

Contenido

I	Introducción a espacio de estados - English	2
1.	Introduction	3
1.1.	Control theory	3
1.1.1.	Basics	3
1.1.2.	State space	3
II	Control Bootcamp - Castellano	6
2.	Sistemas lineales	7
3.	Estabilidad y autovalores	10
3.1.	Evolución discreta	10
3.2.	Linealizando un sistema	11
4.	Controlabilidad	13
4.1.	Grados de controlabilidad y gramianes	13
4.2.	Posicionamiento de autovalores (polos)	14
5.	Observabilidad	16
5.1.	Estimación de estado completo	17
6.	Control lineal cuadrático	19
6.1.	Filtro Kalman	19
6.2.	Control Lineal Cuadrático Gaussiano	21
6.3.	Regulador lineal cuadrático	21
6.4.	Ejemplo de Carro con péndulo	22
6.4.1.	Descripción del sistema	22
7.	Anexo	24
7.1.	Métodos numéricos para resolución de SEDO	24
7.1.1.	Runge–Kutta orden 4 multivariable	24

Parte I

Introducción a espacio de estados - English

Capítulo 1

Introduction

1.1. Control theory

1.1.1. Basics

Taylor expansion (Linearization) of two-variable nonlinear equation.

$$f(x, y) = f(\bar{x}, \bar{y}) + \left[\frac{\partial f}{\partial x}(x - \bar{x}) + \frac{\partial f}{\partial y}(y - \bar{y}) \right] + \dots$$

Matlab command to convert state space to transfer function `[num,den]=ss2tf(A,B,C,D,iu)` where `iu` must be specified for systems with more than one input.

1.1.2. State space

$\mathbf{x}(t) \in \mathbb{R}^n$ is the state space vector and is of size $n \times 1$ for a system of n number of state variables.

$\mathbf{u}(t) \in \mathbb{R}^p$ is inputs vector and is of size $p \times 1$ for a given control system, i.e: $\mathbf{u}(t) = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$ for two input system, $p = 2$.

$\mathbf{y}(t) \in \mathbb{R}^q$ is the output vector of size $q \times 1$.

$\mathbf{z}(t) \in \mathbb{R}^r$ is the *disturbance input*. Only applies to dynamical systems and is of size $r \times 1$

Thus we define the **state space variables** so that system output is purely in function of current system state variables and input variables.

$$\text{System Output} = f(\text{Current System State, System Input})$$

We will define X or \mathbf{x} as our system state variables. There are some important aspects to note about state space variables such as

- System output $\mathbf{y}(t)$ will be a function of them
- They change over time
- They are internal to the system
- They may include system outputs (outputs will be a function of themselves in part)
- Their selection is inherent part of the system design process and there are different methods of selecting them.
- We will assume there is a minimal quantity of state variables that is sufficient to accurately describe the system

- If all system inputs u_j are defined beforehand for $t \geq t_0$ then $\mathbf{x}(t)$ defines all system states for time $t \geq t_0$

The mathematical representation of state space variables will be that of the **state vector** $\mathbf{x}(t)$ of size $n \times 1$.

To model our system we then define the equations that govern it in **state space**¹. These are the **state-space equations** of the system. For a dynamic system these must include a variable that serves as memory of inputs for $t \geq t_1$. *Integrators* serve as memory devices for *continuous-time* models, however, our state-space representation is discrete! This is when state-space variables come in handy: The outputs of integrators can be considered as the variables that define the internal state of the dynamic system (Ogata).

For a system of size $p = q = n = 1$ one has the state-space representation defined as:

$$\dot{x}(t) = g[t_0, t, x(t), x(0), u(t)], \quad y = h[t, x(t), u(t)]$$

For a *linear time-variant dynamical system* of arbitrary size it is convenient to represent it in its linearized form

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) + \mathbf{E}(t)\mathbf{z}(t) \quad (1.1)$$

$$\mathbf{y}(t) = \mathbf{C}(t)\mathbf{x}(t) + \mathbf{D}(t)\mathbf{u}(t) \quad (1.2)$$

where

$\mathbf{A}_{n \times n}$ System matrix. Relates future state change with current state. May be zero. Also referred to as the state matrix in some bibliographies.

$\mathbf{B}_{n \times p}$ Control/input matrix. How system input influences state change. May be zero.

$\mathbf{C}_{q \times n}$ Output matrix. How system state influences system output.

$\mathbf{D}_{q \times p}$ Feed forward or feedthrough matrix. How system input influences system output. Is usually zero for most physical systems.

$\mathbf{E}_{n \times r}$ Input matrix for disturbances. Applies only for dynamical systems.

the system is said to be **time-invariant** if the above matrices are not dependent of time. An example of a **time-variant** system is a spacecraft, whose mass changes due to fuel consumption.

One method of state space variable selection is **physical selection**. This method is based on energy accumulators. It can be said that *the minimum number of state-space variables needed to model the system accurately is equal to the number of independent energy accumulators*. When state-space variable is not a energy variable it is said to be an augmented variable.

The general solution to the linear differential equation of state:

$$\dot{\mathbf{x}}(t) = \mathbf{A} \mathbf{x}(t) + \mathbf{B} \mathbf{u}(t)$$

es

$$\mathbf{x}(t) = e^{\mathbf{A}(t-t_0)}\mathbf{x}(0) + \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B} \mathbf{u}(\tau)d\tau$$

¹State space can be thought of an n -dimensional space whose axes are the state variables ($x_1, x_2 \dots$)

Definition 1.1 (Matrix Exponential).

$$e^{\mathbf{A}t} = \mathbf{I} + \sum_{i=1}^{\infty} \frac{(\mathbf{A}t)^i}{i!} \quad (1.3)$$

A MATLAB function is provided to calculate the matrix exponential.

CÓDIGO 1.1: matrixexponential.m

```
A = rand(3);  
t0 = 0.5;  
fprintf('e^(At)='); disp(expt(t0,1e5,A));  
function y = expt(t,n,A)  
    y = eye(size(A));  
    for i=1:n  
        y = y + (A*t)^i/factorial(i);  
    end  
end
```

Parte II

Control Bootcamp - Castellano

Capítulo 2

Sistemas lineales

El planteo de la evolución de un sistema de primer orden es dado por (2.1). Su solución analítica requiere evaluar exponencial de la matriz \mathbf{A}

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) + \mathbf{w}_d \quad (2.1)$$

$$\mathbf{y}(t) = \mathbf{C}(t)\mathbf{x}(t) + \mathbf{D}(t)\mathbf{u}(t) + \mathbf{w}_n \quad (2.2)$$

donde $\mathbf{w}_d = \mathbf{E}(t)\mathbf{z}(t)$ son las perturbaciones sobre el sistema y \mathbf{w}_n es el ruido en la medición.

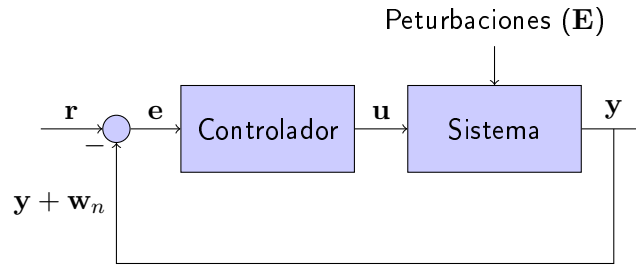


FIGURA 2.1: Esquema de un sistema de control genérico.

La exponencial de una matriz (2.3) es poco práctica para calcular con la matriz \mathbf{A} y muy costosa numéricamente.

Definition 2.1. Exponencial de una matriz

$$e^{\mathbf{X}} = \mathbf{I} + \sum_{k=1}^{\infty} \frac{\mathbf{X}^k}{k!} \quad (2.3)$$

donde \mathbf{I} es la matriz identidad.

En cambio, lo que se hace en la practica es usar los autovalores y autovectores para efectuar una transformación de coordenadas de las coordenadas de \mathbf{x} a las coordenadas de algún autovector donde es mas fácil escribir la exponencial de una matriz y facilita entender el sistema también.

Un **autovector** $\xi \in \mathbb{C}^n$ cumple con la siguiente igualdad. $\lambda \in \mathbb{C}$ son los **autovalores** del sistema.

$$\mathbf{A}\xi = \lambda\xi$$

Una forma de visualizar esto es que el producto entre la matriz \mathbf{A} y el autovector mantiene la dirección del autovector.

$$\mathbf{T} = [\xi_1, \xi_2, \dots, \xi_n]$$

$$\mathbf{D} = \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_n \end{bmatrix}$$

Es posible diagonalizar el sistema siempre que no se tengan dos autovectores cuasi-paralelos o un sistema degenerado (autovectores generalizados) (entre otros casos).

Esto nos deja escribir la relación

$$\mathbf{A}\mathbf{T} = \mathbf{T}\mathbf{D}$$

$$\mathbf{x} = \mathbf{T}\mathbf{z} \Rightarrow \mathbf{T}^{-1}\mathbf{A}\mathbf{T} = \mathbf{D}$$

$$\mathbf{T}\dot{\mathbf{z}} = \mathbf{A}\mathbf{T}\mathbf{z} \Rightarrow \dot{\mathbf{z}} = \mathbf{T}^{-1}\mathbf{A}\mathbf{T}\mathbf{z}$$

Se obtienen entonces un sistema de ecuaciones desacoplado! El cambio de la variable z_i depende de si misma

$$\dot{\mathbf{z}} = \mathbf{D}\mathbf{z}$$

Se pueden obtener estas matrices en MATLAB en una linea:

`[T, D] = eig(A);`

La solución del sistema va ser simple

$$\mathbf{z}(t) = \begin{bmatrix} e^{\lambda_1 t} & & & 0 \\ & e^{\lambda_2 t} & & \\ & & \ddots & \\ 0 & & & e^{\lambda_n t} \end{bmatrix} \cdot \mathbf{z}(0)$$

Es de interes poder mapear entre los dos espacios. Usando la expresión $\mathbf{A} = \mathbf{T}\mathbf{D}\mathbf{T}^{-1}$ se puede simplificar la exponencial de una matriz empleando conocimientos de algebra lineal

$$e^{\mathbf{A}t} = e^{\mathbf{T}\mathbf{D}\mathbf{T}^{-1}t} = \mathbf{T}e^{\mathbf{D}t}\mathbf{T}^{-1}$$

Cabe destacar que es barato calcular $e^{\mathbf{D}t}$ en términos computacionales.

Reescribimos la solución al sistema recordando $\mathbf{x} = \mathbf{T}\mathbf{z}$

$$\mathbf{x}(t) = \mathbf{T} \underbrace{e^{\mathbf{D}t} \underbrace{\mathbf{T}^{-1}\mathbf{x}(0)}_{\mathbf{z}(0)}}_{\mathbf{z}(t)}$$

La igualdad de arriba se usa para computar la evolución de \mathbf{x} en el tiempo aprovechando la simplicidad del cálculo de $e^{\mathbf{D}t}$.

Que hicimos?

- Descubrimos que si sabemos los autovectores/valores de \mathbf{A} podemos transformar el sistema a un sistema de coordenadas donde es más facil resolver el sistema y estudiar su dinámica

El próximo paso es agregar la matriz de control y el vector de entrada para empezar a controlar el sistema.

Capítulo 3

Estabilidad y autovalores

Para el estudio de estabilidad podemos primero mirar a la igualdad

$$\mathbf{x} = \mathbf{T}e^{\mathbf{D}t}\mathbf{T}^{-1}\mathbf{x}(0)$$

Si uno de los valores diagonal de $e^{\mathbf{D}t}$ se va a infinito entonces la combinación resultante que iguala a \mathbf{x} también se va ir al infinito. Recordemos que $\lambda \in \mathbb{C}$

$$\lambda = a + ib \Rightarrow e^{\pm \lambda t} = e^{at} [\cos(bt) \pm i \sin(bt)]$$

Esto cuenta la siguiente historia

$$\text{si } a > 0 \text{ El sistema aumenta hasta llegar a infinito (**Inestable**)} \quad (3.1)$$

$$\text{si } a < 0 \text{ El sistema converge a cero a tiempo infinito (**Estable**)} \quad (3.2)$$

$$(3.3)$$

Esto significa que tal vez comencemos con un sistema inestable, es decir, que nuestra matriz \mathbf{A} de la siguiente ecuación tiene autovalores con $a > 0$

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$$

Esto se puede remediar agregando el término $\mathbf{B}\mathbf{u}$ de tal forma que lleve los autovalores de la zona inestable ($a > 0$) a la zona estable ($a < 0$).

3.1. Evolución discreta

$$\mathbf{x}_{k+1} = \tilde{\mathbf{A}}\mathbf{x}_k, \quad \mathbf{x}_k = \mathbf{x}(k\Delta t)$$

donde $\tilde{\mathbf{A}} = e^{\mathbf{A}\Delta t}$. Sabiendo el vector de estado inicial podríamos calcular el estado para cualquier otro momento

$$\mathbf{x}_N = \tilde{\mathbf{A}}^N \mathbf{x}_0$$

En coordenadas de autovector, cada vez que multiplicamos la matriz estamos elevando nuestros autovalores de $\tilde{\mathbf{A}}$ a una potencia. Estos pueden agrandarse o achicarse dependiendo de su 'radio'

$$\lambda^N = R^N e^{iN\theta}$$

si el radio R es menor a uno, la magnitud va decaer a medida que pasa el tiempo. Si el radio es mayor que

uno crecerá sin cota superior.

```
[Tt, Dt] = eig(At);
aval = diag(Dt);
inestables = aval(aval>1);
```

3.2. Linealizando un sistema

1. Encontramos los puntos fijos $\bar{\mathbf{x}}$ tal que $f(\bar{\mathbf{x}}) = \mathbf{0}$
2. Linealizamos alrededor de $\bar{\mathbf{x}}$

Para un sistema 2×2 el jacobiano es

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix}$$

$$\dot{\mathbf{x}} = f(\mathbf{x}) = f(\bar{\mathbf{x}}) + \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\bar{\mathbf{x}}} \cdot (\mathbf{x} - \bar{\mathbf{x}}) + \left. \frac{\partial^2 \mathbf{f}}{\partial \mathbf{x}^2} \right|_{\bar{\mathbf{x}}} \cdot (\mathbf{x} - \bar{\mathbf{x}})^2 \dots$$

Como linealizamos alrededor de un entorno reducido, los términos no-lineales van a ser muy pequeños si \mathbf{x} es cercano a $\bar{\mathbf{x}}$. Y dado que es un punto fijo, $f(\bar{\mathbf{x}}) = \mathbf{0}$.

Además, proponemos un cambio de variable $\Delta x_i = x_i - \bar{x}_i$ el cual va representar el distanciamiento de x_i desde el punto donde linealizamos. Nuestro sistema linealizado va quedar así

$$\Delta \dot{\mathbf{x}} = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\bar{\mathbf{x}}} \Delta \mathbf{x} \Rightarrow \boxed{\Delta \dot{\mathbf{x}} = \mathbf{A} \Delta \mathbf{x}}$$

Teorema 3.1 (Hartman–Grobman). Si los autovalores de \mathbf{A} tienen todos parte real entonces se puede describir el sistema como lineal en un vecindario de $\bar{\mathbf{x}}$.

Ejercicio 3.1. Determinar la estabilidad de un péndulo en su posición normal e invertida. Factor de fricción δ .

$$\ddot{\theta} = -\frac{g}{\ell} \sin(\theta) - \delta \dot{\theta}$$

$$\mathbf{x} = \begin{Bmatrix} \theta \\ \dot{\theta} \end{Bmatrix}, \quad \dot{\mathbf{x}} = \begin{bmatrix} x_2 \\ -\frac{g}{\ell} \sin(x_1) - \delta x_2 \end{bmatrix}$$

Nuestro jacobiano es

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{bmatrix} 0 & 1 \\ -\frac{g}{\ell} \cos(x_1) & -\delta \end{bmatrix}$$

Los puntos fijos son

$$\bar{\mathbf{x}} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \pi \\ 0 \end{bmatrix}$$

Matriz del sistema péndulo en posiciones normal d e invertida u :

$$\mathbf{A}_d = \begin{bmatrix} 0 & 1 \\ -\frac{g}{\ell} & -\delta \end{bmatrix}, \quad \mathbf{A}_u = \begin{bmatrix} 0 & 1 \\ \frac{g}{\ell} & -\delta \end{bmatrix}$$

Los autovalores son

$$\lambda_d = \begin{cases} -\frac{\ell \delta + \sqrt{-\ell(4g - \ell \delta^2)}}{2\ell} \\ -\frac{\ell \delta - \sqrt{-\ell(4g - \ell \delta^2)}}{2\ell} \end{cases}, \quad \lambda_u = \begin{cases} -\frac{\ell \delta + \sqrt{\ell(\ell \delta^2 + 4g)}}{2\ell} \\ -\frac{\ell \delta - \sqrt{\ell(\ell \delta^2 + 4g)}}{2\ell} \end{cases}$$

Si se estudia el caso de un péndulo con valores $\frac{g}{\ell} = 1$ y $\delta = 0, 1$

$$\lambda_d = -0,05 \pm 0,9987i, \quad \lambda_u = \begin{cases} -1,0512 \\ 0,9512 \end{cases}$$

Podemos ver que los autovalores del péndulo normal tienen parte real menor a cero. Esto significa que es un sistema **estable**, tiende a cero la solución. En cambio, el péndulo invertido tiene un autovalor mayor que cero, característica de un sistema **inestable**.

Capítulo 4

Controlabilidad

En el capítulo 3 ([Estabilidad y autovalores](#)) se vio que constituye un sistema estable. Para poder controlarlo se deben poder mover los autovalores desde el plano real hacia el complejo (estabilizar el sistema). Esto se hace con control óptimo.

Recordando la ecuación de nuestro sistema (2.1), escribimos nuestro sistema de una forma diferente para poder caracterizar el efecto del input \mathbf{u} sobre la estabilidad (los autovalores):

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} - \mathbf{BK}\mathbf{x} \\ \dot{\mathbf{x}} &= (\mathbf{A} - \mathbf{BK})\mathbf{x}\end{aligned}$$

donde $\mathbf{x} \in \mathbb{R}^n$ y $\mathbf{u} \in \mathbb{R}^p$. El control 'óptimo' para un sistema lineal se logra realimentando $-\mathbf{K}\mathbf{x}$, es decir:

$$\mathbf{u} = -\mathbf{K}\mathbf{x} \quad \text{Control 'óptimo'}$$

Nuestro objetivo ahora es elegir \mathbf{K} para modificar las propiedades de mi sistema, como por ejemplo, la estabilidad. Si nuestro sistema es **controlable** va ser posible hacer estas modificaciones.

Definition 4.1. A grosso modo, mi sistema es controlable si puedo elegir $\mathbf{u} = -\mathbf{K}\mathbf{x}$ y así poner mis autovalores de $\mathbf{A} - \mathbf{BK}$ en cualquier lugar del plano complejo. Si puedo elegir la posición de mis autovalores entonces se puede controlar la evolución del *state-space* eligiendo \mathbf{u} .

Para determinar si un sistema es controlable se construye la matriz de controlabilidad. El sistema es controlable si y solo si se verifica que la cantidad de columnas linealmente independientes sea igual a n .

Definition 4.2. Matriz de controlabilidad

$$\mathbf{Y} = \begin{bmatrix} \mathbf{B} & \mathbf{AB} & \mathbf{A}^2\mathbf{B} & \dots & \mathbf{A}^{n-1}\mathbf{B} \end{bmatrix}$$

En MATLAB se usa la función `ctrb` para obtener la matriz de controlabilidad

CÓDIGO 4.1: Obtención del rango de \mathbf{Y}

```
Y = ctrb(A,B);  
r = rank(Y);
```

4.1. Grados de controlabilidad y gramianes

Mirar el rango de la matriz de controlabilidad nos da un valor binario de la controlabilidad del sistema. Hay estudios más ricos que se pueden hacer para conocer que tan controlable es el sistema.

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}(0) + \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau$$

Definition 4.3. Gramian de controlabilidad

$$\mathbf{W}_t = \int_0^t e^{\mathbf{A}\tau}\mathbf{B}\mathbf{B}^\top e^{\tau\mathbf{A}^\top}d\tau \in \mathbb{R}^{n \times n}$$

Si \mathbf{A} y \mathbf{B} tienen valores reales y positivos entonces \mathbf{W}_t va tener autovalores reales tal que

$$\mathbf{W}_t \xi = \lambda \xi$$

donde los autovectores (ξ de \mathbf{W}_t) correspondientes a los autovalores más grandes serán las 'direcciones' más controlables en el espacio de estados.

Una aproximación del gramian de controlabilidad para sistemas discretos

$$\mathbf{W}_t \approx \mathbf{Y}\mathbf{Y}^\top$$

donde los autovalores de $\mathbf{Y}\mathbf{Y}^\top$ son los valores singulares de \mathbf{Y}

```
[U,SIG, V] = svd(Y, 'econ');
```

donde se listan las columnas de \mathbf{U} como las direcciones más controlables en orden decreciente. La primer columna de \mathbf{U} va ser la dirección más controlable en el espacio de estados.

Es decir, si nuestro vector de input \mathbf{u} tiene norma 1 nuestro sistema \mathbf{x} va poder evolucionar $\xi_1\lambda_1, \xi_2\lambda_2$, etc. Esto tiene fuerte implicaciones en el estudio de estabilidad ya que nos interesa que las direcciones en \mathbf{x} inestables (ξ inestables) sean controlables.

Definition 4.4. Estabilización Un sistema es estabilizable si y solo si todos los autovectores inestables¹ de \mathbf{A} están contenidos en el subespacio controlable.

Definition 4.5. Popov-Belevitch-Hautus test El par \mathbf{A} y \mathbf{B} es controlable si y solo si

$$\text{rango}[(\mathbf{A} - a\mathbf{I}) \quad \mathbf{B}] = n \quad \forall a \in \mathbb{C}$$

Consecuencias del ensayo PBH:

1. $\text{rango}(\mathbf{A} - a\mathbf{I}) = n$ excepto para autovalores $a = \lambda$. Por ende, solo se necesita hacer el test para los autovalores de \mathbf{A}
2. \mathbf{B} necesita tener algun componente en cada en cada dirección de los autovectores de \mathbf{A} .
3. Si \mathbf{B} es una proyección aleatoria, entonces es muy probable que el sistema sea controlable. Esto se debe a que \mathbf{B} solo necesita tener una componente en dirección de cada autovector de \mathbf{A}

Si tengo multiplicidad de autovalores (sistema degenerado) entonces voy a necesitar más de una columna en \mathbf{B} para que se cumpla PBH.

4.2. Posicionamiento de autovalores (polos)

Equivalencias de la definición 4.1

¹También se incluye a veces los autovectores amortiguados en esta definición

1. El sistema es controlable
2. Se pueden elegir posiciones arbitrarias para los autovalores (polos): $\mathbf{u} = -\mathbf{K}\mathbf{x} \Rightarrow \dot{\mathbf{x}} = (\mathbf{A} - \mathbf{BK})\mathbf{x}$
3. Accesibilidad completa en \mathbb{R}^n . Es decir, para todo estado $\xi \in \mathbb{R}^n$ existe un input $\mathbf{u}(t)$ tal que $\mathbf{x}(t) = \xi$.

Para posicionar autovalores en MATLAB se usa el comando `place`

CÓDIGO 4.2: Posicionamiento de autovalores en las posiciones `eigs` deseadas.

```
K = place(A,B,eigs)
```

Capítulo 5

Observabilidad

Ahora hablaremos de estimadores y de la observabilidad del sistema. El algebra para determinar la observabilidad es muy similar al visto en el capítulo de controlabilidad y se comienza a notar una dualidad entre la matriz \mathbf{A} y las matrices \mathbf{B} , \mathbf{C} .

Definition 5.1. Matriz de observabilidad

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix}$$

Definition 5.2. Observabilidad El sistema es observable si el rango (espacio fila) de la matriz \mathcal{O} es igual a n . Verificado en MATLAB:

CÓDIGO 5.1: Como calcular \mathcal{O} y su rango en MATLAB

```
OB = obsv(A,C)
r = rank(OB)
```

Si el sistema es observable entonces se puede estimar todo valor de \mathbf{x} con los valores medidos u observados \mathbf{y} . Más adelante veremos que los filtros Kalman tienen su propio sistema lineal dinámico con autovalores que indican que tan rápido converge el estimador $\hat{\mathbf{x}}$ a \mathbf{x}

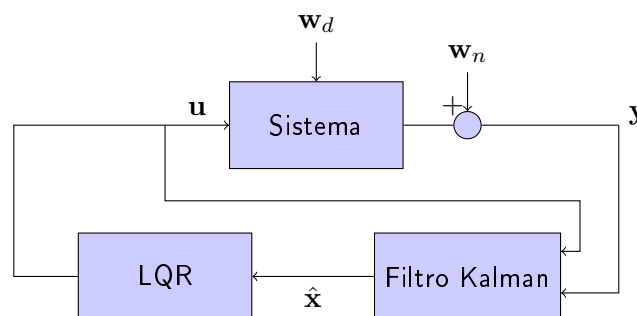


FIGURA 5.1: Sistema de control óptimo con estimador Kalman

Similarmente al estudio de gramianes de controlabilidad, se puede efectuar un *singular value decomposition* para determinar el grado de observabilidad del sistema. En MATLAB:

```
[U, SIG, V] = svd(OB)
```

donde \mathbf{V} va a contener los vectores singulares de la matriz de observabilidad que apuntan en la dirección que se tiene mejor relación señal a ruido, también conocido en ingles como *Signal to Noise ratio* y S/N .

Lo que se hace en la práctica es asegurar que el sistema sea observable verificando el rango de \mathcal{O} , mirar la descomposición singular para determinar direcciones en las cuales se tiene mejor S/N, y construir un filtro de Kalman que estime las variables de estado dado que hay ruido y perturbaciones en el sistema.

Ejercicio 5.1. Estimación de variables 'más' observables Se obtiene el gramian del sistema en MATLAB y se toma el determinante para obtener el volumen del elipsoide de controlabilidad. Cambiando la matriz \mathbf{C} se puede encontrar el esquema de medición más controlable

```
sys = ss(A,B,C,D)
det(gram(ss,'o')) % Cuanto mas grande, mas controlable
```

5.1. Estimación de estado completo

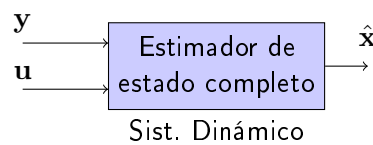


FIGURA 5.2: Estimador de *full-state*.

Un estimador es un sistema dinámico lineal, al igual que los problemas de control que estudiamos. Su dinámica tiene la forma

$$\frac{d}{dt}\hat{\mathbf{x}} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\mathbf{u} + \mathbf{K}_f(\mathbf{y} - \hat{\mathbf{y}}) \quad (5.1)$$

$$\hat{\mathbf{y}} = \mathbf{C}\hat{\mathbf{x}} \quad (5.2)$$

donde la expresión $\mathbf{K}_f(\mathbf{y} - \hat{\mathbf{y}})$ actúa como una corrección al estado actual. Cada medición efectuada \mathbf{y} se compara con la estimada y esto 'corrige' el estado estimado $\hat{\mathbf{x}}$. Si reemplazamos (5.2) en (5.1) se obtiene

$$(\mathbf{A} - \mathbf{K}_f\mathbf{C})\hat{\mathbf{x}} + [\mathbf{B} \quad \mathbf{K}_f] \begin{bmatrix} \mathbf{u} \\ \mathbf{y} \end{bmatrix} \quad (5.3)$$

Definimos un error, el cual vamos a querer reducir para estimar nuestro sistema bien

$$\boldsymbol{\varepsilon} = \mathbf{x} - \hat{\mathbf{x}} \quad (5.4)$$

Recordando la ecuación (2.1) (sin perturbaciones) podemos escribir el cambio del error en el tiempo

$$\begin{aligned} \frac{d}{dt}\boldsymbol{\varepsilon} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{A}\hat{\mathbf{x}} + \mathbf{K}_f\mathbf{C}\hat{\mathbf{x}} - \mathbf{K}_f\mathbf{y} - \mathbf{B}\mathbf{u} \\ &= \mathbf{A}(\mathbf{x} - \hat{\mathbf{x}}) - \mathbf{K}_f\mathbf{C}(\mathbf{x} - \hat{\mathbf{x}}) \\ &= (\mathbf{A} - \mathbf{K}_f\mathbf{C})\boldsymbol{\varepsilon} \end{aligned} \quad (5.5)$$

Lo que me dice esto es que si es observable el sistema entonces puedo elegir un \mathbf{K}_f tal que el sistema sea estable (posicionamiento de autovalores). De esta forma $\boldsymbol{\varepsilon}$ converge a 0. La razón por la que no es deseable tener autovalores negativos (convergencia muy rápida) es por la presencia de perturbaciones en el sistema

y ruido en las mediciones. La decision de la matriz de ganancia \mathbf{K}_f del filtro Kalman se elige en base a la magnitud de estas dos últimas variables.

Capítulo 6

Control lineal cuadrático

6.1. Filtro Kalman

El filtro Kalman es un estimador óptimo que tiene el balance perfecto para rechazar perturbaciones y despreciar ruido. Como mencionamos anteriormente, para diseñar un filtro Kalman se necesita saber que magnitud tienen nuestras perturbaciones \mathbf{w}_d y ruido \mathbf{w}_n . Las suponemos de distribución gaussiana. La razón entre las varianzas \mathbf{v}_d y \mathbf{v}_n nos dirá cual confiamos más de \mathbf{w}_d y \mathbf{w}_n . Si el sistema tiene perturbaciones grandes entonces confiaremos más en las mediciones \mathbf{y} , si se mide mucho más ruido que perturbaciones confiaremos más en el estimador $\hat{\mathbf{x}}$.

Nuestro filtro Kalman ideal minimizará una función de costo \mathcal{J} (estimador lineal cuadrático)

$$\mathcal{J} = E \left((\mathbf{x} - \hat{\mathbf{x}})^T (\mathbf{x} - \hat{\mathbf{x}}) \right) \quad (6.1)$$

donde el valor esperado dependerá de las varianzas \mathbf{v}_d y \mathbf{v}_n . En MATLAB se puede calcular la ganancia de nuestro filtro Kalman \mathbf{K}_f con el comando `lqe`

```
Kf = lqe(A, C, vd, vn)
```

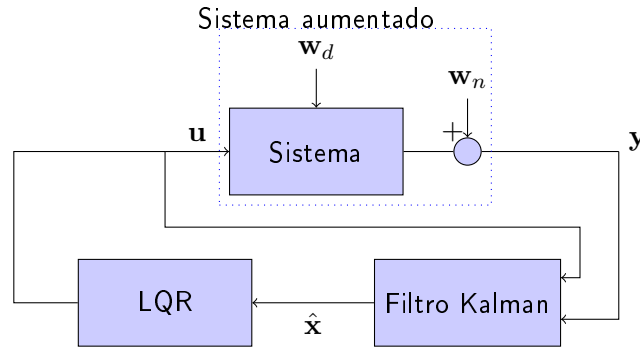


FIGURA 6.1: Sistema de control óptimo con estimador Kalman

Ecuaciones del sistema aumentado

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \underbrace{\mathbf{B}\mathbf{u} + \mathbf{V}_d\mathbf{d}}_{=\mathbf{B} \text{ del sist. aum.}} + \mathbf{0}\mathbf{n} \quad (6.2)$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \underbrace{\mathbf{D}\mathbf{u} + \mathbf{0}\mathbf{d} + \mathbf{V}_n\mathbf{n}}_{=\mathbf{D} \text{ del sist. aum.}} \quad (6.3)$$

donde $\mathbf{V}_d \in \mathbb{R}^{n \times n}$ y $\mathbf{V}_n \in \mathbb{R}^{q \times q}$ son matrices de covarianzas entre perturbaciones de variables de estado y ruido en las mediciones, respectivamente. Si es una matriz diagonal entonces el ruido/perturbación está

desacoplado. El vector de entrada para el sistema aumentado será

$$\begin{bmatrix} \mathbf{u} \\ \mathbf{d} \\ \mathbf{n} \end{bmatrix} \quad (6.4)$$

Mi nuevo vector de entrada tomará los inputs \mathbf{u} , las perturbaciones \mathbf{d} , y el ruido \mathbf{n} . La matriz \mathbf{B} asociada a mi nuevo vector de entrada de mi sistema aumentado puede construirse en MATLAB :

```
BF = [B Vd 0*B] % Input aumentado con perturbaciones y ruido
```

La matriz \mathbf{D} asociada a mi nuevo vector de entrada tendrá la forma

```
DF = [D 0*D Vn] % Input aumentado con perturbaciones y ruido
```

Como el filtro Kalman es su propio sistema dinámico lineal se puede armar su sistema en MATLAB

```
[Kf,P , E] = lqe(A, Vd, C, Vd, Vn)
Kf = (lqr(A', C', Vd, Vn))'
sysKF = ss(A-Kf*C, [B Kf], eye(n), D*[B Kf])
```

donde el vector de entrada al filtro Kalman será

$$\begin{bmatrix} \mathbf{u} \\ \mathbf{y} \end{bmatrix} \quad (6.5)$$

CÓDIGO 6.1: Ejemplo de simulación de sistema $q = 1$ y filtro kalman sin LQR.

```
sysFullOutput = ss(A, BF, eye(n), zeros(n,size(BF,2)))
dt = 0.01;
t = dt:dt:50;
uDIST = randn(4,size(t,2)); % Señal de perturbacion
uNOISE = randn(size(t)); % señal de ruido
u = 0*t; % un solo input
u = (100:120) = 100; % input force right
u(1500:1520) = -100; % input force left

uAUG = [u; Vd*Vd*uDIST; uNOISE];
%%
figure(1)
[y, t] = lsim(sysC, uAUG, t); % lsim simula el sistema con un dado input
plot(t,y);
%%
[xtrue, t] = lsim(sysFullOutput, uAUG, t)
hold on
plot(t, xtrue(:,1), 'r', 'LineWidth',2.0)
%%Plotea variables de estado estimados (punteada) sobre los verdaderos
figure(2)
[xest, t] = lsim(sysKF, [u;y'], t);
plot(t,xtrue,'-',t,xest, 'k--', 'LineWidth', 2.0)
```

6.2. Control Lineal Cuadrático Gaussiano

Si miramos al esquema de la figura 5.1 vemos que se combina un estimador Kalman con un regulador cuadrático lineal para obtener lo que se conoce como un Controlador Lineal Cuadrático Gaussiano (LQG). Lineal porque la dinámica de nuestro sistema es lineal, cuadrático porque la función que quiere minimizar es cuadrática, y Gaussiano porque la distribución del ruido y las perturbaciones se suponen Gaussianas. Se trabaja la ecuación del sistema:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} - \mathbf{B}\mathbf{K}_r\hat{\mathbf{x}} + \mathbf{w}_d \\ &= \mathbf{A}\mathbf{x} - \mathbf{B}\mathbf{K}_r\mathbf{x} + \mathbf{B}\mathbf{K}_r(\mathbf{x} - \hat{\mathbf{x}}) + \mathbf{w}_d\end{aligned}$$

Si a la ecuación (5.5) se le agregan los términos de perturbaciones y ruido se llega a la siguiente expresión

$$\dot{\varepsilon} = (\mathbf{A} - \mathbf{K}_f\mathbf{C})\varepsilon + \mathbf{w} - \mathbf{K}_f\mathbf{w}_n \quad (6.6)$$

luego se puede armar el sistema para el LQG tomando como inputs las perturbaciones y ruidos

$$\frac{d}{dt} \begin{bmatrix} \mathbf{x} \\ \varepsilon \end{bmatrix} = \begin{bmatrix} (\mathbf{A} - \mathbf{B}\mathbf{K}_r) & \mathbf{B}\mathbf{K}_r \\ \mathbf{0} & (\mathbf{A} - \mathbf{K}_f\mathbf{C}) \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \varepsilon \end{bmatrix} + \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{I} & -\mathbf{K}_f \end{bmatrix} \begin{bmatrix} \mathbf{w}_d \\ \mathbf{w}_n \end{bmatrix} \quad (6.7)$$

6.3. Regulador lineal cuadrático

Hasta ahora hablamos de estimación de estados y de la idea de poder controlar un sistema realimentando un input que está en función del estado del sistema. Nos faltaría determinar cual es esta matriz \mathbf{K}_r que crea el input. Si uno conoce los mejores lugares para los autovalores del sistema \mathbf{A} , \mathbf{B} y si el sistema es controlable entonces puede posicionar los autovalores donde desee. Usando MATLAB

```
Kr = place(A,B,eigs);
```

Una pregunta valida llegado a este punto es ¿donde están los mejores autovalores? Dependiendo de donde se posicionen los autovalores va cambiar la respuesta de los actuadores en función de \mathbf{y} y $\hat{\mathbf{x}}$. Aquí es donde aparece uno de los conceptos más poderosos en control, el **regulador lineal cuadrático** (LQR).

La idea detrás del LQR es que se puede poner un costo a cada variable de estado y input. Esta función costo \mathcal{J} aumenta con el tiempo que nuestro sistema no estabiliza en el equilibrio propuesto para $\mathbf{x}(t)$ y con la 'energía' $\mathbf{u}(t)$ utilizada.

$$\mathcal{J} = \int_0^\infty (\mathbf{x}^\top \mathbf{Q} \mathbf{x} + \mathbf{u}^\top \mathbf{R} \mathbf{u}) dt \quad (6.8)$$

La matriz $\mathbf{Q} \in \mathbb{R}^{n \times n}$ asociada con el equilibrio suele ser diagonal. Se puede proponer $\mathbf{Q} = \mathbf{I}$ para empezar a probar el sistema e ir aumentando el elemento diagonal asociado con la variable de estado que más nos interesa que se cumpla. Un valor alto en la diagonal de \mathbf{Q} va implicar un uso alto de energía para cumplir el equilibrio de esa variable de estado asociada. La matriz $\mathbf{R} \in \mathbb{R}^{p \times p}$ asociada a los inputs tiene el mismo formato pero a diferencia de \mathbf{Q} , un valor alto en la diagonal penaliza el uso del actuador asociado a el elemento. Es decir, un valor alto en \mathbf{R} va minimizar el uso de energía.

Resulta que si tenemos \mathbf{Q} y \mathbf{R} hay un protocolo de control $\mathbf{u} = -\mathbf{K}_r \mathbf{x}$ que minimiza \mathcal{J} y se llama el **regulador lineal cuadrático**. En MATLAB se obtiene \mathbf{K}_r

```
Kr = lqr(A,B,Q,R)
```

Si me interesa ver donde están los autovalores del LQR

```
[avec,aval] = eigs(A-B*Kr)
```

6.4. Ejemplo de carro con péndulo

6.4.1. Descripción del sistema

Las ecuaciones de movimiento para un carro con péndulo se trabajan con una fuerza genérica F por simplicidad. g es la gravedad y se plantea en sentido negativo, es decir, g es positiva. $\theta = 0$ corresponde para un péndulo en la posición inferior al carro

$$\begin{cases} (M+m)\ddot{x} &= m\ell\dot{\theta}^2 \sin\theta - m\ell\ddot{\theta} \cos\theta + F \\ m\ddot{x} \cos\theta &= -\frac{b}{\ell}\dot{\theta} - m\ell\ddot{\theta} - mg \sin\theta \end{cases}$$

Para armar el sistema de control o resolverlas numéricamente (como por ejemplo, con los métodos detallados en la sección 7.1 del anexo) se tienen que obtener \ddot{x} y $\ddot{\theta}$ despejados

$$\ddot{x} = \frac{-F + mg \sin\theta \cos\theta + \frac{b}{\ell}\dot{\theta} \cos\theta + m\ell\dot{\theta}^2 \sin\theta}{M + m \sin^2\theta}$$

$$\ddot{\theta} = \frac{F \cos\theta - (m+M)g \sin\theta - (1 + \frac{M}{m})\frac{b}{\ell}\dot{\theta} - m\ell\dot{\theta}^2 \sin\theta \cos\theta}{\ell(M + m \sin^2\theta)}$$

Se muestran al lector las linealización alrededor de los puntos $\theta = 0$ péndulo normal y $\theta = \pi$ para péndulo invertido. Se considera que para ángulos pequeños $\dot{\theta}^2$, $\theta^2 = 0$

$$\ddot{x} = \begin{cases} \frac{F+mg\theta+\frac{b}{\ell}\dot{\theta}}{M} & \text{Linealizando alrededor de } \theta = 0 \\ \frac{F+mg\Delta\theta-\frac{b}{\ell}\dot{\theta}}{M} & \text{Linealizando alrededor de } \theta = \pi \end{cases}$$

$$\ddot{\theta} = \begin{cases} \frac{-F-(m+M)g\theta-(1+\frac{M}{m})\frac{b}{\ell}\dot{\theta}}{\ell M} & \text{Linealizando alrededor de } \theta = 0 \\ \frac{F+(m+M)g\Delta\theta-(1+\frac{M}{m})\frac{b}{\ell}\dot{\theta}}{\ell M} & \text{Linealizando alrededor de } \theta = \pi \end{cases}$$

Se pueden considerar fuerzas de fricción y elásticas sobre el carrito $-\mu\dot{x}$ y $-kx$, respectivamente. Entonces nos queda la matriz del sistema \mathbf{A}

$$\mathbf{A}|_{\theta=0} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{k}{M} & \frac{mg}{M} & -\frac{\mu}{M} & \frac{b}{\ell M} \\ \frac{k}{M\ell} & -\frac{(m+M)g}{M\ell} & \frac{\mu}{M\ell} & -\frac{(1+\frac{M}{m})b}{M\ell^2} \end{bmatrix}$$

$$\mathbf{A}|_{\theta=\pi} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{k}{M} & \frac{mg}{M} & -\frac{\mu}{M} & -\frac{b}{\ell M} \\ -\frac{k}{M\ell} & \frac{(m+M)g}{M\ell} & -\frac{\mu}{M\ell} & -\frac{(1+\frac{M}{m})b}{M\ell^2} \end{bmatrix}$$

$$\mathbf{x}_{\theta=0} = \begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix}$$

$$\mathbf{x}_{\theta=\pi} = \begin{bmatrix} x \\ \Delta\theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix}$$

notar del proceso de linealización que $\Delta\theta = \theta - \bar{\theta} = \theta - \pi$.

7.1. Métodos numéricos para resolución de SEDO

7.1.1. Runge–Kutta orden 4 multivariable

El método de Runge Kutta tiene sus ventajas sobre otros métodos numéricos por ser simple de programar y tener bajo error ($\varepsilon = O(\Delta t^4)$), donde Δt es el paso elegido. El primer paso consiste en describir el sistema de n variables de forma similar a como hacemos con nuestros sistemas lineales,

$$\mathbf{x}' = \mathbf{f}(\mathbf{x}) \quad (7.1)$$

que es lo mismo que escribir

$$\begin{aligned} x'_1 &= f_1(x_1, x_2, \dots, x_n) \\ x'_2 &= f_2(x_1, x_2, \dots, x_n) \\ &\vdots \\ x'_n &= f_n(x_1, x_2, \dots, x_n) \end{aligned}$$

Ahora se definen dos instantes de tiempo $\mathbf{x}|_k$ de valores conocidos y $\mathbf{x}|_{k+1}$ desconocido. Para calcular los valores de $\mathbf{x}|_{k+1}$ el algoritmo Runge–Kutta hace lo siguiente

$$\begin{cases} \mathbf{a}|_k &= \mathbf{f}(\mathbf{x}|_k) \\ \mathbf{b}|_k &= \mathbf{f}(\mathbf{x}|_k + \frac{h}{2} \mathbf{a}|_k) \\ \mathbf{c}|_k &= \mathbf{f}(\mathbf{x}|_k + \frac{h}{2} \mathbf{b}|_k) \\ \mathbf{d}|_k &= \mathbf{f}(\mathbf{x}|_k + h \mathbf{c}|_k) \end{cases} \quad (7.2)$$

$$\mathbf{x}|_{k+1} = \mathbf{x}|_k + \frac{h}{6} (\mathbf{a}|_k + 2 \mathbf{b}|_k + 2 \mathbf{c}|_k + \mathbf{d}|_k) \quad (7.3)$$

El nuevo vector $\mathbf{x}|_{k+1}$ te da el estado del sistema después de un pequeño paso h .