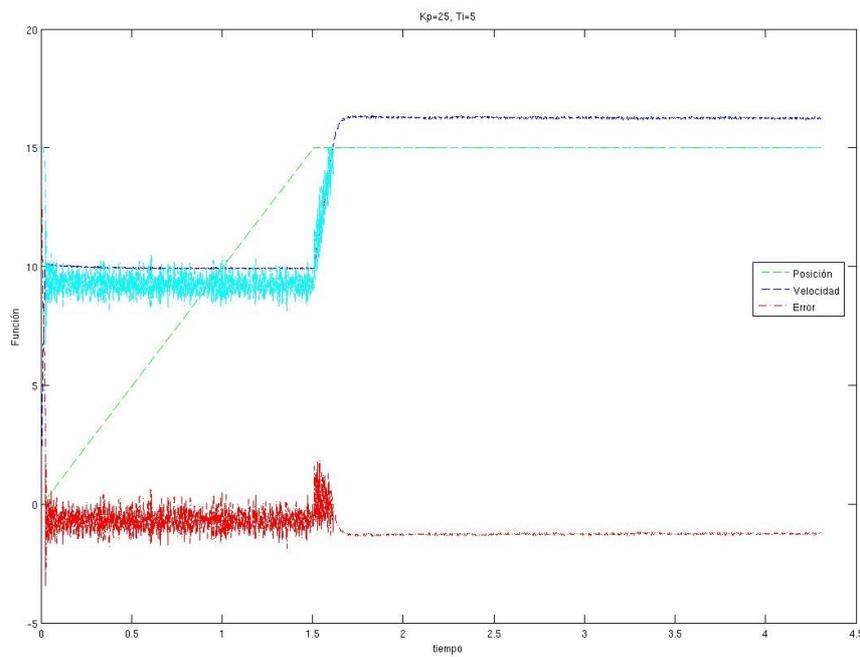


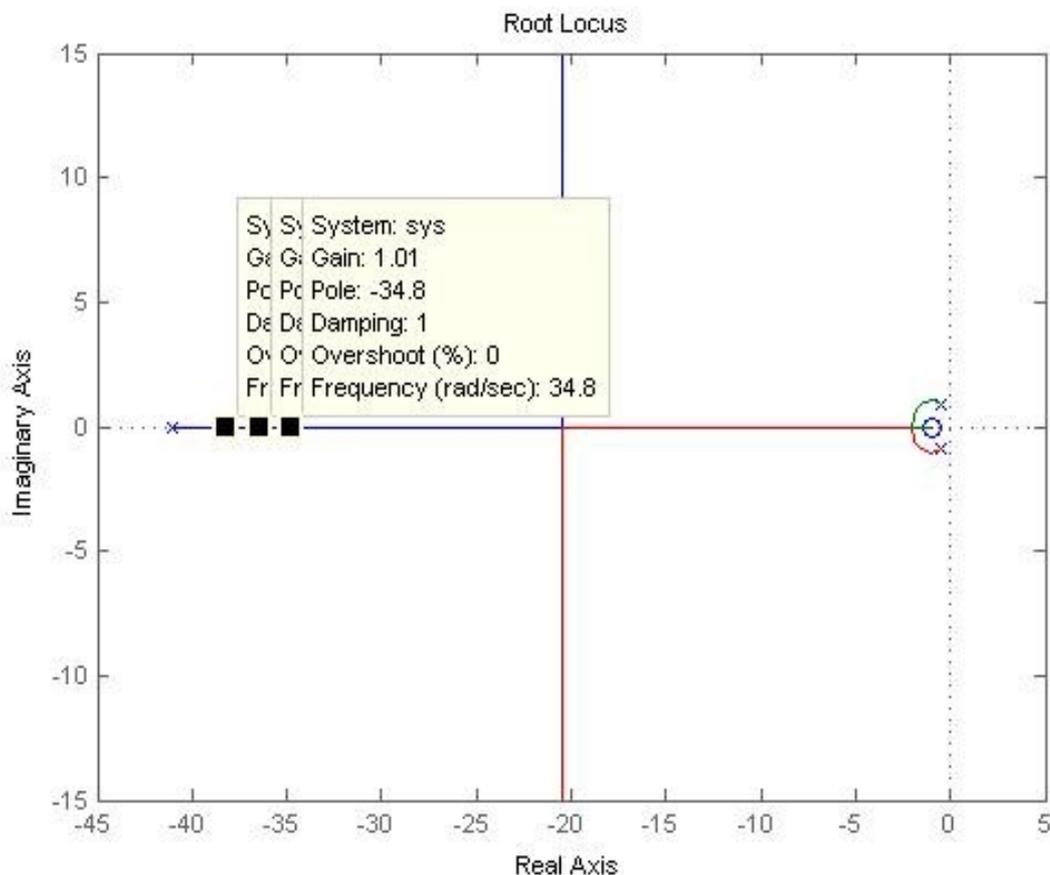
Figura 20: Respuesta del motor al escalón con $K_p = 25$ y $T_i = 5$

En este último caso podemos observar la aparición de ruido a la entrada. Esto se produce porque la entrada es muy rápida, si forzamos más incluso la entrada del motor, debido a que esta subiría más rápidamente aun, podríamos conseguir la ruptura del motor.

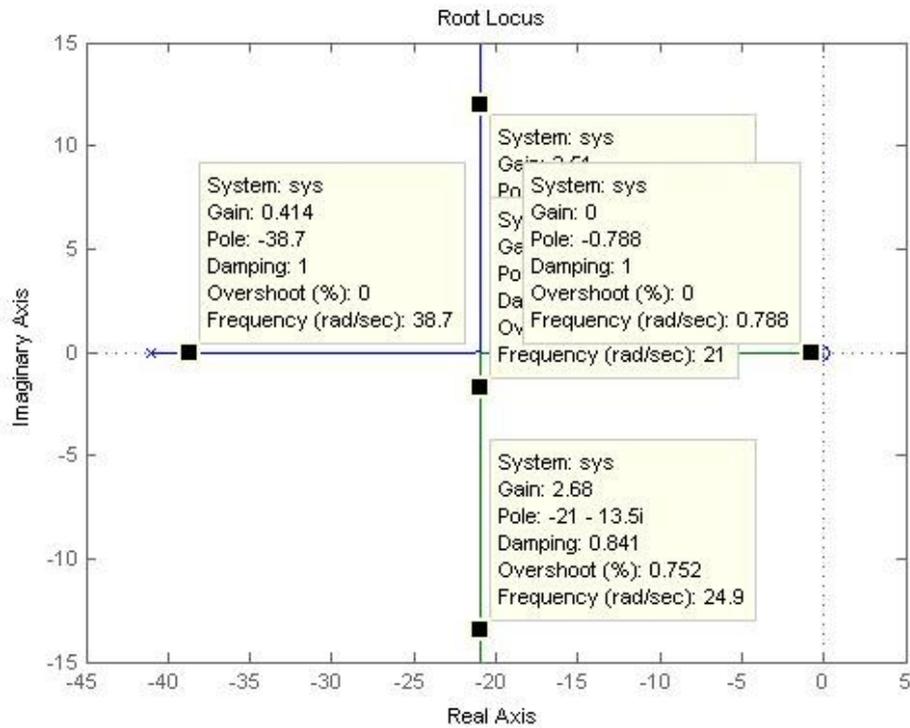
3.3 Lugar de raíces

Ahora estudiaremos la variación de la posición de los polos del controlador antes a través del lugar de raíces.

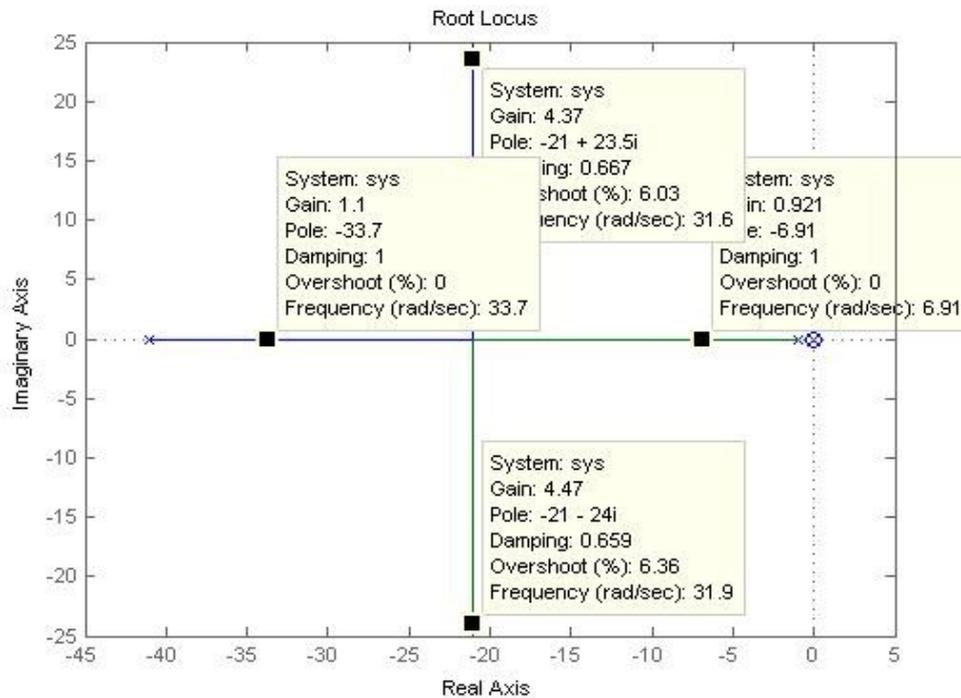
Como ya hemos ido viendo, sabemos que nuestra función de transferencia para la posición en el modelo simplificado en lazo abierto es $H(s) = \frac{K_m * T_i * s + k}{T_i * s^3 + p * T_i * s^2 + K * T_i * s + K}$ en lazo abierto. Además sabemos que la función de transferencia para la velocidad es $G(s) = H(s) * s$. Aplicando este desarrollo y la función rlocus de Matlab hemos obtenido estos resultados para el modelo simplificado.



Esta imagen es la del estudio del PI con la constante integral $T_i=1$ que se ha mostrado en apartado 3.2. Además, pese a que no se añadió a la imagen por problemas de visibilidad, se ha podido comprobar que los otros dos polos se encuentran en la recta real.



En este estudio del lugar de raíces, podemos apreciar que para una $Ti=5$, nos encontramos en la parte real mientras que la ganancia esté por debajo de poco más de 1, y ya con los valores de estudio, ($K=1$ y $K=25$) nos movemos por la zona compleja, estando $K=25$ fuera de la gráfica.



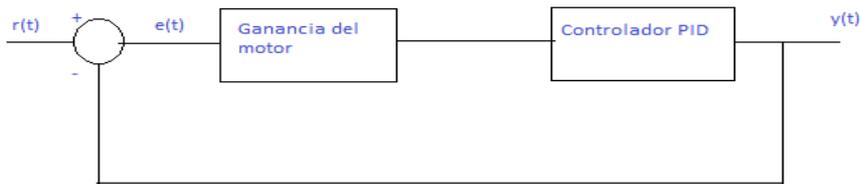
Aquí podemos apreciar como nuestros polos se encuentran en la parte compleja, para un $Ti=25$

4 Estudio del controlador PID

4.1 Estudio teórico

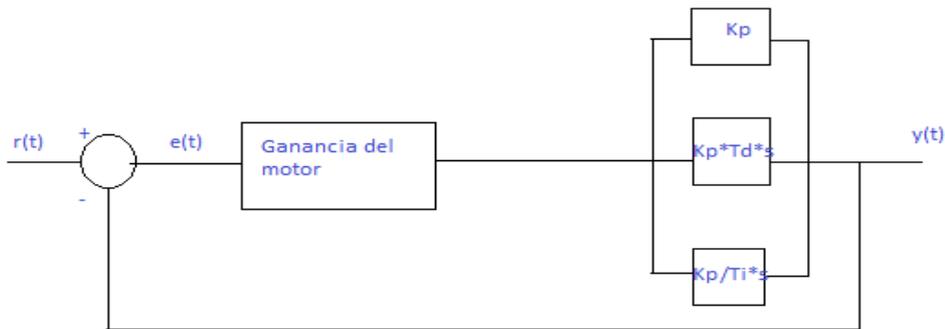
4.1.1 Estudio básico

El esquema de un sistema en lazo cerrado con un controlador PID es el siguiente:



Donde la función de transferencia del controlador PID viene dada por $H(s) = K_p * (1 + T_d * s + \frac{1}{T_i * s})$ donde el término que aparece con T_d corresponde a la parte proporcional y el término que aparece con T_i pertenece a la parte integral. Cada una de las tres partes de las que se compone el sistema nos ofrece una propiedad: La parte proporcional se encarga de estabilizar el sistema, la parte integral se encarga de corregir el error en régimen permanente y la parte derivativa funciona como un controlador predictivo del error de la salida del sistema.

Con la función de transferencia definida anteriormente podemos definir el controlador PID como la suma de tres sistemas en serie, de la siguiente manera:



Como problema importante nos encontramos que la parte derivativa del sistema no es una función causal, por lo que normalmente se suele sustituir esta parte por la aproximación de un sistema con un cero en el origen y un polo muy grande, bastante alejado del que hemos venido usando durante todo este desarrollo. Este término vendría dado por la expresión $H_d(s) = \frac{K_p * T_d * s}{1 + \frac{s * T_d}{N}}$ donde N es suficientemente

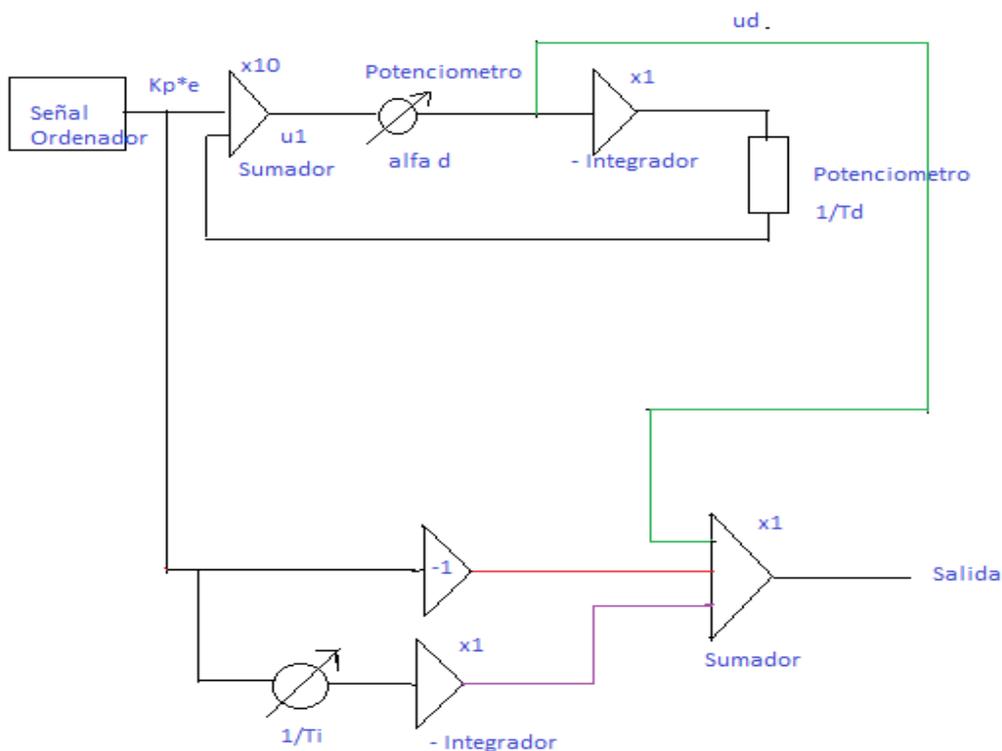
grande para cumplir la hipótesis planteada. Si N tiende a infinito entonces nos queda que $H_d(s) = K_p * T_d * s$ que coincide con la parte derivativa del sistema.

Si calculamos las funciones de transferencia en lazo cerrado para la velocidad y la posición partiendo de la fórmula general para sistemas en lazo cerrado $H(s) = \frac{G*G_c}{1+G*G_c}$ entonces obtenemos las siguientes funciones:

Para la velocidad: $H(s) = \frac{K * K_p * T_d * T_i * s^2 + T_i * K * K_p * s + K * K_p}{T_i * s(s + p) + K * K_p (T_d * T_i * s^2 + T_i * s + 1)}$

Para la posición: $H(s) = \frac{K * K_p * (T_d * T_i * s^2 + T_i * s + 1)}{T_i * s^2(s + p) + K * K_p (T_d * T_i * s^2 + T_i * s + 1)}$

Los ceros de esta ecuación vendrán dados por la ecuación $T_d * T_i * s^2 + T_i * s + 1 = 0$. Si aplicamos la fórmula de resolución de ecuaciones de segundo grado nos quedará que los valores de s que hacen cero la ecuación son de la forma $Z = \frac{1}{2T_d} * (-1 \pm \sqrt{1 - 4Y})$ donde $Y = \frac{T_d}{T_i}$. Con esta consecuencia sabemos que $4T_d < T_i$ para que existan dos ceros reales. Si $Y = 0.25$ existirá únicamente un cero doble. Si definimos los ceros como $-\alpha$ y $-\beta$ la función de transferencia del PID nos quedará $G_c = K_p * T_d * (s + \alpha) * (s + \beta)$ donde $\alpha * \beta = \frac{1}{T_i * T_d}$ y $\alpha + \beta = \frac{1}{T_d}$



En el dibujo la parte superior comprende la parte derivativa (camino verde) y la parte inferior (línea morada) la parte integral y la parte proporcional (línea roja).

Empezamos analizando la parte proporcional que es la más sencilla. La entrada es la señal procedente del ordenador que es $K_p * e$. Como el sumador que colocaremos al final invierte completamente la señal y este término ha de ser positivo acorde a nuestra función de transferencia, le colocamos anteriormente un inversor para que a la salida nos salga positivo.

En la parte integral, antes de pasar por el integrador introducimos un potenciómetro que funcionara como la constante de tiempo $1/T_i$ y que tiene que ser menor que la unidad. A continuación

pasamos por el integrador y obtenemos como señal proveniente de esta parte $\frac{-1}{T_i} * \int K_p * e$ que al pasar por el sumador perderá el signo negativo.

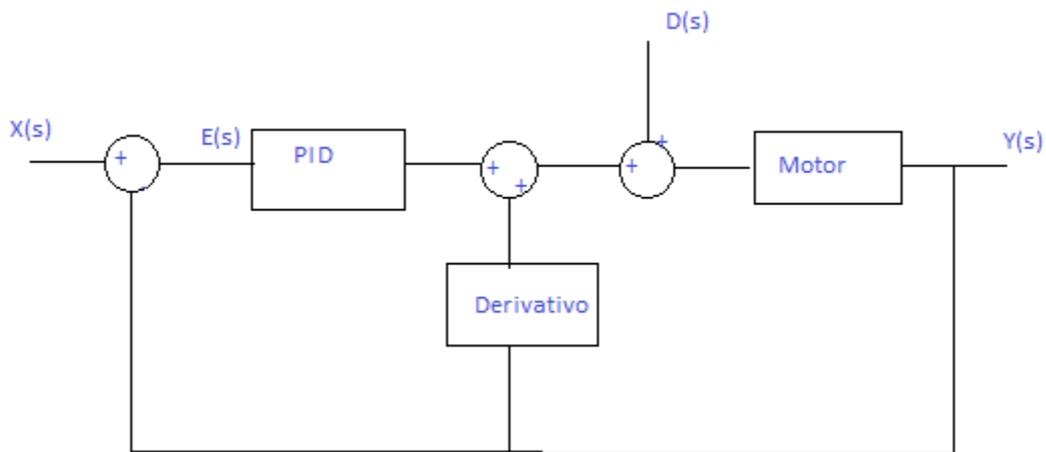
En la parte derivativa llamamos a su salida $ud = ad * u1$. Además integramos ud y la multiplicamos por un factor βd que es igual a $1/T_d$, parámetro que será controlado por un potenciómetro.

Esta salida la sumamos a la entrada del ordenador con lo que obtenemos $u1$, que si lo sustituimos en la ecuación ya descrita anteriormente tenemos que la salida de la parte derivativa es $ud = -K_p * ad * e + 10 * \beta d * ad * \int ud$.

Si sumamos todos los términos obtenemos la salida total del sistema, $salida = K_p * e + \frac{1}{T_i} * \int K_p * e + K_p * ad * e + 10 * \beta d * ad * \int ud$

4.1.2 Estudio de las perturbaciones

Una vez analizando un sistema en conjunto, nos decidimos a analizar un sistema más realista, es decir, con perturbaciones. Por ello, nos encontramos frente a un sistema con dos grados de libertad, ya que la perturbación siempre es una señal desconocida. Además, elegimos analizar aquí las perturbaciones, ya que es el controlador más amplio y con mejores prestaciones de los que hemos desarrollado a lo largo de este estudio, y para ello, hemos optado por hacer una pequeña modificación al sistema, en la cual, como podemos observar en la imagen que hay a continuación, hemos añadido una parte derivativa 'extra', la cual la sumamos a la salida de nuestro PID, manteniendo la forma del PID ya que todos los sistemas usados son lineales.



Esquema de un controlador PID con perturbaciones

Para este análisis, partimos de las siguientes ecuaciones en lazo cerrado:

- $H_x = \frac{Y_x}{X} = \frac{G * G_{c1}}{1 + G_c * G}$ Suponiendo para H_x que no hay perturbación
- $H_d = \frac{Y_d}{D} = \frac{G}{1 + G_c * G}$ Suponiendo para H_d que no hay señal de entrada
- $G_c = G_{c1} + G_{c2}$ Siendo G_{c1} la ganancia del PID, G_{c2} la del módulo derivativo y G_c la total.

Con ello, nos quedan como función de transferencia:

$$\begin{aligned}
 - G(s) &= \frac{-K}{s*(s-p1)} & Km &= \frac{-K}{p2} \\
 - Gc &= Kp*(1 + \frac{1}{Ti*s} + Td*s) = Kp*Td*(s^2 + \frac{1}{Td*s} + \frac{1}{Td*Ti}) = \frac{kp*(s+\alpha)*(s+\beta)}{s} \\
 - \alpha + \beta &= \frac{1}{Td} \\
 - \alpha * \beta &= \frac{1}{Td * Ti}
 \end{aligned}$$

Y con todo ello, obtenemos:

$$Hd = \frac{Km}{s*(s-p1) + Km*Gc} = \frac{Km*s}{s^2*(s-p1) + Km*Kp*(s+\alpha)*(s+\beta)}$$

Si suponemos que la entrada de las perturbaciones va a ser un escalón constante, observamos que el numerador se vuelve cero en el estado estacionario. Así mismo, suponemos que los polos dominantes serán complejos conjugados, por lo que $s = -a \pm jb$ mientras que el polo restante en lazo cerrado ha de ser real con valor $s = -c$

Ahora, si desarrolláramos el polinomio característico, obtendríamos las siguientes igualdades:

$$\begin{aligned}
 - 2*a + c &= Km*Kp - p1 \\
 - a^2 + b^2 + 2*a*c &= Km*Kp*(\alpha + \beta) \\
 - (a^2 + b^2)*c &= Km*Kp*\alpha*\beta
 \end{aligned}$$

Con lo que dando los valores que nos interesen a los polos obtendremos los valores de control necesarios.

4.2 Simulaciones con Matlab

Utilizando el Matlab nos hemos servido para simular varias posibilidades que nos ofrece el PID. Con las funciones en lazo cerrado calculamos su respuesta al escalón para determinados parámetros:

Primer caso: Simulación para la respuesta al escalón con parámetros $K_p=25$, $T_d=5$, $T_i=25$

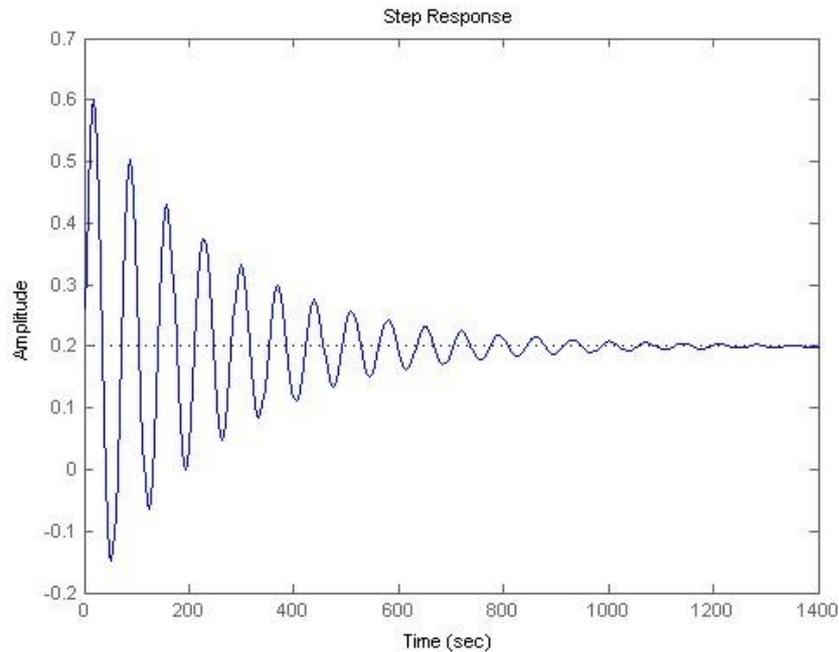


Figura 21: Resultado obtenido para la velocidad

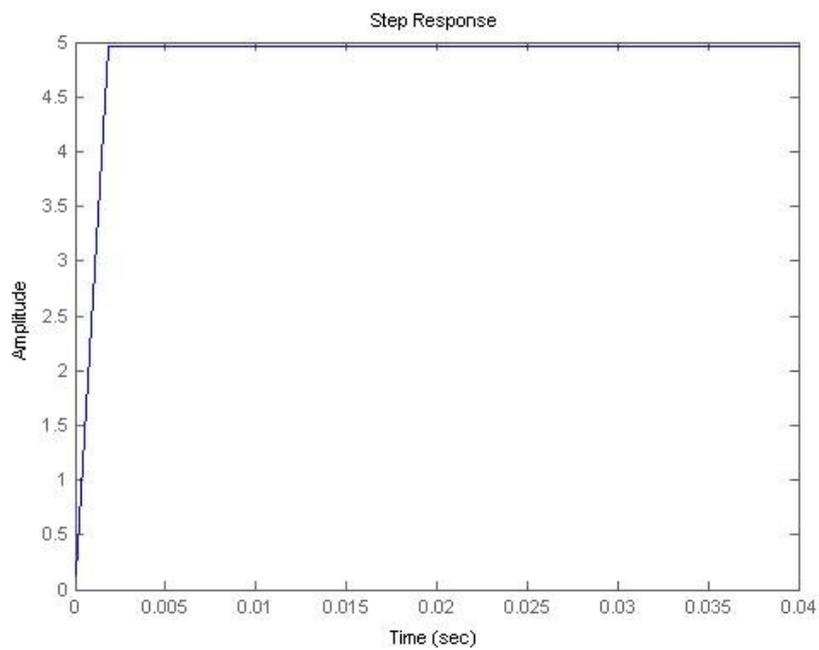


Figura 22: Resultado obtenido para la posición

Si tomamos los mismos parámetros que antes pero variando la K_p y haciéndola 10 obtenemos que:

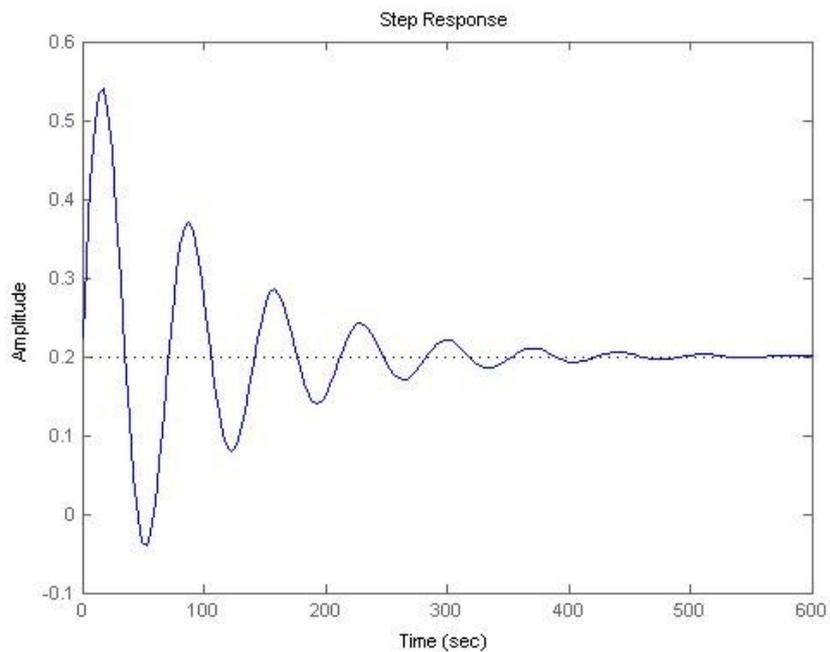


Figura 23: Resultado obtenido para la velocidad

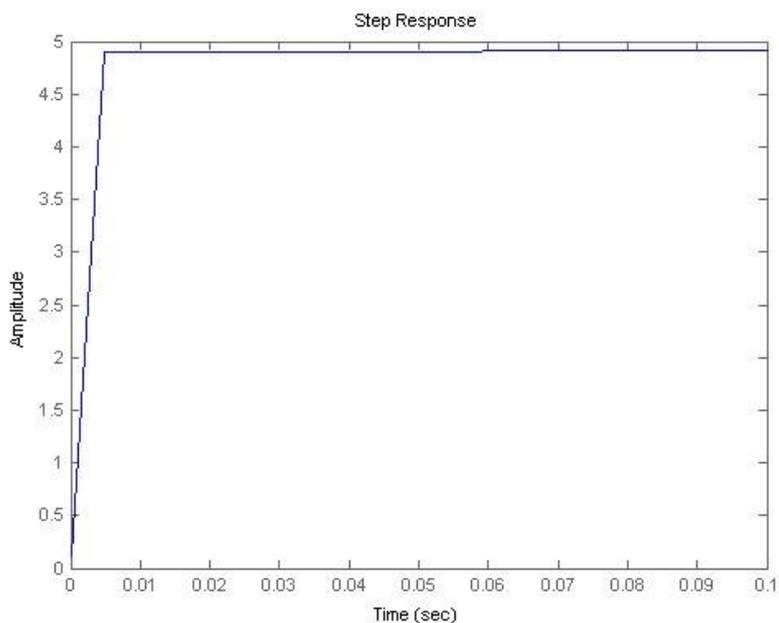


Figura 24: Resultado obtenido para la posición

De lo que deducimos que una disminución de K_p nos lleva a menor tiempo para conseguir el régimen permanente y menor tiempo de oscilación.

Si volvemos al caso original y disminuimos T_i a 4 con T_d a 1 que sería el caso extremo:

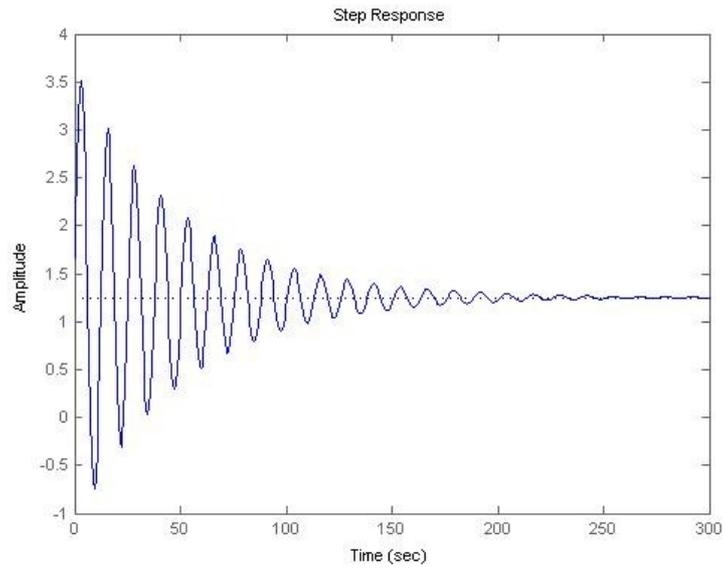


Figura 25: Resultado obtenido para la velocidad

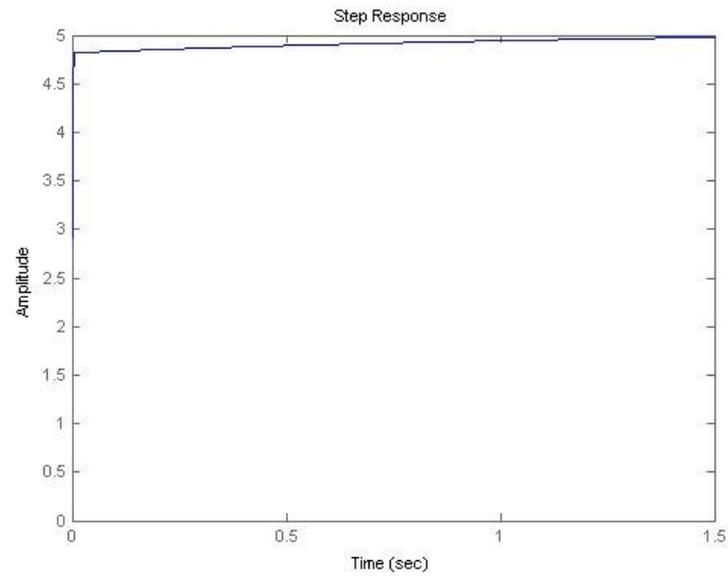


Figura 26: Resultado obtenido para la posición

Como se ve en las imágenes la disminución de las constantes de tiempo provoca una respuesta casi inmediata a la salida, mayor amplitud de oscilación pero menor duración. Con esto apenas obtenemos régimen transitorio y rápidamente pasamos a régimen permanente.

4.3 Estudio práctico

Con el montaje utilizado en la calculadora analógica siguiendo el proceso desarrollado en el estudio teórico hemos obtenido los siguientes resultados:

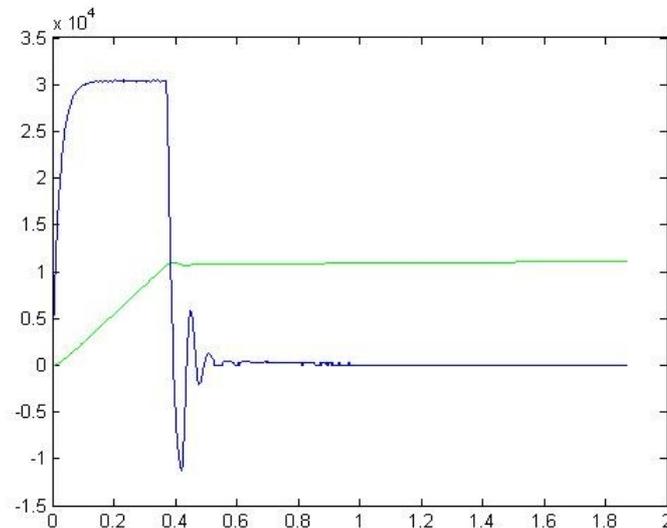


Figura 27: Resultado obtenido con la calculadora analógica

El eje X representa el tiempo y el eje Y la variación en unidades representadas por el sistema siendo la línea verde la velocidad y la línea azul la posición a la salida del sistema. El tiempo de establecimiento del régimen permanente es de 0.4 segundos. Se produce una sobreelección máxima de 7 Voltios para una entrada de 5 en la primera oscilación y de 2 V en segunda oscilación. Podemos ver que la posición tiende a cero con el paso del tiempo, así que pasado un tiempo = 1 segundo, permanecerá en su posición. Los valores introducidos en la calculadora analógica fueron en potenciómetros de $10K\Omega$ para el potenciómetro de $8'95K\Omega$ para la variable $\beta=1/Td$ y de $9'54K\Omega$ para la variable $\alpha_i=1/Ti$.

Para reducir la sobreelección del sistema disminuimos el valor del primer potenciómetro

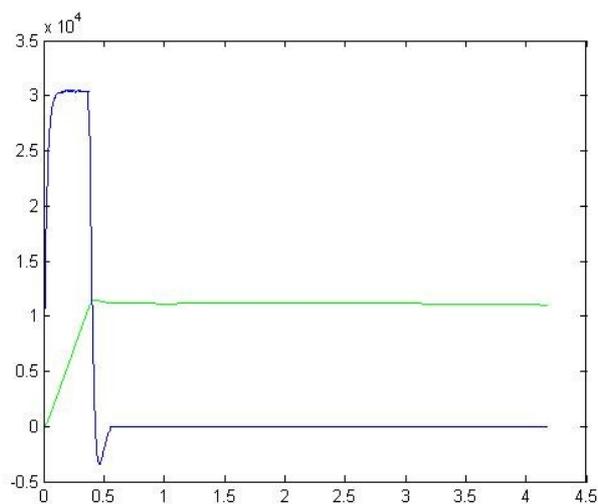


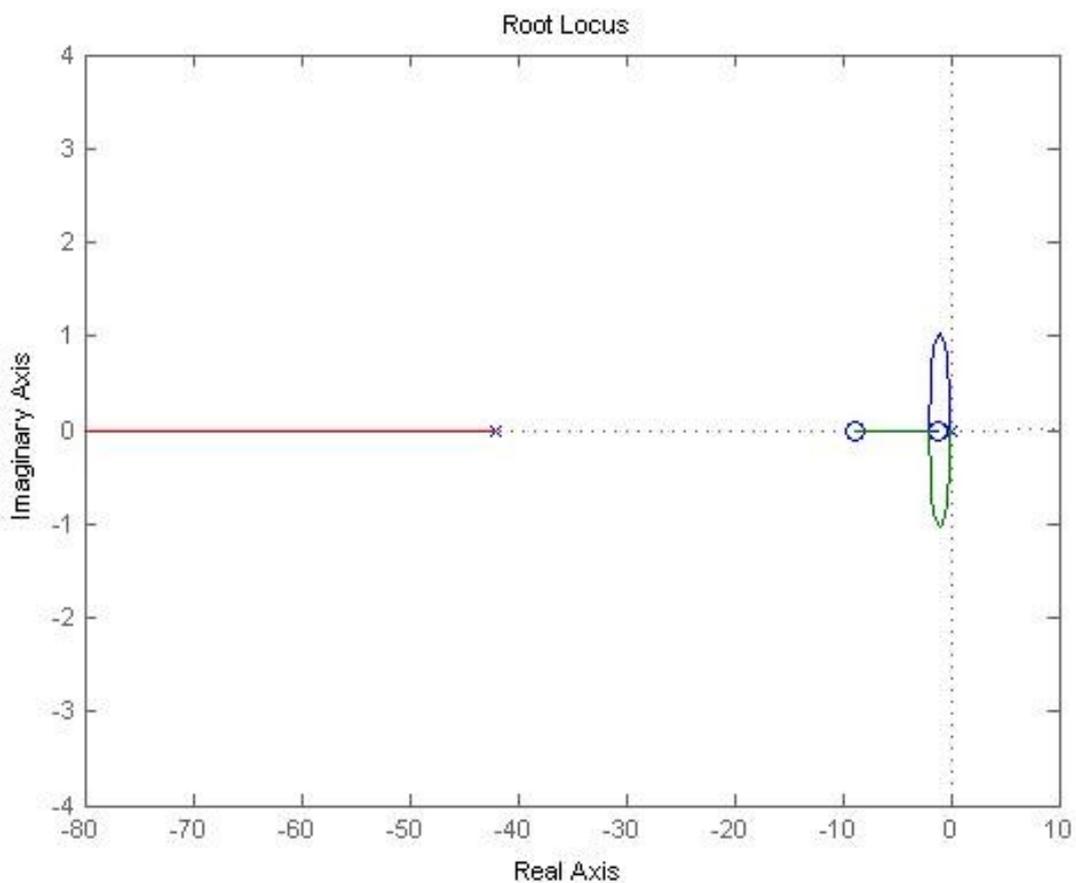
Figura 28: Resultado obtenido con la calculadora analógica

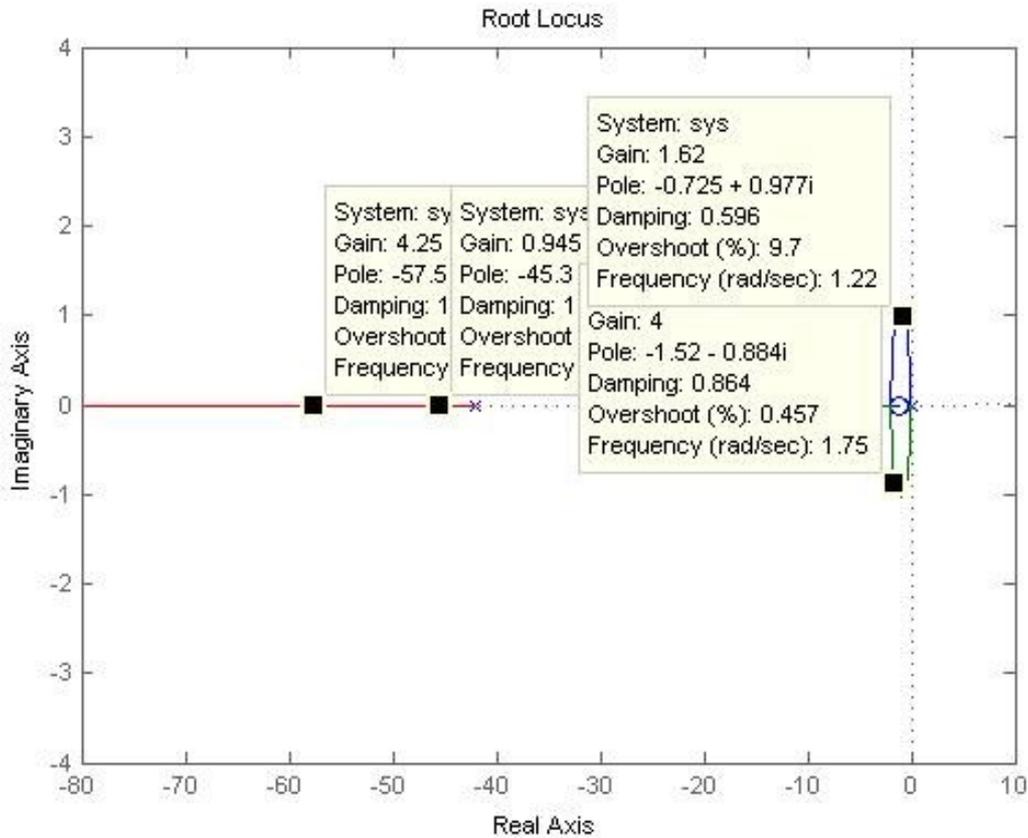
Como podemos ver en este caso ya ni tan siquiera se produce una segunda sobreelección. Además la amplitud de la primera se reduce a 2 V.

4.4 Lugar de raíces

Ahora estudiaremos la variación de la posición de los polos del controlador antes a través del lugar de raíces.

Como ya hemos ido viendo, sabemos que nuestra función de transferencia para la posición en el modelo simplificado en lazo abierto es $H(s) = \frac{K * Td * Ti * s^2 + K * Ti * s + K}{Ti * s^2 * (s + p) + K * Ti * s + K * Ti * Td * s^2 + K}$ en lazo abierto. Además sabemos que la función de transferencia para la velocidad es $G(s) = H(s) * s$. Aplicando este desarrollo y la función rlocus de Matlab hemos obtenido estos resultados para el modelo simplificado.





Ambas imágenes son la misma, solo que la segunda muestra los valores de la K en función de su posición.

4.5 Calculadora analógica

En este apartado, se va a analizar los problemas encontrados en el laboratorio al usar la calculadora analógica, la cual cuenta con elementos reales como son los amplificadores operacionales, los cuales tienen una serie de limitaciones físicas a la hora de utilizarlos, como son la tensión de saturación o el offset a la salida propio de los amplificadores.

4.5.1 Saturación

Este problema aparece al tener que usar sumadores o integradores, donde el amplificador operacional encargado de dicha función, se ve obligado a sumar diversas señales, las cuales, si tienen demasiada amplitud pueden llegar a requerir una tensión de salida superior a la tensión de alimentación del amplificador lo cual causa que la salida deje de mantener una relación lineal con la entrada y pase a generar una señal continua mientras que en la entrada se tienen unas señales que varían con el tiempo, estando su resultado matemático de la operación por encima del valor de saturación. Esto causa que el sistema deje de controlar correctamente y el sumador/integrador se comporte como una entrada de tensión continua independiente de la entrada, por lo que, en el sistema de PID estudiado en este apartado, si todos los sumadores/integradores estuvieran saturados, la función de transferencia del controlador

sería, $G_c = \frac{V_{sat}}{s}$ siendo V_{sat} la tensión de offset del último sumador. Como se ve, se ha perdido toda relación con la entrada así como todas y cada una de las partes que componían el PID.

Este es un problema que se corrige en el diseño, y para ello hay dos vías. La primera consiste en ajustar las tensiones de entrada de nuestro sistema para que se mantengan dentro de un margen que sabemos que no saturará ninguno de nuestros amplificadores. Este método nos puede acarrear problemas ya que algunas aplicaciones pueden requerir de altas tensiones para trabajar por lo que el precio de nuestro controlador se dispararía si usáramos siempre amplificadores capaces de soportar esas tensiones, lo cual nos lleva a la segunda opción, que consiste en atenuar la entrada a nuestro controlador para así evitar que se sature el sistema, y luego, la salida final amplificarla lo atenuado para que, si tomamos todo el controlador íntegro como una 'caja negra', este nuevo controlador sea indistinguible de el modelo anterior sin esta etapa de atenuación-amplificación. Este sistema, también tiene un defecto y es que al atenuar la señal, se pierde resolución y se reduce la robustez frente al ruido interno del controlador, ya que la señal de entrada se ve reducida para evitar la saturación. A cambio, nos ofrece la posibilidad de generar un controlador con componentes normales, y luego aplicarlo a un sistema de alta potencia simplemente añadiendo el atenuador y el amplificador previamente descritos.

4.5.2 Offset

Este problema también viene causado por los amplificadores operacionales reales, ya que estos a la salida, además de ofrecer una señal lineal para con la entrada (suponiendo que no hay saturación) ofrece una tensión constante propia de la física interna del propio amplificador e independiente de la señal de entrada. Esto causa que el sistema nunca alcance el un régimen permanente en el valor deseado ya que el offset siempre estará presente. Además, en el caso de los derivadores, como han sido implementados con un integrador y un sumador, este ligero offset siempre se está sumando a la entrada del integrador favoreciendo su saturación. Para mitigar este efecto la única posibilidad sería recurrir a usar un método visto antes, pero a la inversa, es decir, podríamos intentar trabajar con la señal pre-amplificada, para que luego al atenuarla al final de controlador, el offset se vea dividido por un número suficientemente alto como para que su efecto sea despreciable.

En nuestro caso práctico, si pudiéramos pre-amplificar la señal con un x10, luego nuestro offset de salida se vería reducido a escasos mV, la cual podría ya resultar despreciable a efectos prácticos, pero esta medida nos es inviable ya que nos obligaría a saturar el sistema.

Finalmente, otra posible opción que podría ser planteada es la inclusión de un filtro a la corriente continua, que es la generada por el offset, pero a poco que se empieza a pensar en su implantación queda patente su inviabilidad ya que el propio sistema trata de tender hacia un valor estable, que genera también una corriente continua, y que obviamente, se vería afectada por el filtro.

5 Bibliografía

- Sistemas Automáticos de Control, de Benjamin C. Kuo, Segunda Edición (CECSA)
- Ingeniería de Control Moderna, de Katsuhiko Ogata, Cuarta Edición (Prentice Hall)
- Estudio de un PID, de Félix Monasterio-Huellín,
(http://robolabo.etsit.upm.es/a_lsctr/apuntes2/controlador_pid.pdf)
- http://www.automatas.org/hardware/teoria_pid.htm

6 ANEXO: Código de Matlab

Código empleado para calcular los polos:

```
%X son los valores que alcanza el motor
%t es el tiempo es en el que lo alcanza
x = [4801.5 5025.5 8278.5 9439.5 10232.5000000000000000 11058.5000000000000000];
t = [0.0135540002956986 0.0143120000138879 0.0300789996981621 0.0417600013315678 0.0550389997661114
0.0903519988059998];

%Calcula el lugar donde se encuentra el polo realizando la media.

%No se ha utilizado la función mean aunque también era posible
y = -log ( 1-(x/11320))./t
c = sum (y)/length(y)
d=[ c c c c c c]
plot(x, y, 'b', x,d,'g')
xlabel ('Tensión (UnidadesMotor)')
ylabel ('Localización del polo')
grid

axis ([4500 11500 40.7 44])
```

Código empleado para el calculo de la elongación máxima:

```
Kp = [1 5 25 50 100 200];

Ep = (10.16./Kp).^(1/2)
Mu = 5*exp ( -pi*Ep./sqrt(1- Ep.^2))
%Elongacion en funcion de Kp
subplot (2,1,1)
plot( Kp, Ep, 'g--')
grid
title ( ' K y Ep')
xlabel ('K')
ylabel ('Ep')
[x y] = ginput(1)
subplot (2,1,2)
plot (Kp, Mu, 'b--')
grid
title ('Kp y Mu')
xlabel ('Kp')
ylabel ('Mu')

Epn = [0:0.1:1.3];
Kn= 10.16./Epn.^2;
Mun= exp ( -pi*Epn./sqrt(1- Epn.^2));
```

Código empleado para mostrar los resultados de un experimento con un controlador PI:

```
format long
load 'PI1_1'
load 'PI5_1'
load 'PI25_1'
load 'PI50_1'
load 'PI100_1'
load 'PI200_1'

x1 = PI1_1(:,1);
y1 = PI1_1(:,3)/2184.5;
x2 = PI5_1(:,1);
y2 = PI5_1(:,3)/2184.5;
x3 = PI25_1(:,1);
y3 = PI25_1(:,3)/2184.5;
x4 = PI50_1(:,1);
y4 = PI50_1(:,3)/2184.5;
x5 = PI100_1(:,1);
y5 = PI100_1(:,3)/2184.5;
x6 = PI200_1(:,1);
y6 = PI200_1(:,3)/2184.5;

figure
subplot (3,1,1)
plot( x1, y1, 'g--')
title ( 'Kp=1, Ki=1')
xlabel ('Tiempo')
ylabel ( 'Velocidad')
subplot (3,1,2)
plot( x2, y2, 'g--')
title ( 'Kp=5, Ki=1')
xlabel ('Tiempo')
ylabel ( 'Velocidad')
subplot (3,1,3)
plot( x3, y3, 'g--')
title ( 'Kp=25, Ki=1')
xlabel ('Tiempo')
ylabel ( 'Velocidad')

figure
subplot (3,1,1)
plot( x4, y4, 'g--')
title ( 'Kp=50, Ki=1')
xlabel ('Tiempo')
ylabel ( 'Velocidad')
subplot (3,1,2)
plot( x5, y5, 'g--')
title ( 'Kp=100, Ki=1')
xlabel ('Tiempo')
ylabel ( 'Velocidad')
subplot (3,1,3)
plot( x6, y6, 'g--')
title ( 'Kp=200, Ki=1')
```

```
xlabel ('Tiempo')
ylabel ('Velocidad')
```

```
figure
plot (x1,y1,'g-', x2,y2,'b--',x3,y3,'r:', x4,y4,'c-', x5,y5,'k--', x6,y6,'m-')
legend ('K=1', 'K=5', 'K=25', 'K=50', 'K=100', 'K=200', 'Location', 'SouthEast')
grid
xlabel ('Tiempo')
ylabel ('Velocidad')
```

Código empleado para representar gráficamente los datos obtenidos con un PID:

```
load PID_2.txt

%Obtiene los datos del fichero PID2 y los pinta

x= PID_2(:,1);

y= PID_2(:,2);

z= PID_2(:,3);

t= PID_2(:,4);

plot (x,y,'g', x, z, 'b', x,t, 'r')
```

Código para observar la variación de un PI según la variación del parámetro Ti

```
format long
load 'Rampa1'
load 'Rampa2'
load 'Rampa3'
load 'Rampa4'
load 'Rampa5'
load 'Rampa6'

x1 = Rampa1(:,1);
y1 = Rampa1(:,3)/2184.5;
x2 = Rampa2(:,1);
y2 = Rampa2(:,3)/2184.5;
x3 = Rampa3(:,1);
y3 = Rampa3(:,3)/2184.5;
x4 = Rampa4(:,1);
y4 = Rampa4(:,3)/2184.5;
x5 = Rampa5(:,1);
y5 = Rampa5(:,3)/2184.5;
x6 = Rampa6(:,1);
y6 = Rampa6(:,3)/2184.5;
s1 = Rampa1(7374:7384,1);
t1 = Rampa1(7374:7384,3)/2184.5;
s2 = Rampa2(2384:7384,1);
t2 = Rampa2(2384:7384,3)/2184.5;
s3 = Rampa3(2384:7384,1);
```

```
t3 = Rampa13(2384:7384,3)/2184.5;
s4 = Rampa14(2384:7384,1);
t4 = Rampa14(2384:7384,3)/2184.5;
s5 = Rampa15(2384:7384,1);
t5 = Rampa15(2384:7384,3)/2184.5;
s6 = Rampa16(2384:7384,1);
t6 = Rampa16(2384:7384,3)/2184.5;
```

```
figure
subplot(3,1,1)
plot(x1, y1, 'g--')
title('K=1, alfa=5')
xlabel('Tiempo')
ylabel('Velocidad')
subplot(3,1,2)
plot(x2, y2, 'g--')
title('K=5, alfa=5')
xlabel('Tiempo')
ylabel('Velocidad')
subplot(3,1,3)
plot(x3, y3, 'g--')
title('K=25, alfa=5')
xlabel('Tiempo')
ylabel('Velocidad')
```

```
figure
subplot(3,1,1)
plot(x4, y4, 'g--')
title('K=50, alfa=5')
xlabel('Tiempo')
ylabel('Velocidad')
subplot(3,1,2)
plot(x5, y5, 'g--')
title('K=100, alfa=5')
xlabel('Tiempo')
ylabel('Velocidad')
subplot(3,1,3)
plot(x6, y6, 'g--')
title('K=200, alfa=5')
xlabel('Tiempo')
ylabel('Velocidad')
```

```
max1= max(y1);
max2= max(y2);
max3= max(y3);
max4= max(y4);
max5= max(y5);
max6= max(y6);
```

```
max= [max1 max2 max3 max4 max5 max6]
```

```
mean1= mean(t1);
mean2= mean(t2);
mean3= mean(t3);
```

```
mean4= mean(t4);
mean5= mean(t5);
mean6= mean(t6);
```

```
mean= [mean1 mean2 mean3 mean4 mean5 mean6]
```

```
difmax = (max-mean)./mean
```

Código para la respuesta de tercer orden

```
format long
```

```
Ke=2.2*10^-2;
L=0.00058;
J=1.4 *10^-6;
B=9.3*10^-5;
R=5.3;
Km=2.2*10^-2;
V= 5;
```

```
g = tf([Km],[(J*L) (R*J+B*L) (Km*Ke+R*B) 0])
b1 = zpk(g);
p1 = b1.p{:};
rlocus (g)
```

```
%Respuesta al escalón para la función de velocidad
```

```
num =[Km];
den =[L*J L*B+R*J R*B];
step (num, den);
pause
```

```
%Respuesta al escalón para la función de posición
```

```
denp =[L*J L*B+R*J R*B+Ke*Km 0];
step (num, denp)
```

Código para la simulación del motor con un controlador PID

```
Kp=25;
```

```
K= 43.64;
```

```
p=42.11;
```

```
Vin= 5;
```

```
Td=1;
```

```
Ti=4;
```

```
%Respuesta al escalón para la función en Velocidad de un controlador PID
```

```
num= [K*Kp*Vin*Ti K*Kp*Vin];
```

```
den= [Ti p*Ti+K*Kp*Ti K*Kp];
```

```
step(num, den)
```

```
pause
```

```
tf(num,den)
```

```
den1 =[Ti p*Ti K*Kp*Ti K*Kp];
```

```
%Respuesta al escalón para la función en Posición de un controlador PID
```

```
step(num, den1)
```

Código para la simulación del motor con un controlador Proporcional y con un PI

```
K=43.64;
```

```
p= 42.11;
```

```
Vin= 5;
```

```
Ti=5;
```

```
Kp=25;
```

```
Td=1;
```

```
%Respuesta al escalón para la función en Velocidad de un controlador Proporcional
```

```
num1 = [K*Kp*Vin];
```

```
den1 = [1 p+K*Kp];
```

```
step(num1, den1)
```

```
pause
```

```
%Respuesta al escalón para la función en Posición de un controlador Proporcional
```

```
num2 =[K*Kp*Vin];
```

```
den2 = [1 p K*Kp];
```

```
step(num2, den2)
```

```
pause
```

```
%Respuesta al escalón para la función en Velocidad de un controlador Proporcional-Integral
```

```
num3 = [Ti*K*Kp*Vin K*Kp*Vin];
```

```
den3=[Ti p*Ti K*Kp*(1+Ti) ];
```

```
step(num3, den3)
```

```
pause
```

```
%Respuesta al escalón para la función en Posición de un controlador Proporcional-Integral
```

```
num4=[K*Kp*Ti*Vin K*Kp*Vin];
```

```
den4= [Ti p*Ti K*Kp*Ti K*Kp];
```

```
step(num4, den4);
```