ALGORITMI ITERATIVI

I sistemi dinamici possono anche *descrivere il comportamento degli algoritmi*

- In particolare vediamo adesso gli algoritmi iterativi (es. Metodo di Newton), che possono essere riscritti appunti come sistemi dinamici TD
  - Il *tempo* t del sistema non è più in generale il tempo ma bensì *l'iterazione dell'algoritmo*
  - Lo *stato* $x(t)$ del sistema è rappresentato dalle *variabili in memoria*
  - Gli *ingressi* $u(t)$ sono gli *input dell'algoritmo stesso*
  - L'*uscita* $y(t)$ è l'eventuale *output dell'algoritmo*

ALGORITMO DEL GRADIENTE

Algoritmo per trovare il minimo di una funzione J. Ovvero ad esempio *ottimizzare* i costi. In generale:

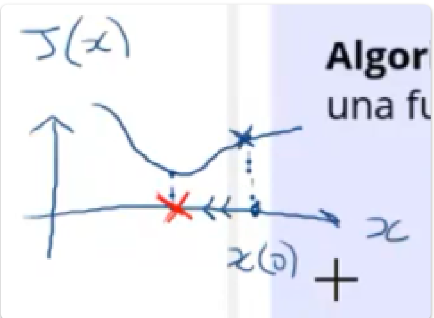
$$\min_x \; J(x)$$

- è una funzione di più variabili

L'approccio del calcolo della derivata diventa poco utile quando le funzioni di più variabili sono complesse o a maggior ragione se non conosco la sua espressione analitica

Si parte perciò a tentativo da un punto $x(0)$ e ci si muove:

- Verso la direzione in cui abbiamo una decrescita maggiore



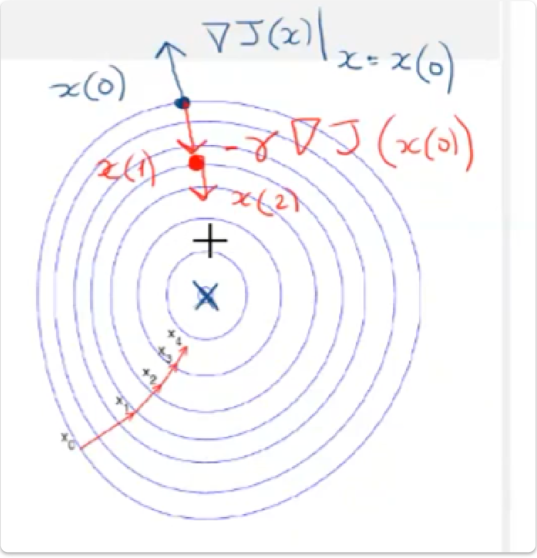
- idea intuitiva: metto una pallina su $x(0)$ e poi per gravità questa si muoverà verso il valore minimo

MODELLO MATEMATICO

- Si calcola il *gradiente*, ovvero il vettore delle derivate parziali

$$\nabla J(x) = \begin{bmatrix} \frac{\partial \, J(x)}{\partial \, x_1} \\ \vdots \\ \frac{\partial \, J(x)}{\partial \, x_n} \end{bmatrix}$$

- il gradiente è perpendicolare (verso l'esterno) alla curva di livello calcolata nel punto di riferimento (in partenza $x(0)$). *Ci si muove pertanto nella direzione dell'antigradiente* (freccia in rosso - direzione di massima discesa)



In generale, l'equazione è la seguente:

$$\boxed{x(t+1) = x(t) + \gamma \nabla J\big(x(t)\big)}$$

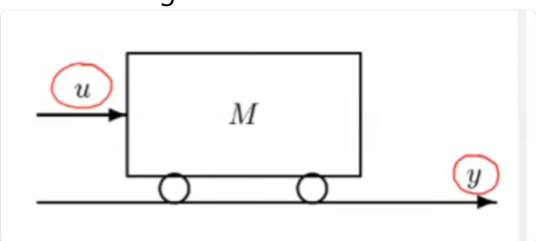
- γ è detto *passo di discesa*
- **È un sistema TD autonomo e tempo-invariante** (perché non dipende dalla iterazione t)
  - Diventa tempo variante se il passo di discesa non è costante (in generale negli algoritmi usati in realtà γ diventa sempre più piccolo con l'aumentare dell iterazioni)
- Sono sistemi dinamici
- Nota: con questo algoritmo si trova un minimo locale (e solo sotto opportune ipotesi)

MODELLI DI SISTEMI FISICI

- Partendo dalle equazioni della fisica si ricavano le informazioni necessarie per creare un sistema dinamico (comodo da gestire/studiare)
  - I sistemi fisici sono dinamici perché *evolvono nel tempo*
- Lo stato nei sistemi fisici è rappresentato dagli *elementi che immagazzinano energia* (ovvero coloro che hanno "memoria")
  - Ad esempio: correnti sugli induttori, tensioni sui condensatori, temperature, posizioni, velocità (cinetica/potenziale)...

ESEMPIO: SISTEMA MECCANICO

- Con un solo grado di libertà "orizzontale"



Trovo le equazioni fisiche descrittive - ovvero il secondo principio di Newton (massa x accelerazione = somma delle forze)

$$M\,\ddot{y}(t) = u(t) - b\,\dot{y}(t)$$

- supponendo come forza d'attrito quello viscoso $-b\,\dot{y}(t)$
- $y(t)$ è la posizione al tempo t

Scegliamo come stato la *posizione e la velocità*

- A partire da queste variabili, si possono scrivere le **equazioni di stato** che cerchiamo:

$$x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} y(t) \\ \dot{y}(t) \end{bmatrix} = \begin{bmatrix} \text{posizione} \\ \text{velocita'} \end{bmatrix}$$

  Formalizzando si arriva a scrivere le seguenti equazioni:
- 1. come varia le posizione nel tempo
- 2. come varia la velocità nel tempo

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = -\frac{b}{M}x_2(t) + \frac{1}{M}u(t) \end{cases}$$

- non autonomo (ingresso $u(t)$) tempo invariante (supponendo che la massa del carrello rimanga la stessa)

Essendo un *sistema lineare*, si può riscrivere tutto in forma matriciale:

$$\begin{aligned} \dot{x}(t) &= \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & -b/M \end{bmatrix}}_{A} x(t) + \underbrace{\begin{bmatrix} 0 \\ 1/M \end{bmatrix}}_{B} u(t) \\ y(t) &= \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{C} x(t) \end{aligned}$$

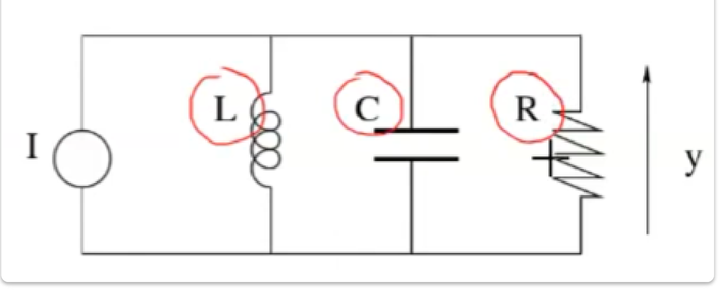
- dove l'uscita $y(t)$ è la posizione, ovvero $y(t) = x_1(t)$

Quindi in generale rientra nel sistema del tipo:

$$\boxed{\begin{cases} \dot{x}(t) = \mathrm{A}\,x(t) + \mathrm{B}\,u(t) \\ y(t) = C\,x(t) \end{cases}}$$

ESEMPIO: CIRCUITO RLC

Sfruttando le leggi della fisica costitutive di resistore, induttore e condensatore e grazie anche a quelle di Kirchoff si giunge a scrivere, per il seguente circuito dotato di generatore di corrente:



- Equazioni costitutive dei tre componenti

$$\begin{aligned} v_R(t) &= R\, I_R(t) \\ v_L(t) &= L\, \dot{I}_L(t) \\ I_C(t) &= C\, \dot{v}_C(t) \end{aligned}$$

- Leggi di Kirchoff

$$\begin{aligned} I(t) &= I_L(t) + I_C(t) + I_R(t) \\ v_R(t) &= v_C(t) = v_L(t) \end{aligned}$$

- Nota: **l'induttore e il condensatore vengono gestite come variabili di stato** perché compare come derivata della corrente, quindi come elemento con memoria. Ovvero:

$$x(t) = \begin{bmatrix} v_c \\ I_L \end{bmatrix} = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \quad , \quad u(t) = I(t)$$

- Combinando equazioni costitutive e leggi di Kirchoff

$$\begin{aligned} I(t) &= I_L(t) + C\dot{v}_C(t) + v_C/R \\ v_c(t) &= L\dot{I}_L(t) \end{aligned}$$

- Riarrangiando i termini

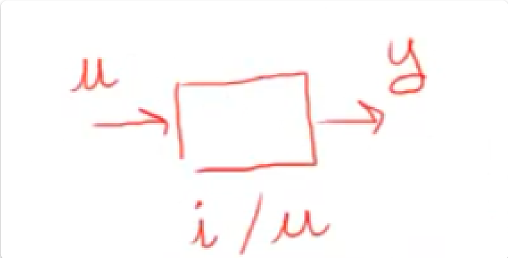
$$\begin{aligned} \dot{x}(t) &= \underbrace{\begin{bmatrix} -1/RC & -1/C \\ 1/L & 0 \end{bmatrix}}_{A} x(t) + \underbrace{\begin{bmatrix} 1/C \\ 0 \end{bmatrix}}_{B} u(t) \\ y(t) &= \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{C} x(t) \end{aligned}$$

RAPPRESENTAZIONE INGRESSO - USCITA

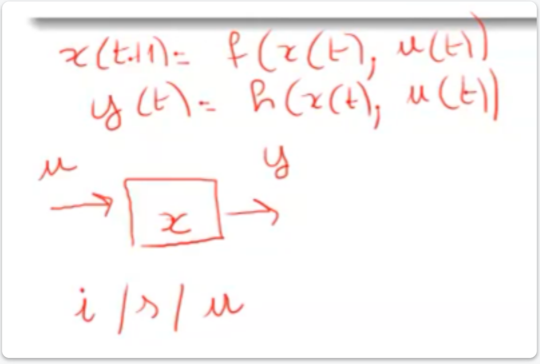
INTRODUZIONE

Un'altra rappresentazione dei sistemi dinamici (meno utilizzata in questo corso rispetto alla rappresentazione con equazione di stato)

- Osservano l'esclusiva configurazione del sistema all'ingresso e all'uscita. Non tengono conto di cosa succede all'interno del sistema stesso. Per questo vengono dette anche *rappresentazioni esterne*



Rappresentazioni interne

> Le rappresentazioni interne invece sono quelle che abbiamo visto: portano a le equazioni di stato
>
> 
>

TEMPO DISCRETO (TD)

Per sistemi tempo discreto, la rappresentazione ingresso uscita è una funzione:

- del tempo (se il sistema è tempo variante)
- autoregressiva delle uscite
- autoregressiva degli ingressi

Autoregressiva: funzione che dipende da sé stessa agli istanti precedenti

In generale (**caso TD**):

$$y(t) = g(t, y(t-1), \ldots, y(t-n), \underbrace{u(t), \ldots, u(t-m)}_{\text{se non autonomo}})$$

- n massimo ritardo con cui compare l'uscita
- m massimo ritardo con cui compare l'ingresso
- *lo stato non compare* (esplicitamente)

PASSAGGIO ALLE EQ. DI STATO: REGRESSORE (CASO TD)

Si può sempre passare da questa rappresentazione a quella equazione di stato

- in linea generale, dovremo reperire le informazioni necessarie per descrivere gli istanti successivi $t+1$

Si tengono in memoria pertanto (cfr. Esempio successione Fibonacci):

- gli ultimi m ingressi
- le ultime n uscite
  In generale considerando il caso non autonomo, si ottiene il seguente **regressore**:

$$x(t) = \begin{bmatrix} y(t-1) \\ \vdots \\ y(t-n) \\ u(t-1) \\ \vdots \\ u(t-m) \end{bmatrix} = \begin{bmatrix} x_1(t) \\ \vdots \\ x_n(t) \\ x_{n+1}(t) \\ \vdots \\ x_{n+m}(t) \end{bmatrix}$$

- **abbiamo** $n + m$ variabili di stato
- come si nota, si tengono in memoria solo le suquende delle uscite e degli eventuali ingressi

FORMULAZIONE GENERALE

Da qui si passa alla formulazione equazione di stato, in questo modo (cfr. Fibonacci per esempio specifico + esercizi)

- Data una rappresentazione ingresso uscita

$$y(t) = g(t, y(t-1), \dots, y(t-n), u(t), \dots, u(t-m))$$

- Scegliamo come stato il regressore

$$\begin{aligned} x(t) &= \begin{bmatrix} y(t-1) & \cdots & y(t-n) & u(t-1) & \cdots & u(t-m) \end{bmatrix}' \\ &= \begin{bmatrix} x_1(t) & \cdots & x_n(t) & x_{n+1}(t) & \cdots & x_{n+m}(t) \end{bmatrix}' \end{aligned}$$

- dinamica dello stato

$$\begin{cases} x_1(t+1) = y(t) = g(t, x_1(t), \dots, x_n(t), u(t), x_{n+1}(t), \dots, x_{n+m}(t)) \\ x_2(t+1) = y(t-1) = x_1(t) \\ \vdots \\ x_n(t+1) = y(t-n+1) = x_{n-1}(t) \\ x_{n+1}(t+1) = u(t) = u(t) \\ x_{n+2}(t+1) = u(t-1) = x_{n+1}(t) \\ \vdots \\ x_{n+m}(t+1) = u(t-m+1) = x_{n+m-1}(t) \end{cases}$$

- equazione di uscita

$$y(t) = g(t, x_1(t), \dots, x_n(t), u(t), x_{n+1}(t), \dots, x_{n+m}(t))$$

Si noti come:

- la prima equazione di stato $x_1(t+1)$ si ricava semplicemente dalla relazione ingresso uscita, infatti è uguale a $y(t)$
- le altre tengono in memoria il necessario, e si ottengono eseguendo uno shift (vale per gli ingressi e per le uscite)
- l'equazione di uscita è anch'essa data dalla semplice $y(t)$, ovvero la rappresentazione ingresso-uscita

ESEMPIO: SUCCESSIONE FIBONACCI

$$y(t) = g(y(t-1) + y(t-2)) \quad , \quad y(0) = 1 \text{ e } y(1) = 1$$

- sistema autonomo (m non presente)
- $n = 2$

Equazioni di stato: individuo ciò che ha memoria (ovvero gli ultimi due valori della successione che servono per calcolare quello nuovo):

$$x(t) = \begin{bmatrix} y(t-1) \\ y(t-2) \end{bmatrix} = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

L'uscita del sistema (immediata) è data da:

$$y(t) = x_1(t) + x_2(t)$$

L'equazioni che legano lo stato invece, per ogni componente di stato sono della forma:

$$x(t+1) = f(x(t))$$

- cioè l'stante successivo dipende dall'istante attuale inserito in una apposita funzione f
  Pertanto:
- per $x_1(t+1)$ prendo $x_1(t) = y(t-1)$ e faccio scorrere di 1 l'indice temporale, quindi: $x_1(t+1) = y(t)$
- faccio lo stesso per $x_2(t+1)$

$$\begin{cases} x_1(t+1) & = & y(t) & = & x_1(t) + x_2(t) \\ x_2(t+1) & = & y(t-1) & = & x_1(t) \end{cases}$$

$$y(t) = x_1(t) + x_2(t)$$

Sono le **equazioni dello stato del sistema**

- da cui come vedremo con l'analisi si può studiare il comportamento