Guía Experiencia 1:
Nociones básicas de programación en MATLAB-SIMULINK para el estudio de sistemas dinámicos

Guía preparada por el profesor: Juan C. Agüero

para la asignatura *Laboratorio de Control Automático* (ELO-271)

Sesión 1: 19 de Marzo de 2024
Sesión 2: 26 de Marzo de 2024

Índice

1. Introducción

La presente experiencia tiene por objetivo introducir el software de programación MATLAB, y su entorno de simulación SIMULINK. Este software se utilizará durante todas las experiencias restantes del curso, sin embargo, dada la versatilidad que este posee, su dominio le será útil para el estudio de muchos otros tópicos relacionados con ingeniería.

En esta experiencia se proponen algunas actividades de desarrollo simple para su implementación en MATLAB/SIMULINK. Las actividades planteadas servirán para refrescar su conocimiento respecto a *sistemas lineales de tiempo continuo*, pero además servirán para introducir nuevas herramientas como la llamada *aproximación de Euler*, o el *algoritmo de mínimos cuadrados*. Note que este documento no es un tutorial de Matlab, por lo que **se espera que usted estudie previamente** los conceptos básicos (entorno de programación, sintaxis, comandos básicos, etc).

2. Trabajo Previo

El software Matlab es ámpliamente utilizado en la academia, por lo que existen muchos tutoriales en internet que usted puede usar para aproximarse a Matlab (revise el apéndice de la guía para sugerencias). El objetivo no es que usted se convierta en un experto, si no que se familiarice con los comandos básicos y con los comandos útiles para el desarrollo de nuestro trabajo. A continuación se presenta una lista de comandos y tareas que deberían ser familiares para usted una vez haya estudiado el (o los) tutorial (es) que haya escogido.

- Entornos de Matlab: workspace, command windows, history, current folder.
- Búsqueda de ayuda: help, lookfor
- Tipos de variables: arrays, cell, struct, sym
- Definición de vectores, matrices, y trabajo con índices (min, max, find, reshape, etc).
- Funciones básicas: sin, cos, exp, log, ^ (potencias).
- Operaciones básicas de matrices: suma, resta, multiplicación de matrices (*), multiplicación por componente (.*), inversa, transpuesta.
- Generación de figuras: plot, stem, bar, hold on, grid on, legend, axis, xlabel, ylabel, title, subplot, semilogx, linspace, logspace.
- Editar figuras, guardar en formatos específicos.
- Desarrollo de scripts en MATLAB .
- Bucles y condiciones: for, end, while, if, else,
- Guardar, cargar y ver variables en el *workspace*: save, load, whos.
- Limpieza: clc, clear, close all
- Representación de sistemas dinámicos en MATLAB : tf, zpk, ss.
- Análisis de sistemas dinámicos en MATLAB : pole, zero, step, impulse, bode, freqresp.
- Simulaciones de sistemas dinámicos en SIMULINK .

El trabajo con estos comandos y entornos **forma parte del trabajo previo de la experiencia 1**, y se espera que usted llegue a la sesión en laboratorio con dichos comandos y tareas estudiadas.

Se asumirá que usted maneja los conceptos asociados a sistemas lineales tales como la respuesta a entrada escalón y sinusoidal de sistemas de primer y segundo orden, y respuesta en frecuencia. Si no es el caso, el estudio de estos conceptos se considerará parte del trabajo previo a desarrollar.

3. Fundamentos y conceptos asociados a la experiencia

3.1. Simulación de sistemas dinámicos

En el curso de Análisis de Sistemas Lineales usted estudió la representación de sistemas lineales invariantes en el tiempo y su respuesta a ciertas entradas. Para el caso de sistemas de tiempo continuo, si usted conoce una función analítica para describir la señal de entrada, entonces puede obtener una función analítica para la salida resolviendo la ecuación diferencial asociada. Es por ello que usted ya conoce la respuesta (analítica) de un sistema lineal a entrada escalón, impulso, sinusoidal, etc.

Ahora, imagine usted que por algún motivo no puede calcular analíticamente la solución de la ecuación diferencial asociada (por ejemplo, si la señal de entrada es difícil de describir analíticamente, o si solo la conoce a medida que llega al sistema y no previamente). Para aquellos casos se pueden emplear métodos numéricos para obtener una solución aproximada de la ecuación diferencial relacionada. En particular en esta experiencia introduciremos el método de Euler (pero sepa que existen muchos más).

Aproximación de Euler

Dada una señal $y(t)$, su derivada (con respecto al tiempo) está definida como

$$\dot{y}(t) = \lim_{\Delta \to 0} \frac{y(t+\Delta) - y(t)}{\Delta} \tag{3.1}$$

Si relajamos el requerimiento $\Delta \to 0$, podemos obtener una aproximación de $\dot{y}(t)$ para un tiempo determinado. Esta idea es explotada en la *aproximación de Euler*, que tiene la forma

$$\dot{y}(t)|_{t=n\Delta} \approx \frac{y(n\Delta + \Delta) - y(n\Delta)}{\Delta} \tag{3.2}$$

donde $n = 0, 1, 2, \ldots$, y Δ es un intervalo de tiempo pequeño. Note que esta aproximación se hace para instantes discretos de tiempo, espaciados un intervalo constante Δ, por lo que las señales involucradas en la aproximacion pueden interpretarse como señales de tiempo discreto.

Qué relación tiene esta aproximación con la simulación de sistemas dinámicos?

Si deseamos simular el comportamiento de un sistema continuo mediante un computador, debemos tener claro que lo que realmente se simula es una aproximación discreta del modelo del sistema continuo real. Esto ya que un computador trabaja con señales digitales (de tiempo discreto), y por lo tanto no pueden representar con resolución infinita la dinámica de señales de tiempo continuo. En MATLAB/Simulink, los sistemas de tiempo continuo pueden *representarse* de diversas formas (por ejemplo usando el comando tf, ss, etc), pero el *cálculo de la respuesta* de dicho sistema a una determinada señal de entrada requiere una aproximación como la indicada anteriormente.

Método de Euler para aproximar sistemas continuos

Considere un sistema lineal G_c, con condición inicial cero, representado en función de transferencia de Laplace por

$$G_c(s) = \frac{b_n s^n + b_{n-1} s^{n-1} + \ldots + b_0}{a_n s^n + a_{n-1} s^{n-1} + \ldots + a_0}. \tag{3.3}$$

Podemos utilizar el método de Euler para aproximar el sistema lineal continuo descrito por $G_c(s)$ a través de una función de transferencia de un sistema de tiempo discreto de muestreo Δ (lineal e invariante en el tiempo) simplemente reemplazando s por $\frac{z-1}{\Delta}$ en $G_c(s)$ (por qué?)

Así, el sistema de tiempo discreto queda descrito por la función de transferencia

$$G_d(z) = \frac{b_n \left(\frac{z-1}{\Delta}\right)^n + b_{n-1}\left(\frac{z-1}{\Delta}\right)^{n-1} + \ldots + b_0}{a_n \left(\frac{z-1}{\Delta}\right)^n + a_{n-1}\left(\frac{z-1}{\Delta}\right)^{n-1} + \ldots + a_0}. \tag{3.4}$$

A continuación se presentan, de forma ilustrativa, las aproximaciones de sistemas de tiempo continuo de primer y segundo orden.

Sistema de primer orden.

Considere el siguiente sistema lineal de primer orden:

$$\tau \dot{y}(t) + y(t) = Ku(t), \tag{3.5}$$

con τ la constante de tiempo y K la ganancia a contínua del sistema[1]. La función de transferencia correspondiente viene dada por

$$G(s) = \frac{K}{\tau s + 1}. \tag{3.6}$$

El sistema discreto obtenido al aplicar el método de Euler para este sistema esta dado por

$$G_d(z) = \frac{K}{\tau \dfrac{z-1}{\Delta} + 1}. \tag{3.7}$$

Dado que $Y_d(z) = G_d(z)U_d(z)$, podemos usar la transformada $\mathcal{Z}$ inversa para obtener la ecuación

$$\tau \frac{y(n\Delta + \Delta) - y(n\Delta)}{\Delta} + y(n\Delta) = Ku(n\Delta), \tag{3.8}$$

resultado que es equivalente a aproximar la primera derivada de $y(t)$ al instante $t = n\Delta$ en (3.5). Note que esta ecuación puede ser re-escrita como

$$y(n\Delta + \Delta) = \frac{\Delta}{\tau}\left(Ku(n\Delta) - y(n\Delta)\right) + y(n\Delta), \tag{3.9}$$

o equivalentemente como una ecuación de diferencia (o recursiva) en términos del tiempo k como

$$y[k+1] = \frac{\Delta}{\tau}\left(Ku[k] - y[k]\right) + y[k] = \frac{\Delta}{\tau} Ku[k] + \frac{\tau - \Delta}{\tau} y[k], \tag{3.10}$$

donde $k = 1, 2, ...$ es el índice temporal normalizado, y por lo tanto no representa el tiempo absoluto sino que el tiempo relativo asociado a un tiempo de muestreo Δ.

Sistema de segundo orden.

Considere un sistema lineal de tiempo-continuo de segundo orden descrito por

$$G_c(s) = \frac{b_1 s + b_0}{a_2 s^2 + a_1 s + a_0} \tag{3.11}$$

Utilizando el método de Euler, el sistema puede aproximarse a través de:

$$\begin{aligned} a_2 \frac{y(n\Delta + 2\Delta) - 2y(n\Delta + \Delta) + y(n\Delta)}{\Delta^2} + a_1 \frac{y(n\Delta + \Delta) - y(n\Delta)}{\Delta} + a_0 y(n\Delta) \\ = b_1 \frac{u(n\Delta + \Delta) - u(n\Delta)}{\Delta} + b_0 u(n\Delta), \end{aligned}$$

[1] Recuerde que 4τ sirve como criterio para considerar que la respuesta del sistema paso su estado transiente.

o equivalentemente en términos del tiempo k:

$$a_2 \frac{y[k+2] - 2y[k+1] + y[k]}{\Delta^2} + a_1 \frac{y[k+1] - y[k]}{\Delta} + a_0 y[k] = b_1 \frac{u[k+1] - u[k]}{\Delta} + b_0 u[k].$$

Reordenando se puede obtener la ecuación de recurrencia de un sistema de segundo orden (complete usted los detalles). Note que, dado que el resultante es un sistema lineal e invariante en el tiempo, se puede hacer un desplazamiento en el tiempo para que el término $y[k+2]$ corresponda al término "actual", y los demás instantes correspondan al pasado. Esto último se puede hacer si las condiciones iniciales originales se desplazan también en el tiempo. En esta experiencia de laboratorio supondremos que las condiciones iniciales de los sistemas son nulas.

Simulación de sistemas dinámicos usando Simulink

En Simulink, es posible simular sistemás de tiempo continuo especificando el método numérico para aproximarlos. Para ello, debe abrir un nuevo modelo de Simulink, ir a preferences/configuration parameters/solver, donde podrá configurar en la sección *solver option* el método numérico que prefiera (la ventana mostrada en Figura 3.1). Existen métodos más sofisticados para resolver ecuaciones diferenciales e integrales numéricamente que el método de Euler. Muchos de ellos consideran un valor Δ variable en vez de fijo como lo descrito en la sección anterior.

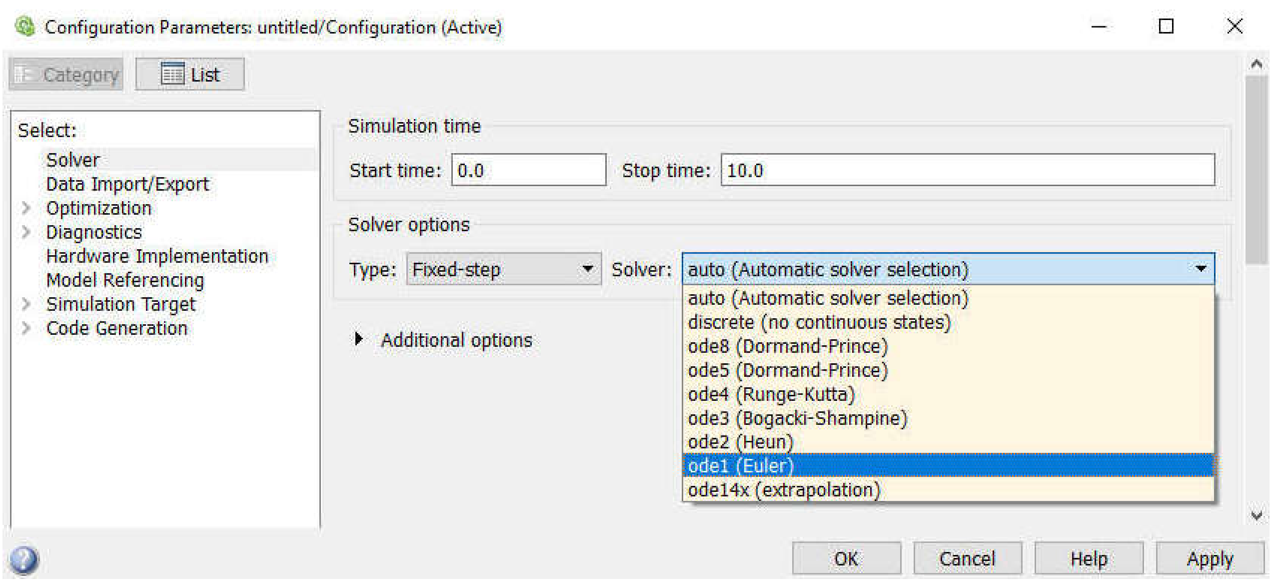

Figura 3.1: Pestaña *Configuration Parameteres*

Para configurar el método de Euler, debe escoger *Fixed-Step* en ves de *Variable-Step* en el apartado Type, y luego escoger la opción *ode1* en el apartado *Solver* (ver Figura 3.1). En el apartado *additional options* usted puede escoger el valor Δ que desee.

Para simular un sistema dinámico en Matlab usted puede implementar la ecuación diferencial asociada puede hacerlo con un bloque que representa la función de transferencia del sistema (para el caso lineal). Para ello vaya al Simulink Library Browser, y a la pestaña Simulink, donde encontrará varias pestañas que contienen diversos bloques con distintos propósitos (vea Figura 3.2).

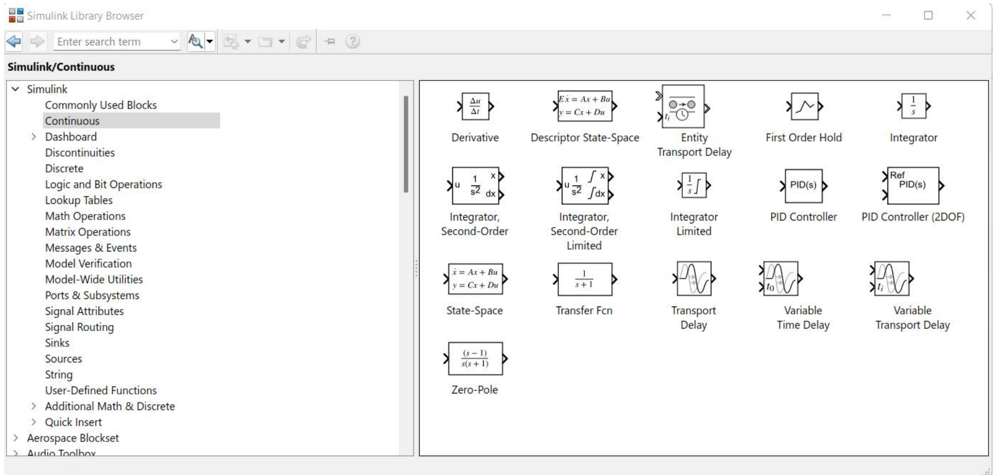

Figura 3.2: Library Browser: Contenido pestaña *continuous*

Por ejemplo, si desea obtener de forma simulada la respuesta a escalón del sistema lineal descrito por la ecuación diferencial

$$\ddot{y}(t) + a_1\dot{y}(t) + a_0 y(t) = u(t),$$

usted puede obtenerla implementando el esquema de la Figura 3.3, donde se han especificado las pestaña donde encontrar los elementos correspondientes. Si desea hacerlo usando la función de transferencia del sistema, puede implementar el modelo de la Figura 3.4.

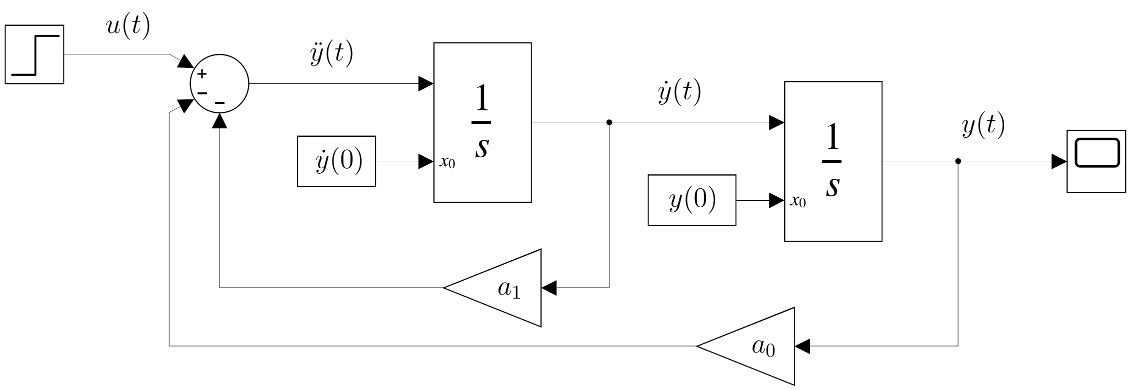

Figura 3.3: Simulación sistema de segundo orden usando la ecuación diferencial

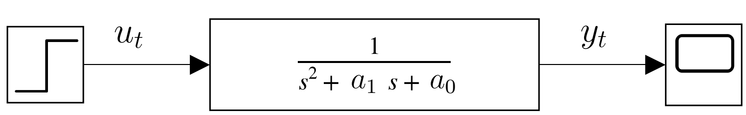

Figura 3.4: Simulación sistema de segundo orden usando función de transferencia

Ambos esquemas simulan lo mismo, teniendo ambos como respuesta lo mostrado por el osciloscopio (scope) en la Figura 3.5. Es importante indicar que los osciloscopios entregan por defecto una imagen con fondo negro. **En las imágenes de los informes usted debe cambiar dicho fondo negro** por uno blanco (similar a lo hecho en la Figura 3.5), para así apreciar de forma clara las curvas (y para que también se pueda imprimir el informe -si es necesario- sin gastar tinta innecesariamente).

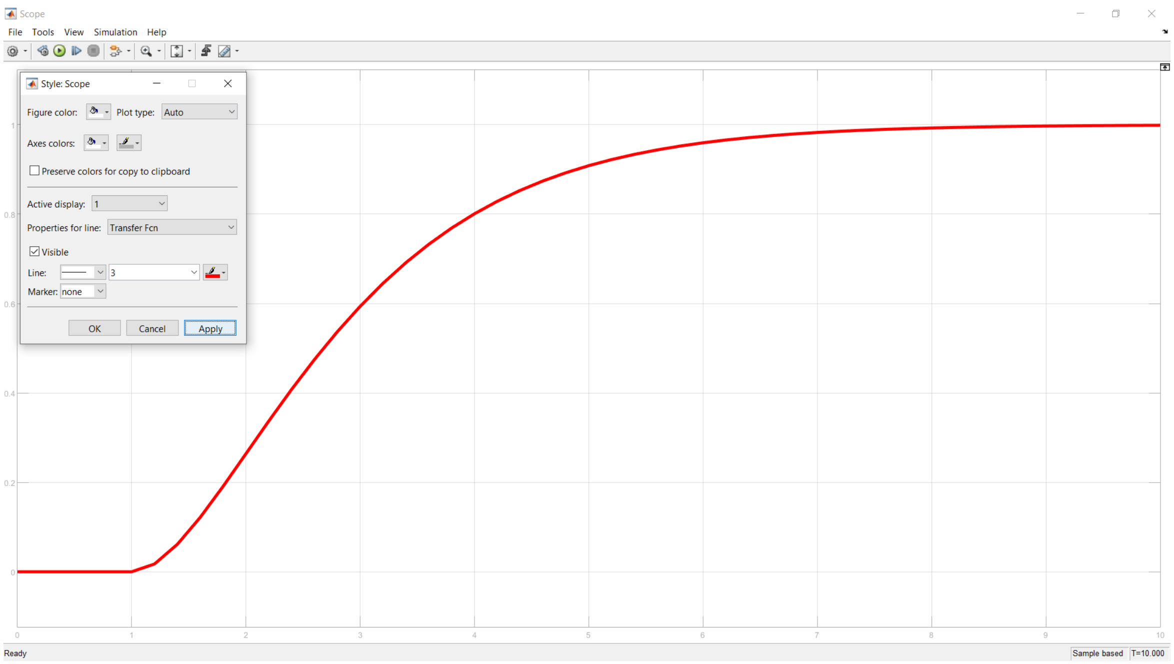

Figura 3.5: Respuesta a escalón del sistema de primer orden

Si el sistema a simular tiene ceros, entonces la metodología anterior tiene que modificarse de la siguiente manera. Considere el sistema

$$G_c(s) = \frac{Y(s)}{U(s)} = \frac{b_1 s + b_0}{s^2 + a_1 s + a_0} \tag{3.12}$$

Luego para simular este sistema se define una nueva variable $W(s)$ de la siguiente manera

$$W(s) = \frac{1}{s^2 + a_1 s + a_0} U(s) \tag{3.13}$$

con lo cual la salida del sistema se puede reescribir como

$$Y(s) = (b_1 s + b_0) W(s). \tag{3.14}$$

La simulación se puede obtener como en la figura 3.6. Note que el sub-sistema en (3.13) es el mismo tratado en el caso anterior, es decir, hay que modificar la simulación anterior para incluir los ceros. Además como se observa en la figura 3.6 se requiere el conocimiento de las condiciones iniciales para la variable $w(t)$ y su derivada. Estos valores se obtienen de la siguiente manera:

$$y(0) = b_1 \dot{w}(0) + b_0 w(0) \tag{3.15}$$

$$\dot{y}(0) = b_1 \ddot{w}(0) + b_0 \dot{w}(0) \tag{3.16}$$

Se obtiene de (3.13) que $\ddot{w}(0) = u(0) - a_1 \dot{w}(0) - a_0 w(0)$ con lo cual:

$$\dot{y}(0) = b_1 u(0) - b_1 a_1 \dot{w}(0) - b_1 a_0 w(0) + b_0 \dot{w}(0) \tag{3.17}$$

luego de (3.15) y (3.17) se resuelve un sistema de ecuaciones cuya solución son las condiciones iniciales para la variable $w(t)$ y $\dot{w}(t)$, así

$$\begin{bmatrix} w(0) \\ \dot{w}(0) \end{bmatrix} = \begin{bmatrix} b_0 & b_1 \\ -b_1 a_0 & b_0 - b_1 a_1 \end{bmatrix}^{-1} \begin{bmatrix} y(0) \\ \dot{y}(0) - b_1 u(0) \end{bmatrix} \tag{3.18}$$

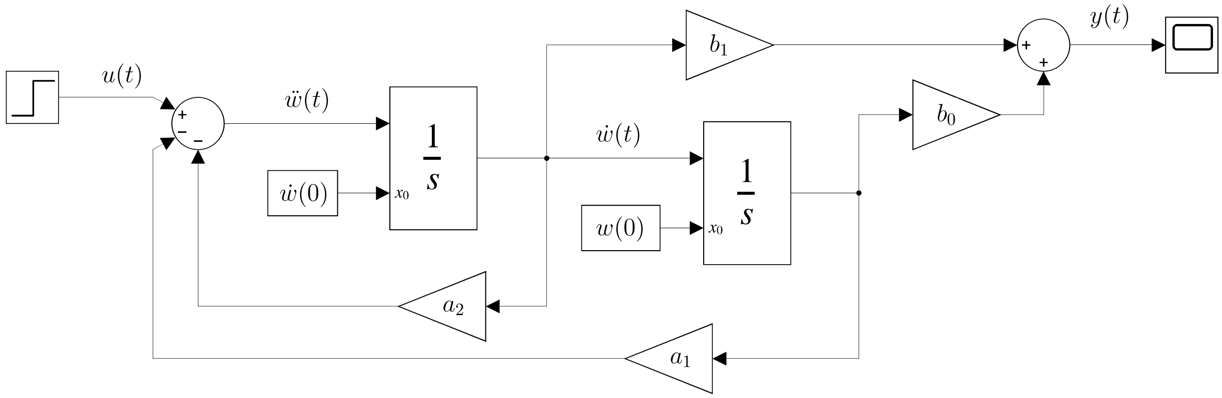

Figura 3.6: Simulación sistema de segundo orden usando la ecuación diferencial

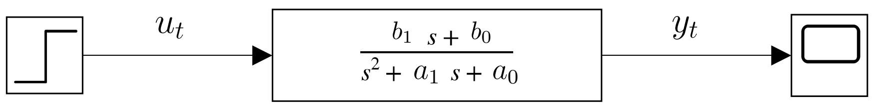

Figura 3.7: Simulación sistema de segundo orden usando función de transferencia

Note que este procedimiento es general y puede ser aplicado a la planta (3.3) para cualquier cantidad de polos y ceros.

3.2. Método de mínimos cuadrados

En esta sección introduciremos un algoritmo, muy usado en problemas de ingeniería, que permite estimar parámetros desconocidos a partir de mediciones. Este algoritmo debe su nombre al hecho que es el resultado de un problema de optimización que busca minimizar el cuadrado de una función de costo que representa el error de estimación.
En esta experiencia, nuestro propósito es usar Matlab para implementar dicho algoritmo, sin embargo en una futura experiencia, este algoritmo será de utilidad para estimar características de los sistemas reales con los que trabajará.

Para introducir el método de mínimos cuadrados, consideraremos un problema donde usted tiene un conjunto de datos experimentales, graficados en la Figura 3.8 (donde x e y representan variables asociadas al experimento), y desea obtener una función analítica que se aproxime a los datos medidos.

Podríamos, por ejemplo, buscar la recta que mejor se adecúe a los datos experimentales obtenidos, como se ilustra en la Figura 3.9. El problema entonces se transforma en encontrar los parámetros a y b tal que la recta descrita por $y = ax + b$ (la curva roja en la Figura 3.9) sea lo más cercana posible a los datos experimentales disponibles.

Para presentar la solución, considere que los datos experimentales se ordenan de a pares denotados

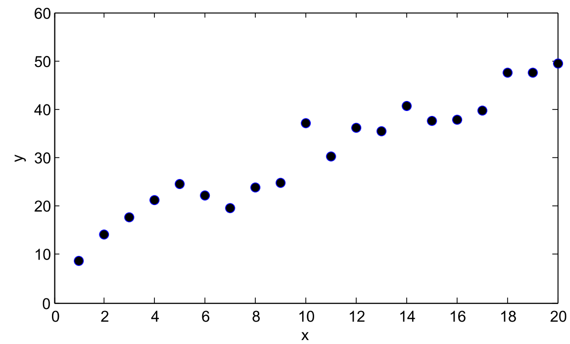

Figura 3.8: Gráfica de datos experimentales

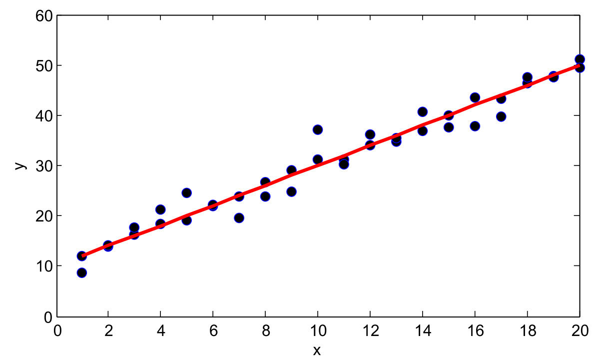

Figura 3.9: Recta que se aproxima a los datos experimentales

por (y_i, x_i), y defina

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \Phi = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}, \quad \hat{\theta} = \begin{bmatrix} b \\ a \end{bmatrix} \tag{3.19}$$

Note que si los datos experimentales describieran una recta exacta, existiría un vector $\hat{\theta}$ tal que $Y = \Phi\hat{\theta}$, sin embargo esto no sucederá en general pues existe una diferencia entre los datos reales y la recta propuesta.

El método de mínimos cuadrados nos dice que los parámetros se pueden aproximar como

$$\hat{\theta} = (\Phi^T \Phi)^{-1} \Phi^T Y, \tag{3.20}$$

donde $(\cdot)^T$ denota la transpuesta del argumento.(La demostración del resultado anterior no es parte de los contenidos de este curso, sin embargo puede consultar la bibliogría correspondiente si desea saber más.)

El resultado anterior se puede generalizar para estudiar curvas de orden superior. También estos resultados se pueden asociar a la estimación de parámetros de sistemas dinámicos. Para ello, considere una señal escalar en tiempo discreto $y[k]$ la cuál ha sido generada por la siguiente relación:

$$y[k] = \sum_{i=1}^{n} \varphi_i[k]\theta_i + w[k], \tag{3.21}$$

donde $\varphi_i[k]$ son señales conocidas, θ_i son parámetros desconocidos y $w[k]$ es una señal de ruido de medición. En este caso nos interesa encontrar el vector $\hat{\theta}$ que mejor se aproxime al vector θ definido como

$$\theta = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_n \end{bmatrix} \tag{3.22}$$

El estimador de mínimos cuadrados, $\hat{\theta}$ se obtiene a partir de la colección de datos $y[k]$ y $\varphi_i[k]$ para $k = 1 \ldots N$, tiene la misma forma anterior, es decir,

$$\hat{\theta} = (\Phi^T\Phi)^{-1}\Phi^T Y, \tag{3.23}$$

pero con la consideración que

$$Y = \begin{bmatrix} y[1] \\ \vdots \\ y[N] \end{bmatrix}, \quad \Phi = \begin{bmatrix} \varphi_1[1] & \cdots & \varphi_n[1] \\ \vdots & \ddots & \vdots \\ \varphi_1[N] & \cdots & \varphi_n[N] \end{bmatrix}. \tag{3.24}$$

4. Trabajo de Laboratorio

La presente sección indica el trabajo que debe realizar en las dos sesiones de laboratorio. Esta consta de dos partes: la primera busca practicar las operaciones básicas de Matlab a través de la implementación del algoritmo de mínimos cuadrados, mientras que la segunda busca estudiar la simulación y análisis de sistemas dinámicos (incluida la aproximación de Euler). Note que debe analizar los resultados obtenidos, por ejemplo, para la segunda parte debe hacer las actividades que se piden pero también discutir sus resultados (que características del sistema influyen en las distintas respuestas que observará (polos, ceros, factor de amortiguamiento, etc.).

Recuerde que el informe que usted redacte debe incluir todos los códigos de MATLAB , SIMULINK que utilizó. Se puede entregar tanto el informe (recuerde que el informe debe ser en formato pdf) como los códigos de MATLAB y SIMULINK en formato digital.

4.1. Sesión 1: Algoritmo de mínimos cuadrados

En esta experiencia se implementará el método de mínimos cuadrados para dos casos, y luego se evaluará el efecto de usar otro método numérico distinto al de Euler.

Actividad 1: Datos de un plano

1. Implemente en SIMULINK el siguiente sistema

$$y(t) = a_1 u_1(t) + a_2 u_2(t) + w(t) \tag{4.1}$$

   donde $a_1 = 1$, $a_2 = 2$, $u_1(t)$ es un escalón en $t = 1$, $u_2(t)$ es una sinusoide de amplitud 1 y frecuencia $\omega_0 = 10$, y $w(t)$ es una señal de ruido blanco Gaussiano de varianza 0.1 (pregúntele a su profesor qué es el ruido Gaussiano si no está familiarizado con este concepto).

2. Simule de $t = 0$ hasta $t = 10[s]$, utilizando el método de Euler con $\Delta = 10^{-1}[s]$ y tome datos cada $0.5[s]$. Denote como $y[k]$, $u_1[k]$, $u_2[k]$ para ($k = 1 \dots N$, índice temporal normalizado) los datos que ha recolectado.

3. Desarrolle un programa de MATLAB para estimar los valores de a_1 y a_2 utilizando el método de mínimos cuadrados. Note que en este caso se tiene que la matriz Φ esta dada por

$$\Phi = \begin{bmatrix} u_1[1] & u_2[1] \\ u_1[2] & u_2[2] \\ \vdots & \vdots \\ u_1[N] & u_2[N] \end{bmatrix} \tag{4.2}$$

Actividad 2: Datos de una función cuadrática

1. Implemente en SIMULINK el siguiente sistema Considere la misma configuración de la actividad anterior, con las mismas entradas y ruido, pero ahora suponga que los datos obedecen la ecuación

$$y(t) = a_0 + a_1 u_1(t) + a_2 u_2(t)^2 + w(t) \tag{4.3}$$

   donde $a_0 = 3$, $a_1 = 2$, $a_2 = 1$, $u_1(t)$ es un escalón en $t = 1$, $u_2(t)$ es una sinusoide de amplitud 1 y frecuencia $\omega_0 = 10$, y $w(t)$ es una señal de ruido blanco Gaussiano de varianza 0.1.

2. Simule de $t = 0$ hasta $t = 10[s]$, utilizando el método de Euler con $\Delta = 10^{-1}[s]$ y tome datos cada $0.5[s]$. Denote como $y[k]$, $u_1[k]$, $u_2[k]$ para ($k = 1 \dots N$, índice temporal normalizado) los datos que ha recolectado.

3. Desarrolle un programa de MATLAB para estimar los valores de a_0, a_1 y a_2 utilizando el método de mínimos cuadrados. Note que en este caso se tiene que la matriz Φ esta dada por

$$\Phi = \begin{bmatrix} 1 & u_1[1] & u_2[1]^2 \\ 1 & u_1[2] & u_2[2]^2 \\ \vdots & \vdots & \vdots \\ 1 & u_1[N] & u_2[N]^2 \end{bmatrix} \tag{4.4}$$

Actividad 3: Efecto del método numérico:

Considere los sistemas

$$y(t) = a_1 u_1(t) + a_2 \sqrt{u_2(t)} \tag{4.5}$$

$$\frac{dy(t)}{dt} = a_1 u_1(t) + a_2 \sqrt{u_2(t)}, \qquad y(0^-) = 0 \tag{4.6}$$

donde a_i son parámetros constantes, u_i corresponden a entradas conocidas al sistema e y corresponde a la salida.

1. Discuta qué tipo de sistemas son los descritos en (4.5) y (4.6) (dinámico o algebraico, lineal o no lineal, invariante o variante en el tiempo, causal o no causal.)

2. Considere $a_1 = 2$, $a_2 = 1$, $u_1(t) = \sin(100t)$, $u_2(t) = (1.5\sin(10t) + 2)\mu(t)$, donde $\mu(t)$ es la señal escalón unitario. Simule utilizando SIMULINK ambos sistemas (4.5) y (4.6) considerando el método de Euler con $\Delta = 10^{-2}$, y tiempo de simulación de 10 segundos. Exporte a Workspace las salidas de ambos sistemas.

3. Repita el punto anterior pero ahora considerando otro método numérico, por ejemplo el método de Heun (ode2), conservando $\Delta = 10^{-2}$. Exporte a Workspace las salidas de ambos sistemas (use nombres distintos al punto anterior para no sobreescribir los datos).

4. Escriba un código que compare la salida del sistema (4.5) usando Euler y usando el otro método. Son iguales las salidas? Haga la misma comparación pero ahora considerando la salida del sistema (4.6). Comente.

5. Considerando lo observado en el punto anterior, cree que hubiesen cambiado las estimaciones hechas en las actividades anteriores si se hubiese usado otro método numérico en vez del método de Euler? Justifique.

4.2. Sesión 2: Simulación de sistemas dinámicos

El trabajo que debe desarrollar en esta sección consiste en **simular y analizar** la respuesta de sistemás dinámicos cuando son excitados por distintas entradas. Durante el desarrollo se aplicará el método de Euler con $\Delta = 10^{-3}$.

Actividad 1: Sistema de primer orden.

Considere el sistema en tiempo-continuo representado en función de transferencia como

$$G_c(s) = \frac{b}{s+a} \tag{4.7}$$

donde los parámetros para a y b tienen 3 posibles valores cada uno, descritos en la Tabla 4.1

Caso	a	b
(I)	1	1
(II)	10	10
(III)	100	100

Cuadro 4.1: Valores para los parámetros del sistema de primer orden

Respuesta escalón

Desarrolle lo siguiente para todos los casos descritos anteriormente:

1. Obtenga una expresión analítica para la respuesta a escalón (use las tablas al final del documento). Considere un escalón dado por:

$$u(t) = \begin{cases} 0 & t < 1 \\ 1 & t \geq 1 \end{cases} \tag{4.8}$$

   Use la expresión analítica para graficar la respuesta a escalón usando el comando *plot* hasta un tiempo $t = t_{final} = 10$. Para ello construya un vector de tiempo espaciado cada $\Delta = 10^{-3}$ segundos. Guarde los datos.

2. Determine la ecuación de recurrencia que se utilizaría para simular los sistemas dinámicos de primer orden cuando se utiliza el método de Euler para Δ general.

3. Utilice SIMULINK para obtener la respuesta a escalón y exporte los resultados al workspace de MATLAB . Para ello implemente la ecuación diferencial correspondiente y también la función de transferencia, y verifique que ambos esquemas entregan la misma respuesta. Utilice el método de Euler con $\Delta = 10^{-3}$.

4. Compare gráficamente usando el comando *plot* y *hold on* los resultados obtenidos por el método Euler (datos guardados desde Simulink) con la respuesta exacta (analítica). Mida además el error cuadrático medio, dado por

$$E = \frac{1}{N}\sum_{n=0}^{N-1}(y_{exacta}(n\Delta) - y_{Euler}(n\Delta))^2 \tag{4.9}$$

   donde $(N-1)\Delta = t_{final}$.

Actividad 2: Sistema de segundo orden.

Considere un segundo sistema descrito por

$$G_c(s) = \frac{b_1 s + b_0}{s^2 + a_1 s + a_0} = \frac{b_1 s + b_0}{s^2 + 2\xi\omega_n s + \omega_n^2}, \tag{4.10}$$

donde ξ es el coeficiente de amortiguamiento y ω_n es la frecuencia natural no amortiguada del sistema. Supondremos distintos casos según los valores definidos en la Tabla 4.2

Caso	a_1	a_0	b_1	b_0
(I)	2*0.7*1	1	0	1
(II)	2*0.3*1	1	0	1
(III)	2*0.7*10	100	0	1
(IV)	2*0.3*10	100	0	1
(V)	2*0.7*1	1	-1	1

Cuadro 4.2: Valores para los parámetros del sistema de segundo orden

Respuesta Impulso y Escalón

Desarrolle lo siguiente para todos los casos descritos anteriormente:

1. Determine los polos y ceros del sistema usando los comandos pole y zero.
2. Compare gráficamente y analice, usando el comando *plot* y *hold on*, la respuesta impulso (use *impulse*) de los sistemas en los casos I-II, y a parte los casos III-IV (use el mismo tiempo de simulación para comparar).
3. Compare gráficamente y analice, usando el comando *subplot*, la respuesta a escalón de todos los casos (use *step*, use el mismo tiempo de simulación por defecto dado por el comando).
4. Guarde los datos obtenidos usando el comando *step*. Determine, usando indexación (acceder a los elementos de las variables usando cierto criterio): la salida estacionaria, el sobrepaso máximo (overshoot), el tiempo de peak, la máxima contrarrespuesta (undershoot) y el tiempo valle, según corresponda.

Respuesta en Frecuencia

En esta sección considere sólo el sistema de segundo orden Caso (I) (en la tabla 4.2) y las señales de entrada dadas por

$$u_1(t) = 10sin(3t)\mu(t) \tag{4.11}$$
$$u_2(t) = 10\left(1 + sin(3t)\right)\mu(t) \tag{4.12}$$
$$u_3(t) = 10sin(10t)\mu(t) \tag{4.13}$$

donde $\mu(t)$ es la señal escalón unitario.

Desarrolle lo siguiente:

1. Grafíque el diagrama de Bode del sistema utilizando MATLAB .
2. Cuál es la respuesta en estado estacionario para las entradas $u_1(t)$, $u_2(t)$ y $u_3(t)$? Explique cómo se puede obtener respuesta a estado estacionario usando el diagrama de Bode, y usando el comando *freqresp*.
3. Utilice SIMULINK para obtener la respuesta del sistema para las entradas $u_1(t)$, $u_2(t)$ y $u_3(t)$. Exporte los resultados al workspace de MATLAB y compare gráficamente. Para la simulación utilice el método de Euler con $\Delta = 10^{-3}$ seg. Use un tiempo de simulación de 30s.
4. Repita las simulaciones del punto anterior pero ahora considere $\Delta = 0.5$ seg. y compare gráficamente (en Matlab) con los resultados obtenidos para $\Delta = 10^{-3}$ seg.

5. Comentarios anexos

- Existen diversos tutoriales en internet sobre MATLAB/Simulink. A la fecha, se pueden encontrar en línea los siguientes tutoriales en español:

  - https://www.youtube.com/channel/UCzLFVTOkvComY_eApFUfrTA
  - https://www.youtube.com/watch?v=UawhvugB9qs
  - https://www.youtube.com/watch?v=j4rRvelnxso

- Más información de métodos numéricos y su aplicación a la simulación de sistemas dinámicos se puede encontrar en la siguientes libros (entre otros):

[1] R. L. Burden. Análisis Numérico. Thomson Editores, 2002.

[2] J. C. Butcher. Numerical methods for ordinary differential equations. Wiley, 2nd edition, 2008.

[3] S. Nakamura. Análisis numérico y visualizacion gráfica con MATLAB. Pearson, 1997.

- El algoritmo de mínimos cuadrados es ampliamente utilizado en el área de identificación de sistemas dinámicos y tiene aplicación en sistemas de control avanzado (adaptativo) y procesamiento de señales. Algunos libros que tratan estos temas son los siguientes:

[4] L. Ljung. System Identification: Theory for the user. Prentice Hall, 2nd edition, 1999.

[5] T. Söderström and P. Stoica. System Identification. Prentice-Hall International, 1989.

[6] G. C. Goodwin and R. Payne. Dynamic System Identification: Experiment design and data analysis. Academic Press, 1977.

$f(t)$	$\mathcal{L}\{f(t)\} = F(s)$	Descripción
$\sum_{i=1}^{l} a_i y_i(t)$	$\sum_{i=1}^{l} a_i Y_i(s)$	Linealidad
$y(at) \quad (a > 0)$	$\frac{1}{a} Y\left(\frac{s}{a}\right)$	Escalamiento
$\frac{dy(t)}{dt}$	$sY(s) - y(0^-)$	Derivada en t
$\frac{d^k y(t)}{dt^k} \quad (k \in \mathbb{N})$	$s^k Y(s) - \sum_{\ell=1}^{k} s^{k-\ell} \left. \frac{d^{\ell-1} y}{dt^{\ell-1}} \right\rvert_{t=0^-}$	Derivada en t de orden superior
$\int_{0^-}^{t} y(\tau) d\tau$	$\frac{1}{s} Y(s)$	Integral definida
$t\, y(t)$	$-\frac{dY(s)}{ds}$	Derivada en s
$t^k y(t) \quad (k \in \mathbb{N})$	$(-1)^k \frac{d^k Y(s)}{ds^k}$	Derivada en s de orden superior
$y(t-\tau)\mu(t-\tau)$	$e^{-s\tau} Y(s)$	Desplazamiento en el tiempo t
$e^{at} y(t)$	$Y(s-a)$	Desplazamiento en la variable s
$\int_{0^-}^{t} y_1(\tau) y_2(t-\tau) d\tau$	$Y_1(s) Y_2(s)$	Convolución de funciones causales
$y_1(t) y_2(t)$	$\frac{1}{2\pi j} \int_{\sigma - j\infty}^{\sigma + j\infty} Y_1(\zeta) Y_2(s-\zeta) d\zeta$	Producto en el tiempo
$\lim_{t \to \infty} y(t)$	$\lim_{s \to 0} sY(s)$	Teorema del Valor Final
$\lim_{t \to 0^+} y(t)$	$\lim_{s \to \infty} sY(s)$	Teorema del Valor Inicial

$f(t) \quad (t \geq 0)$	$\mathcal{L}\{f(t)\} = F(s)$	Región de convergencia
$\delta(t)$	1	$\forall s \in \mathbb{C}$
$1 \quad (= \mu(t))$	$\dfrac{1}{s}$	$\Re\{s\} > 0$
$\mu(t-\tau) \quad (\tau > 0)$	$\dfrac{e^{-s\tau}}{s}$	$\Re\{s\} > 0$
t	$\dfrac{1}{s^2}$	$\Re\{s\} > 0$
$t^n \quad (n \in \mathbb{N})$	$\dfrac{n!}{s^{n+1}}$	$\Re\{s\} > 0$
$e^{\alpha t} \quad (\alpha \in \mathbb{C})$	$\dfrac{1}{s-\alpha}$	$\Re\{s\} > \Re\{\alpha\}$
$t\, e^{\alpha t} \quad (\alpha \in \mathbb{C})$	$\dfrac{1}{(s-\alpha)^2}$	$\Re\{s\} > \Re\{\alpha\}$
$\cos(\omega_o t)$	$\dfrac{s}{s^2+\omega_o^2}$	$\Re\{s\} > 0$
$\operatorname{sen}(\omega_o t)$	$\dfrac{\omega_o}{s^2+\omega_o^2}$	$\Re\{s\} > 0$
$e^{\alpha t}\cos(\omega_o t)$	$\dfrac{s-\alpha}{(s-\alpha)^2+\omega_o^2}$	$\Re\{s\} > \Re\{\alpha\}$
$e^{\alpha t}\operatorname{sen}(\omega_o t)$	$\dfrac{\omega_o}{(s-\alpha)^2+\omega_o^2}$	$\Re\{s\} > \Re\{\alpha\}$
$t\operatorname{sen}(\omega_o t)$	$\dfrac{2\omega_o s}{(s^2+\omega_o^2)^2}$	$\Re\{s\} > 0$
$t\cos(\omega_o t)$	$\dfrac{s^2-\omega_o^2}{(s^2+\omega_o^2)^2}$	$\Re\{s\} > 0$
$\mu(t)-\mu(t-\tau)$	$\dfrac{1-e^{-s\tau}}{s}$	$\forall s \in \mathbb{C}$

$y[t]$	$Y[z] = \mathcal{Z}\{y[t]\}$	Descripción
$\sum_{i=1}^{l} a_i y_i[t]$	$\sum_{i=1}^{l} a_i Y_i[z]$	Linealidad
$\alpha^t y[t] \quad (\alpha \in \mathbb{C})$	$Y\left[\frac{z}{\alpha}\right]$	Escalamiento en z
$y^*[t]$	$Y^*[z^*]$	Conjugación
$\sum_{\ell=0}^{t} y[\ell]$	$\frac{z}{z-1}Y[z]$	Suma acumulada
$y[t-t_0] \quad (t_0 \in \mathbb{N})$	$z^{-t_0}\left(Y[z] + \sum_{\ell=1}^{t_0} y[-\ell]z^{\ell}\right)$	Retardo en t
$y[t+t_0] \quad (t_0 \in \mathbb{N})$	$z^{t_0}\left(Y[z] - \sum_{\ell=0}^{t_0-1} y[\ell]z^{-\ell}\right)$	Adelanto en t
$y[t-t_0]\mu[t-t_0]$	$z^{-t_0}Y[z]$	Retardo secuencia causal
$ty[t]$	$-z\frac{dY[z]}{dz}$	Derivada en z
$t^2y[t]$	$z^2\frac{d^2Y[z]}{dz^2} + z\frac{dY[z]}{dz}$	Derivada de segundo orden en z
$\sum_{l=0}^{t} y_1[l]y_2[t-l]$	$Y_1[z]Y_2[z]$	Convolución de funciones causales
$y_1[t]y_2[t]$	$\frac{1}{2\pi j}\oint_{\Gamma_1} Y_1[\xi]Y_2\left[\frac{z}{\xi}\right]\xi^{-1}\,d\xi$	Producto en el tiempo
$y[0]$	$\lim_{z\to\infty} Y[z]$	Teorema del Valor Inicial
$\lim_{t\to\infty} y[t]$	$\lim_{z\to 1}(z-1)Y[z]$	Teorema del Valor Final

$y[t] \quad (t \in \mathbb{Z}_0^+)$	$Y[z] = \mathcal{Z}\{y[t]\}$	Región de Convergencia
$\delta[t]$	1	$\|z\| > 0$
$1 \quad (= \mu[t])$	$\dfrac{z}{z-1}$	$\|z\| > 1$
$\mu[t-t_0] \quad (t_0 \in \mathbb{N})$	$\dfrac{z^{t_0+1}}{z-1}$	$\|z\| > 1$
t	$\dfrac{z}{(z-1)^2}$	$\|z\| > 1$
$\alpha^t \quad (\alpha \in \mathbb{C})$	$\dfrac{z}{z-\alpha}$	$\|z\| > \|\alpha\|$
$t\,\alpha^t \quad (\alpha \in \mathbb{C})$	$\dfrac{\alpha z}{(z-\alpha)^2}$	$\|z\| > \|\alpha\|$
$e^{j\theta_0 t}$	$\dfrac{z}{z-e^{j\theta_0}}$	$\|z\| > 1$
$\cos(\theta_0 t)$	$\dfrac{z(z-\cos\theta_0)}{z^2-2z\cos\theta_0+1}$	$\|z\| > 1$
$\operatorname{sen}(\theta_0 t)$	$\dfrac{z\operatorname{sen}\theta_0}{z^2-2z\cos\theta_0+1}$	$\|z\| > 1$
$\alpha^t\cos(\theta_0 t)$	$\dfrac{z(z-\alpha\cos\theta_0)}{z^2-2z\alpha\cos\theta_0+\alpha^2}$	$\|z\| > \|\alpha\|$
$\alpha^t\operatorname{sen}(\theta_0 t)$	$\dfrac{z\alpha\operatorname{sen}\theta_0}{z^2-2z\alpha\cos\theta_0+\alpha^2}$	$\|z\| > \|\alpha\|$
$\cos(\theta_0 t+\phi)$	$\dfrac{z^2\cos\phi - z\cos\phi\cos\theta_0 + \operatorname{sen}\phi\operatorname{sen}\theta_0}{z^2-2z\cos\theta_0+1}$	$\|z\| > 1$
$\begin{cases} \alpha^t & ; 0 \leq t < N \\ 0 & ; t \geq N \end{cases}$	$\dfrac{1-(\alpha z^{-1})^N}{1-\alpha z^{-1}}$	$\|z\| > 0$