

Ingegneria e Tecnologia dei Sistemi di Controllo

Laurea in Ingegneria Informatica, Sede di Cremona

Alberto Leva

Dipartimento di Elettronica e Informazione, Politecnico di Milano
leva@elet.polimi.it

a.a. 2009-2010

©2006, 2007, 2008, 2009 A. Leva



Lezione 1 (4L)

Introduzione al corso

I problemi fondamentali: controllo di processo e del moto

Principali sensori e attuatori per il controllo di processo e del moto



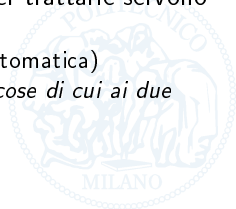
Introduzione al corso

Obiettivi

- Il corso si propone di fornire agli studenti le conoscenze principali su come si costruiscono, si configurano e si gestiscono i sistemi di controllo.
- Il focus è essenzialmente su applicazioni di tipo industriale (ma i concetti sono del tutto generali).
- Dato che il corso è erogato ad allievi Informatici,
 - si dà ampio spazio a come le tecnologie informatiche intervengono nel contesto del controllo
 - ma si parla anche - col grado di dettaglio opportuno - della “fisica” degli impianti e dei relativi organi di controllo, senza la quale il corso sarebbe incompleto.



- Dal nostro punto di vista, quindi, conoscere l'ingegneria e la tecnologia dei sistemi di controllo vuol dire saper
 - fare modelli dinamici degli oggetti da controllare,
 - mettere a punto i relativi regolatori,
 - simulare il sistema di controllo che così si ottiene,
 - implementare i regolatori come software.
- Questo, al livello permesso da 5 CFU, è quel che vi sarà insegnato qui.
- Altre questioni (tipicamente, ma non soltanto) relative agli aspetti hardware dei sistemi di controllo saranno solo menzionate perché per trattarle servono
 - nozioni di Elettronica (che avete),
 - competenza sistemistica (che ottenete dal percorso di Automatica)
 - e molta esperienza, che ci si fa sul campo *se si fanno le cose di cui ai due punti precedenti*.



Introduzione al corso

Collocamento nel percorso didattico (1/2)

- Il corso di Fondamenti di Automatica vi ha insegnato i rudimenti della teoria dei sistemi (LTI) e del controllo (essenzialmente a singolo loop).
- Il tipico problema era: “data la funzione di trasferimento $P(s)$ del processo, trovare la funzione di trasferimento $R(s)$ del regolatore in modo che...”
- Sono quindi aperte alcune domande:
 - chi mi dà $P(s)$?
 - una volta che ho $R(s)$, come lo realizzo?
 - se non devo controllare una sola variabile ma tutto un impianto, come affronto il problema? ...anzi, già che ci siamo: di fronte a un impianto con centinaia di variabili in gioco, chi o cosa sono, cosa significano $P(s)$ e $R(s)$? (capito questo, vi anticipo, capito quasi tutto...)
- In questo corso vedremo (non necessariamente in quest'ordine)
 - i rudimenti su come scrivere *modelli*, ovvero “come passare dalla fisica a $P(s)$ ”;
 - una rassegna degli oggetti fisici che si usano (sensori, attuatori,...);
 - come realizzare i regolatori tramite software (il che, essendo voi Informatici, è il *clou* del corso);
 - le nozioni fondamentali su come costruire una strategia di controllo complessa (molte variabili da controllare, interagenti tra loro).

Introduzione al corso

Collocamento nel percorso didattico (2/2)

- Il corso di Ingegneria e Tecnologia dei Sistemi di Controllo si pone quindi
 - a valle di Fondamenti di Automatica perché ne usa i concetti (assieme a nozioni di base di Fisica, Elettronica e Informatica);
 - in parallelo ad Automazione Industriale (che si occupa del controllo logico mentre qui ci si concentra su quello modulante);
 - a monte di svariati corsi di 2° livello (LS) tra cui, in ordine alfabetico e senza pretese di esaustività,
 - Automazione nei mezzi di trasporto (INF e ATM),
 - Controllo dei robot (INF),
 - Controllo del moto e robotica industriale (ATM),
 - Laboratorio di automazione (per INF o per ATM, sono corsi diversi),
 - Metodi avanzati per il controllo multivariabile (INF),
 - Tecniche di misura e strumentazione avanzate per il controllo (ATM),
 - Tecniche e strumenti di simulazione (INF e ATM)...



- Il corso è organizzato in
 - 28 ore di lezione,
 - 20 ore di esercitazione,
 - 8 ore di laboratorio,per un totale di 5 CFU.
- Le lezioni sono (in parte) guidate con slide,
- le esercitazioni sono svolte con l'ausilio della lavagna e/o del proiettore,
- l'attività di laboratorio è guidata e svolta per gruppi.



Introduzione al corso

Schema sintetico degli argomenti di lezioni ed esercitazioni

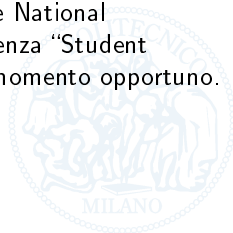
- Introduzione al controllo industriale: controllo di processo e controllo del moto;
- principali sensori e attuatori per il controllo di processo e del moto;
- il controllo di processo: fenomeni principali, modelli di primo principio dei componenti base;
- il controllo del moto: fenomeni principali, modelli di primo principio dei componenti base;
- strutture di controllo;
- schemi tipici per il controllo di processo e del moto;
- architettura hw/sw dei sistemi di controllo e cenni al condizionamento dei segnali;
- regolatori industriali e loro implementazione.



Introduzione al corso

Schema sintetico degli argomenti di laboratorio

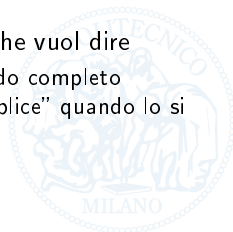
- Realizzazione di un regolatore industriale in ambiente LabVIEW®;
- interfacciamento del regolatore costruito a un apparato fisico per il controllo di temperatura.
- NOTA
LabVIEW è uno strumento professionale per la creazione di sistemi di misura e controllo industriali. In virtù di accordi con il produttore National Instruments, gli studenti del corso potranno avere una licenza “Student Edition” (SE) di LabVIEW: i dettagli saranno spiegati al momento opportuno.



Introduzione al corso

Competenze (rivate dopo aver riassunto il contenuto del corso)

- Ingegneria dei Sistemi di Controllo:
saper scrivere il modello di un sistema, decidere come dev'essere fatto il suo controllo, scrivere anche il modello di tale controllo, metterlo a punto, connetterlo a quello del sistema e simulare il tutto.
- Tecnologia dei Sistemi di Controllo:
conoscere gli elementi fondamentali (sensori, attuatori, regolatori) dei sistemi di controllo e in particolare, dato il *focus* informatico, saper scrivere il software che va nei regolatori.
- Il tutto a livello di un corso di primo livello da 5 CFU, il che vuol dire
 - sapendo risolvere i problemi “abbastanza semplici” in modo completo
 - e sapendo riconoscere un problema “non abbastanza semplice” quando lo si incontra.



Introduzione al corso

Informazioni sul docente e sito del corso

- Alberto Leva
Dipartimento di Elettronica e Informazione
Politecnico di Milano
Tel. 02 2399 3410
E-mail leva@elet.polimi.it
- Ricevimento:
 - a Cremona prima di ogni lezione e in alcune giornate verso la fine del corso;
 - per appuntamento a Cremona o a Milano.
- Pagina web del corso:
<http://home.dei.polimi.it/leva/CorsoITSC/>



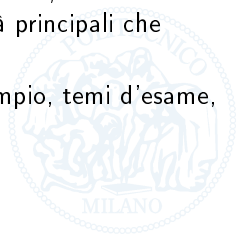
Introduzione al corso

Materiale didattico (1/2)

- Il programma è *interamente* coperto dai testi seguenti:
 - dispense manoscritte del prof. C. Maffezzoni (disponibili in fotocopia presso la Sede);
 - G. Magnani, G. Ferretti, P. Rocco, “Tecnologie dei sistemi di controllo - Seconda edizione”, McGraw-Hill, 2007;
 - slide di guida per le lezioni, disponibili in formato pdf sul sito del corso.

NOTA: le slide *non* sono pensate per sostituire i testi ma come una traccia del corso e/o una guida per prendere meglio appunti a lezione; lo studio sui testi è fondamentale (oltre a essere forse una delle attività principali che faranno di voi dei professionisti).

- Sul sito del corso sono anche disponibili programmi d'esempio, temi d'esame, etc. (ne parleremo a suo tempo).



- Possibili testi di approfondimento e/o consultazione
 - D. Shinskey, "Controlling multivariable processes", Instrument Society of America, 1981;
 - C.L. Albert, D.A. Coggan, "Fundamentals of industrial control", Instrument Society of America, 1992;
 - K.J. Åström, T. Hägglund, "PID controllers: theory, design and tuning, second edition", Instrument Society of America, 1995;
 - B. Wittenmark, K.J. Åström, K.E. Årzén, "Computer control: an overview", IFAC Profesional Brief, 2002, disponibile in formato pdf sul sito <http://www.ifac-control.org>;
 - A. Leva, C. Cox, A. Ruano, "Hands-on PID autotuning: a guide to better utilisation", IFAC Profesional Brief, 2002, disponibile in formato pdf sul sito <http://www.ifac-control.org>;
 - A. O'Dwyer, "Handbook of PI and PID controller tuning rules", World Scientific Press, 2003;
 - A. Leva, "Laboratorio di automatica", Pitagora, 2005.



Introduzione al corso

Software impiegato

- **Scilab** (open source, Scilab license):
software di calcolo che useremo per l'analisi la messa a punto di schemi di controllo basati su sistemi dinamici orientati, ovvero essenzialmente su schemi a blocchi.
- **OpenModelica** (free software, BSD license):
ambiente per la creazione di modelli dinamici acausali, che useremo per simulare alcuni semplici esempi dei modelli di primo principio che impareremo a scrivere.
- **LabVIEW** (software proprietario, possibilità di licenza SE):
ambiente di sviluppo per l'elaborazione dei segnali e il controllo, che useremo per implementare i regolatori in modo “professionalmente realistico” (al livello di questo corso, ovviamente).

Info e download: <http://www.scilab.org/>

Info e download: <http://www.ida.liu.se/~pelab/modelica/OpenModelica.html>

Info: <http://ni.com/labview/>

Introduzione al corso

Modalità d'esame

- Non ci sono prove in itinere.
- L'esame consta di una prova scritta, tipicamente della durata di 2 ore.
- Sono previsti 4 appelli d'esame nei periodi stabiliti dalla Facoltà.



Introduzione al corso

Piano indicativo

Data	Ore	Aula	Argomento
05/10/2009	09.00-13.00 (4L)	B.1.3	Introduzione al corso. I problemi fondamentali: controllo di processo e controllo del moto. Descrizione dei principali sensori e attuatori per il controllo di processo e del moto.
12/10/2009	09.00-13.00 (4L)	B.1.3	Ripasso e complementi sulla sintesi del regolatore in retroazione a tempo continuo. Breve introduzione a OpenModelica.
19/10/2009	09.00-13.00 (4L)	B.1.3	Il controllo di processo: equazioni di conservazione di massa, energia e quantità di moto; modelli dei principali elementi dei processi con fluidi incompressibili (valvole, pompe, condotti, serbatoi a pelo libero).
26/10/2009	09.00-13.00 (4L)	B.1.3	Il controllo del moto: equazioni elettromeccaniche e modello del motore DC a magneti permanenti. Dimensionamento di organi di controllo.
09/11/2009	09.00-13.00 (4E)	B.1.3	Dimensionamento di organi di controllo. Scrittura di modelli per il controllo di processo e del moto.
16/11/2009	09.00-13.00 (4L)	B.1.3	Principali strutture di controllo: feedforward, cascata, disaccoppiamento, predittore di Smith. Breve introduzione a Scilab.
23/11/2009	09.00-13.00 (4E)	B.1.3	Messa a punto di schemi con strutture di controllo.
23/11/2009	09.00-13.00 (4L)	B.1.3	Cenni al condizionamento dei segnali. Architettura hw/sw dei sistemi
30/12/2009	09.00-13.00 (4L)	B.1.3	Regolatori industriali e loro implementazione software.
14/12/2009	09.00-13.00 (4E)	B.1.3	Condizionamento dei segnali di processo e implementazione dei regolatori industriali.
04/01/2010	09.00-13.00 (4E)	B.1.3	Progetto, dimensionamento, simulazione e specifica d'implementazione di un sistema di controllo di temperatura.
11/01/2010	09.00-13.00	Lab 6	Introduzione rapida all'ambiente LabVIEW. Realizzazione guidata di un regolatore PID industriale in LabVIEW.
18/01/2010	09.00-13.00 (4E)	B.1.3	Esercizi di riepilogo.
25/01/2010	09.00-13.00	Lab 6	Realizzazione e prova sperimentale di un sistema di controllo di temperatura.

Introduzione al controllo industriale



Introduzione al controllo industriale

Premessa

- L'argomento "controllo industriale" è vastissimo.
- Per studiare i sistemi di controllo industriali è allora opportuno suddividerli.
- Questo può farsi in vari modi: ad esempio
 - secondo il campo di applicazione
(industria meccanica, generazione elettrica, impianti chimici, zootecnia,...);
 - secondo il tipo di problemi di controllo
(termoregolazione, controllo di pH, controllo d'asse,...);
 - secondo il tipo di funzioni di controllo
(logiche, modulanti, di supervisione,...).
- Naturalmente però
 - in vari campi si trovano più problemi,
 - gli stessi problemi si trovano in più campi,
 - in generale in qualunque campo e/o problema servono più funzioni.
- E' quindi opportuno strutturare la materia in un modo concettualmente diverso. Vediamo come.



Introduzione al controllo industriale

Strutturazione della materia

- Chiediamoci cosa serve per costruire un sistema di controllo in termini di *situazioni tipo*. Scopriamo così alcuni fatti generali.
- Vi sono situazioni in cui bisogna controllare molte variabili, sottoinsiemi anche grandi delle quali sono interagenti, dovendo conseguire obiettivi collocabili in una sorta di piramide gerarchica con molti livelli e pochi enti alla sommità. Questo si riflette in una struttura a molti livelli anche del sistema di controllo, che di solito è realizzato in modo omogeneo e nel quale domina il controllo modulante.

Esempio: in un impianto chimico vi sono centinaia di portate, temperature, pressioni etc. da controllare, molte di esse interagiscono, tuttavia svariate grandezze sono controllate con lo scopo di controllarne altre (ovvero sono loro “asservite”, come la portata di un fluido riscaldante lo è alla temperatura del corpo riscaldato); lo scopo ultimo di tutto è ottenere due o tre prodotti con le caratteristiche volute, per la qual cosa (quasi) ogni singola variabile è rilevante; domina evidentemente il controllo modulante, mentre il controllo logico si occupa di avvio, spegnimento, protezioni e poco altro.

Introduzione al controllo industriale

Strutturazione della materia

- Vi sono invece situazioni in cui bisogna ancora controllare molte variabili, ma con sottoinsiemi interagenti tendenzialmente più piccoli, dovendo conseguire obiettivi ancora collocabili in una piramide gerarchica, ma con meno livelli e più enti paritetici alla sommità. Il sistema di controllo è spesso eterogeneo e vi domina il controllo logico.

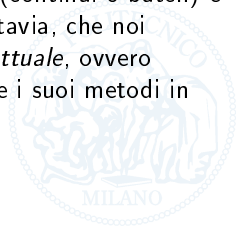
Esempio: in una linea di produzione vi sono molte macchine utensili, ognuna con a bordo un sistema di controllo venduto insieme a essa; l'obiettivo della linea è produrre pezzi, il che richiede di coordinare le macchine, ma al di là di questo non vi sono grosse interazioni a livello di controllo: in altri termini, l'obiettivo generale della linea non è influenzato in dettaglio dai singoli controlli delle macchine, purché queste funzionino correttamente. La parte modulante del sistema sta essenzialmente entro le singole macchine e si occupa del moto dei vari assi, mentre è quella logica che tiene insieme tutta la linea e vi sovrintende.

Introduzione al controllo industriale

Controllo di processo e controllo del moto

- Qualsiasi caso non banale ricade in una di queste due situazioni, o quantomeno contiene “parti” ben individuabili che vi ricadono.
- Quindi, possiamo prendere le due situazioni tipo come i due contesti fondamentali del controllo industriale: nella letteratura tali contesti si chiamano rispettivamente
 - controllo di processo e
 - controllo del moto

perché tendono a presentarsi rispettivamente nei processi (continui o batch) e nelle applicazioni manifatturiere. Non dimentichiamo, tuttavia, che noi adottiamo questa distinzione per la sua importanza *concettuale*, ovvero perché ci permette di presentare (e poi usare) la materia e i suoi metodi in modo organico.



Introduzione al controllo industriale

Controllo di processo e controllo del moto

- I due contesti presentano (con poche eccezioni)
 - differenze negli organi impiegati (sensori, attuatori,...);
 - differenze nel tipo di problemi di controllo (nei processi prevale di solito la necessità di rigettare disturbi, nel controllo del moto quella d'inseguire i set point);
 - forte comunanza di metodi per il progetto dei regolatori (come vedremo, le "strutture" base sono le stesse);
 - differenze architetturali nei sistemi impiegati;
 - comunanza di metodi per l'implementazione, almeno a livello dei "singoli" regolatori.
- Questo spiega come ci muoveremo d'ora in avanti: vedremo
 - prima una rassegna degli organi impegnati,
 - poi come fare i modelli che servono (nei due contesti),
 - poi i metodi (comuni) per strutturare il controllo,
 - poi le architetture (per quel che si riesce in questo corso),
 - e infine come implementare i regolatori.



Principali sensori e attuatori per il controllo di processo e del moto



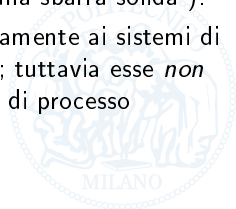
- Vedremo ora una breve descrizione di alcuni
 - organi per governare fluidi¹ (valvole, pompe);
 - sensori di portata (a turbina, a vortici, a strozzamento);
 - sensori di pressione (a deformazione, piezoelettrici);
 - sensori di livello (a galleggiante, a ultrasuoni);
 - sensori di temperatura (termocoppie, termoresistenze);
 - organi per governare il moto (attuatori idraulici, motori elettrici);
 - sensori di posizione analogici (resistivi, induttivi);
 - sensori di posizione digitali (encoder);
 - sensori di velocità (dinamo tachimetrica);
 - sensori di forza (dinamometri, strain gauge);
- Lo scopo di questa panoramica, *non* esaustiva, è
 - vedere in breve “che oggetti si maneggiano”,
 - capire (qualitativamente ma per davvero) il funzionamento di tali oggetti.
- Quando avrete imparato i *principi* per scrivere equazioni, tradurre la comprensione qualitativa in modelli non sarà complicato (e facendo così vi abituerete a lavorare con metodo).

¹In questo corso trattiamo soltanto fluidi incompressibili

In questo corso trattiamo soltanto fluidi incompressibili e, ovunque serva, facciamo l'ipotesi di "moto turbolento completamente sviluppato". Quindi per noi

- massa e volume sono proporzionali tramite la densità, che non dipende dalla pressione (e nemmeno dalla temperatura se non in misura trascurabile);
- in un condotto il profilo di velocità è piatto, ovvero non vi sono scorrimenti di una vena di fluido rispetto a quelle adiacenti, anzi i moti "casuali" delle particelle si elidono globalmente tranne per il moto complessivo del fluido (un modo grossolano ma espressivo di spiegare cosa s'intende è dire che in un condotto a sezione costante il fluido "avanza come fosse una sbarra solida").

Queste ipotesi ci permettono di far vedere quanto serve relativamente ai sistemi di controllo, quindi vanno bene per questo corso (vedasi il titolo); tuttavia esse *non* sono verificate in moltissimi casi d'interesse per la modellistica di processo (quanto manca qui, che è molto, si trova nei corsi appositi).

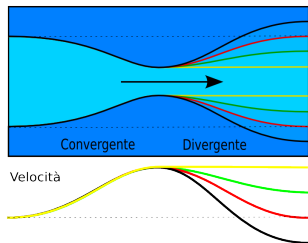


Organi per governare fluidi

Valvole (per fluidi incompressibili)

Principio

Introducendo lungo un condotto un convergente seguito (eventualmente) da un divergente si causa una caduta di pressione e, controllando l'area d'efflusso data dal convergente, si può variare la relazione tra la portata che attraversa la valvola e il salto di pressione che vi cade (in questo senso la valvola equivale a una "resistenza variabile con l'apertura")

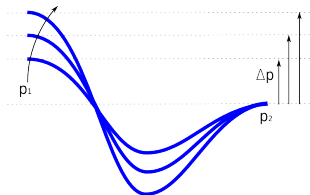


Capiamo la cosa per gradi, ricordando che parliamo di fluidi *incompressibili* in moto *turbolento*.

- Anzitutto, la velocità d'uscita del fluido dipende da quella d'ingresso e dal rapporto tra le sezioni d'uscita e d'ingresso. Si ha quindi, grossolanamente, un andamento della velocità come quello illustrato in figura.

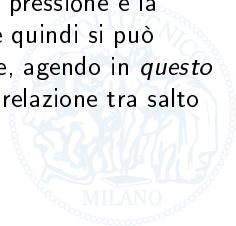
Organi per governare fluidi

Valvole (per fluidi incompressibili)



- Infine, la portata aumenta con il salto di pressione sulla valvola, andando (ovviamente) dal lato a pressione più alta (“monte”) a quello a pressione più bassa (“valle”): una possibile situazione (a pressione di valle impressa) è mostrata in figura.

Concludendo, una valvola istituisce una relazione tra il salto di pressione e la portata, modulabile agendo sull'apertura della valvola stessa, e quindi si può impiegare per “modulare” una portata e/o un salto di pressione, agendo in *questo* senso come “resistenza variabile”. Vedremo in seguito come la relazione tra salto di pressione e portata è fatta.

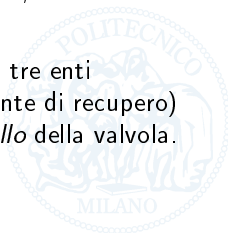


Organi per governare fluidi

Valvole (per fluidi incompressibili)

Le valvole si distinguono principalmente per

- fluidi impiegabili (molto viscosi, corrosivi,...);
- tipo di otturatore (serranda, globo, farfalla,...);
- tipo di attuazione (elettrica, idraulica, pneumatica,...);
- presenza o meno di un sistema di controllo della posizione della valvola (“servoposizionatore”);
- presenza o meno di un dispositivo di “non ritorno” del fluido;
- massimo salto di pressione sopportabile;
- relazione tra portata e salto di pressione, caratterizzata da tre enti (coefficiente d'efflusso, caratteristica d'apertura e coefficiente di recupero) che conosceremo in seguito, quando introdurremo il *modello* della valvola.

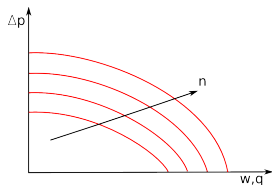


Organi per governare fluidi

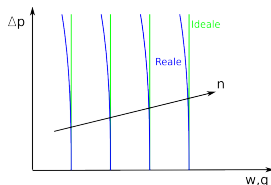
Pompe (per fluidi incompressibili)

Lo scopo di una pompa è innalzare la pressione di un fluido compiendo su di esso del lavoro essenzialmente meccanico (trascuriamo l'aspetto termico perché quantitativamente irrilevante per noi). Tale lavoro proviene da un motore esterno alla pompa vera e propria, anche se spesso compreso nella costruzione del complesso.

Distinguiamo ai nostri fini pompe *centrifughe* e *volumetriche*, ambedue descrivibili con una caratteristica avente sugli assi la portata (w se massica, q se volumetrica, ma per fluidi incompressibili le due sono analoghe) e il salto di pressione Δp .



Pompa centrifuga



Pompa volumetrica

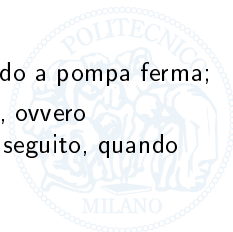
Nelle caratteristiche n indica il "numero di giri", o in senso più lato la velocità, del motore della pompa.

Organi per governare fluidi

Pompe (per fluidi incompressibili)

Le pompe si distinguono principalmente per

- fluidi impiegabili (molto viscosi, corrosivi,...);
- tipo di costruzione e geometria, ovvero
 - centrifughe assiali, radiali,...,
 - volumetriche a pistoncini, a palette deformabili, a ingranaggi, peristaltiche,...;
- tipo di flusso (continuo o pulsante);
- tipo di motore (elettrico, idraulico, pneumatico,...);
- possibilità o meno di variare i giri del motore;
- limiti di funzionamento in termini di pressione e portata;
- presenza o meno di un dispositivo di “non ritorno” del fluido a pompa ferma;
- relazione tra portata e salto di pressione al variare dei giri, ovvero “caratteristica”, le cui possibili espressioni conosceremo in seguito, quando introdurremo i *modelli* dei due tipi di pompa qui trattati.



Sensori tipici del controllo di processo

Portata

- **A turbina**

Il fluido, col suo moto, mette in rotazione una girante, di piccole dimensioni per non perturbare il processo. Dalla velocità di rotazione della girante si risale a quella del fluido e da questa alla portata.

- **A vortici**

Nel condotto è posto un ostacolo di forma tale da generare dei piccoli vortici. Un microfono posto a valle ne sente il passaggio e li conta. Dalla frequenza del passaggio dei vortici si risale alla velocità del fluido e da questa alla portata.

- **A strozzamento**

Una strozzatura nel condotto (orifizio tarato, boccaglio o tubo di Venturi) causa una caduta di pressione, poi recuperata il più possibile per non perturbare il processo. Dalla caduta di pressione si risale alla portata.

Maggiori dettagli e illustrazioni sono sulle dispense e sui testi.



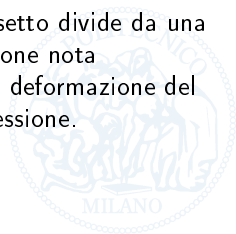
- **A deformazione**

La pressione del fluido causa la deformazione meccanica di un elemento opportunamente costruito (ad esempio sposta un pistoncino contro una molla antagonista, oppure fa diminuire la curvatura di un tubicino ad ansa muovendone in tal modo l'estremità). Dalla deformazione risulta uno spostamento che, misurato (si vedrà poi come), consente di risalire alla pressione.

- **Piezoelettrici**

Il fluido è posto in comunicazione con una cavità che un setto divide da una seconda, posta alla pressione atmosferica o ad altra pressione nota (realizzando così un sensore di pressione differenziale). La deformazione del setto, misurata, conduce a conoscere la (differenza di) pressione.

Maggiori dettagli e illustrazioni sono sulle dispense e sui testi.



Sensori tipici del controllo di processo

Livello

- **A galleggiante**

Il pelo libero del fluido porta con sé un galleggiante, la cui posizione viene misurata e consente quindi di conoscere il livello del fluido stesso entro il serbatoio ove si fa la misura.

- **A ultrasuoni**

Sulla parte superiore del serbatoio è posta una coppia trasmettitore/ricevitore a ultrasuoni, che emette degli impulsi sonori e quindi determina, misurando il tempo intercorrente tra impulso ed eco, la distanza tra sé e il pelo libero del fluido. Da tale distanza si risale al livello.

Maggiori dettagli e illustrazioni sono sulle dispense e sui testi.



Sensori tipici del controllo di processo

Temperatura

● Termocoppie

Le termocoppie sfruttano l'effetto Seebeck, ovvero la differenza di potenziale che si genera alla giunzione tra due conduttori diversi e dipende dalla temperatura della giunzione.

- Una termocoppia è fatta da due giunti con due materiali (ad esempio platino-rodio-platino), detti tipicamente 'giunto freddo' e 'giunto caldo':
 - misurando la tensione ai capi del tutto si risale alla differenza tra le due temperature;
 - mettendo un giunto (tipicamente quello 'freddo') a una temperatura nota (ad esempio perché misurata in altro modo) si ha la temperatura dell'altro.
- Tipicamente i giunti freddi di tutte le termocoppie si pongono in una 'cassetta di compensazione' in modo che basti una sola misura della loro temperatura, e soprattutto che essa sia la stessa per tutte le termocoppie.
- Le termocoppie si distinguono per il 'tipo' (ad esempio quella ferro-costantina è detta 'tipo J'), dal che discende il range di temperature misurabili. Tra i sensori a contatto (escludendo cioè ad esempio le termocamere) le termocoppie raggiungono le temperature più alte.

Maggiori dettagli e illustrazioni sono sulle dispense e sui testi.

- **Termoresistenze**

Vi sono materiali la cui resistività varia con la temperatura, in modo abbastanza lineare entro opportuni range. Se si costruisce un resistore con essi, la relativa resistenza dipenderà quindi dalla temperatura secondo una legge del tipo

$$R = R(T^0) + \alpha(T - T^0).$$

Ponendo un tale resistore in contatto termico con la temperatura da misurare e, dal punto di vista elettrico, in una configurazione a partitore o a ponte di Wheatstone, è semplice generare una tensione linearmente dipendente dalla temperatura da misurare.

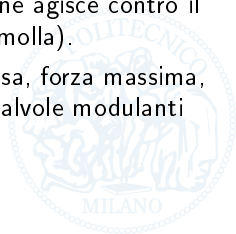
- Occorre tenere bassa la corrente nella termoresistenza per evitare che questa si autoriscaldi per effetto Joule.
- Rispetto alle termocoppie le termoresistenze sono meno precise e raggiungono temperature meno elevate, però sono più semplici da installare e complessivamente costano tendenzialmente meno (ad esempio non occorre nessuna cassetta di compensazione).

Maggiori dettagli e illustrazioni sono sulle dispense e sui testi.

Organi per governare il moto

Attuatori idraulici

- Ve ne sono di diverse geometrie ma il principio si spiega facilmente pensando a quella più comune, ovvero al *cilindro*.
- Un cilindro idraulico è costituito da una camera entro cui scorre un pistone. Facendo entrare o uscire un fluido incompressibile (di solito olio) si provoca il moto del pistone.
- Nei cilindri “a doppio effetto” vi sono due semicamere (una dietro e una davanti al pistone), il che consente sia di estrarre che di ritrarre il pistone medesimo. Nei cilindri “a singolo effetto”, più rari, il pistone agisce contro il solo carico o un dispositivo antagonista (ad esempio una molla).
- I cilindri si distinguono essenzialmente per dimensioni, corsa, forza massima, singolo o doppio effetto, pressione di lavoro, presenza di valvole modulanti oppure on/off per il governo della portata di fluido.



Organi per governare il moto

Motori elettrici

- Ve ne sono di molti tipi: con alimentazione a corrente continua o alternata, sincroni e asincroni, passo-passo, e così via.
- Talmente vasta è la materia che il corso di Azionamenti elettrici ne è occupato per buona parte.
- In questo corso tratteremo più avanti il motore in corrente continua a magneti permanenti: rimandiamo pertanto al prosieguo del corso per dettagli che non ha senso introdurre ora.



Sensori di posizione

Sensori resistivi e induttivi

- **Sensori resistivi**

Facendo in modo che la posizione da misurare corrisponda alla rotazione del cursore di un potenziometro rotativo o alla traslazione di quello di uno lineare si trasduce una posizione in resistenza, in base alla quale è semplice ottenere la misura della posizione stessa.

- **Sensori induttivi**

Il principio è analogo, con la sola differenza che (anche qui con diverse geometrie possibili) si sposta un nucleo ferromagnetico entro un solenoide e si causa quindi una variazione d'induttanza, il che richiede ovviamente per la misura un'alimentazione in corrente alternata.



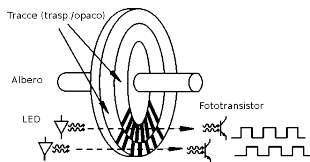
Sensori di posizione

Encoder

- Un encoder è costituito da un disco con parti opache e trasparenti che, alla rotazione del disco medesimo, lasciano o meno passare la luce di alcuni LED, che corrispondenti fototransistor trasformano in un segnale digitale.
- Vi sono due tipi fondamentali di encoder, distinti per l'uso del disco, che si qualificano per caratteristiche meccaniche e “passi per giro”.



Assoluto: n corone circolari successive sono divise ognuna nel doppio dei settori di quella più interna, cosicché il tutto, usando n coppie LED/fototransistor poste radialmente, codifica un numero digitale di n bit che si ripete a ogni giro. I passi per giro sono 2^n .

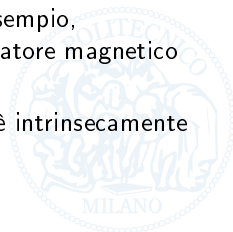


Incrementale: si usano più corone (in generale 4, si veda la dispensa) con n settori opachi e n trasparenti alternati, contando gli impulsi e guardando lo sfasamento tra i segnali prodotti per dedurne il verso di rotazione, in modo da avere la posizione (anche su più giri una volta fissato lo zero del conteggio). I passi per giro sono $2n$.

Sensori di velocità

Dinamo tachimetrica e sensori a conteggio d'impulsi

- Una dinamo tachimetrica è semplicemente un motore a corrente continua “usato a rovescio”.
- Come vedremo più avanti studiando appunto il modello del motore, essa genera una tensione proporzionale alla velocità di rotazione del suo albero.
- Un altro tipo di sensore di velocità è basato sul conteggio (e quindi sul calcolo della frequenza) degli impulsi prodotti ogni volta che un elemento su una parte mobile (ad esempio un foro o un piccolo elemento metallico posti su un disco) passa davanti a uno su una parte fissa (ad esempio, riepistivamente, una coppia LED/fototransistor o un captatore magnetico posti vicino a tale disco).
- Naturalmente un tale sensore, ad differenza della dinamo, è intrinsecamente digitale.



Sensori di forza

Dinamometri e strain gauge

- Un dinamometro trasduce una forza in uno spostamento, tipicamente facendola agire contro un elemento antagonista (ad esempio una molla) di costante elastica nota, e quindi misura tale spostamento.
- Un sensore a strain gauge, invece, sfrutta il fatto che alcuni materiali modificano la loro resistività quando vengono assoggettati a una forza che li deforma. Questo trasduce quindi la forza in una resistenza elettrica, donde è facile ottenere la misura voluta.



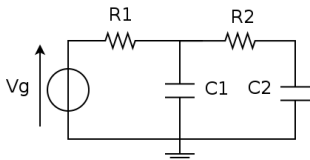
Conclusioni della lezione 1

- Abbiamo
 - introdotto il corso e chiarito obiettivi e collocamento,
 - definito gli argomenti che tratteremo,
 - fatto una breve rassegna dei principali “oggetti fisici” che si usano.
- Non abbiamo trattato i problemi relativi alla caratterizzazione (precisione, ripetibilità,...) dei sensori e nemmeno quelli relativi alla loro *calibrazione* perché, pur importanti, non sono centrali per questo corso. Ovviamente sono trattati nelle sedi opportune (ad esempio a Misure).
- Al momento opportuno tradurremo le nostre nozioni, per ora qualitative, in *modelli dinamici*.



Compito per casa

- Ripassare l'argomento "sintesi del regolatore in retroazione" studiato a Fondamenti di Automatica, che sarà oggetto della prima parte della prossima lezione;
- scaricare da rete e installare OpenModelica, leggere gli esempi introduttivi e provarne qualcuno;
- scrivere e simulare per $t \in [0, 8]$ il modello del circuito mostrato in figura



$$R_1 = 1\text{ k}\Omega, \quad R_2 = 680\Omega, \\ C_1 = 220\mu\text{F}, \quad C_2 = 180\mu\text{F},$$

$$V_g(t) = \begin{cases} 5 \sin(10(t - 1)) & 1 \leq t \leq 5 \\ 0 & \text{altrimenti} \end{cases}$$

partendo da condizioni iniziali nulle.

- La prossima volta vediamo l'esercizio risolto ma non aspettate e provateci da soli. Con le nozioni che avete e una breve ma attenta lettura degli esempi introduttivi siete in grado di farlo (o meglio, con la vostra preparazione ci si aspetta che lo siate: abituatevi presto ad applicare da soli le nozioni apprese che è meglio per voi).
- La prossima volta portare della carta semilogaritmica.

Lezione 2 (4L)

Ripasso e complementi sulla sintesi del regolatore in retroazione a tempo continuo

Breve introduzione a OpenModelica

Questa lezione è svolta usando la lavagna e la carta semilogaritmica (per la prima parte) e in modo prevalentemente interattivo con OpenModelica (per la seconda parte); per la prima parte non ci sono slide guida.



Breve introduzione a OpenModelica



- Modelica (<http://www.modelica.org>) è un “object-oriented modelling language”.
- In Modelica si scrivono *modelli* (*non programmi!*) e qui vedremo quel poco della sintassi del linguaggio che serve per il corso.
- Per simulare un modello Modelica ci vuole un “Modelica translator” che
 - acquisisce il modello Modelica,
 - verifica che il modello sia chiuso correttamente (di questo parleremo),
 - lo manipola simbolicamente se necessario (di questo non parleremo),
 - genera da esso un sorgente in un linguaggio di programmazione (c, c++,...)
che lo simula,
 - compila questo programma e lo linka a quel che serve (solutore numerico,...)
 - e infine lo esegue.
- Vi sono traduttori Modelica commerciali (ad esempio Dymola, <http://www.dynasim.se>) e open source.
- OpenModelica
(<http://www.ida.liu.se/~pelab/modelica/OpenModelica.html>) è un traduttore Modelica open source (licenza BSD).

- I modelli in Modelica sono a-causali, ovvero non orientati, ovvero scritti in modo indipendente da come saranno connessi ad altri modelli.
- Chi approfondirà l'argomento "simulazione", oggi uno dei settori di punta della professione dell'ingegnere, scoprirà che l'approccio a-causale è di enorme potenza; qui forse lo intuirete.
- Al nostro livello, Modelica si regge su 4 concetti base:
 - il concetto di "terminale" o "porta" o "connettore" (in Modelica *connector*),
 - i concetti di variabile "di sforzo" e "di flusso" (in Modelica *effort*, che si omette, e *flow*),
 - il costrutto *equation*, che ammette l'operatore derivata temporale (in Modelica *der*) e quindi le equazioni differenziali ordinarie (ODE),
 - il costrutto *connect*.
- Adesso li spiegheremo e, facendolo, risolveremo l'esercizio dato per casa. Chi l'avesse già fatto è invitato a discutere, ovviamente.



- Un *connector* corrisponde a un terminale “fisico” di un modello.
- Un *connector* contiene delle variabili, di sforzo e/o di flusso.
- Quando il costrutto *connect* connette due terminali si generano due equazioni:
 - una dice che le variabili di sforzo sono uguali,
 - l'altra che la somma di quelle di flusso è zero.
- La cosa si capisce bene definendo il *connector* Pin, che useremo per i terminali elettrici (“i fili”) degli elementi del circuito dell'esercizio:

```
connector Pin
    Real v;          // tensione sul pin
    flow Real i;     // corrente entrante nel pin
end Pin;
```

- Nota: usiamo il tipo Real per semplicità, anche se in Modelica sono definite tutte le unità dimensionali del SI.

- E' ora semplice capire il funzionamento di *connect*. Il testo

```
...  
Pin tizio, caio;  
...  
connect(tizio,caio);  
...
```

genera (notare il consueto significato del punto) le due equazioni

```
tizio.v = caio.v;  
tizio.i+caio.i = 0;
```



- Ora vediamo com'è strutturato un *model* scrivendo e commentando il modello Resistor del resistore: la sintassi dovrebbe quasi essere autoesplicativa.

```
model Resistor
  Pin a,b;
  parameter Real R=1000;
equation
  0 = a.i+b.i;
  0 = a.v-b.v-R*a.i;
end Resistor;
```

- I *parameter* non variano durante una simulazione e si può dar loro un default, che ovviamente può essere “overridden” quando s’istanzia il modello.
- Nella sezione *equation* ci sono *equazioni* (*non assegnamenti!*), che il solutore si occuperà di rendere vere in ogni istante. Chi non ha chiara la differenza tra equazione (che abbiamo scritto come “ $0 = \dots$ ” per meglio evidenziarne la natura) e assegnamento lo dica subito e chiariamo la cosa perché è cruciale.

- Infine vediamo un *model* dinamico, ovvero dove si usa *der* e vi sono variabili di stato. Per farlo scriviamo e commentiamo il modello Capacitor del condensatore

```
model Capacitor
  Pin a,b;
  Real V(start=V_start);
  parameter Real C=1e-6;
  parameter Real V_start=0;
equation
  0 = a.i+b.i;
  0 = a.v-b.v-V;
  0 = C*der(V)-a.i;
end Capacitor;
```

- rendendo *V_start* un *parameter* è possibile assegnare il valore iniziale della variabile di stato *V* per ogni simulazione.



- Già che ci siamo, scriviamo i modelli del generatore di tensione (comandato, e infatti V_g non è un *parameter*) e della massa.

```
model IdealVoltageGenerator
```

```
  Pin a,b;
```

```
  Real  $V_g$ ;
```

```
equation
```

```
   $0 = a.i + b.i$ ;
```

```
   $0 = a.v - b.v - V_g$ ;
```

```
end IdealVoltageGenerator;
```

```
model Ground
```

```
  Pin a;
```

```
equation
```

```
   $0 = a.v$ ;
```

```
end Ground;
```

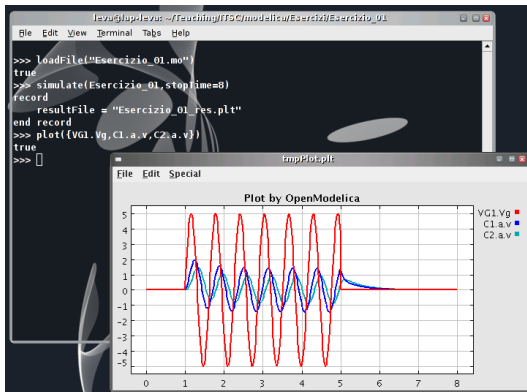


```
model ExerciseIntroModelica01
  Resistor          R1(R=1000);
  Resistor          R2(R=680);
  Capacitor         C1(C=220e-6,V_start=0);
  Capacitor         C2(C=180e-6,V_start=0);
  IdealVoltageGenerator VG1;
  Ground            G;

equation
  connect(VG1.a,R1.a);
  connect(R1.b,C1.a);
  connect(R1.b,R2.a);
  connect(R2.b,C2.a);
  connect(VG1.b,G.a);
  connect(C1.b,G.a);
  connect(C2.b,G.a);
  VG1.Vg = if (time>=1 and time<=5)
    then 5*sin(10*(time-1))
    else 0;
end ExerciseIntroModelica01;
```



- Scritto nel file `ExerciseIntroModelica01.mo` tutto quanto sopra (dalla definizione del pin al modello complessivo), è possibile simulare il modello con OpenModelica e vedere i risultati.



- Ora useremo un po' la shell di OpenModelica e faremo qualche altro piccolo esempio. Mi raccomando di prenderci la mano prima possibile.

- Scrivere in OpenModelica il modello del sistema dinamico con ingresso $u(t)$ e uscita $y(t)$ descritto dalla funzione di trasferimento

$$\frac{Y(s)}{U(s)} = \frac{\mu}{1 + 2\frac{\xi}{\omega_n}s + \frac{s^2}{\omega_n^2}}$$

- Posto $\mu = 1.5$, $\omega_n = 2$, $\xi = 0.4$, $u(t) = \text{sca}(t - 1)$, simulare il sistema per $0 \leq t \leq 10$.
- Vedremo più avanti la soluzione (ExerciseIntroModelica02.mo).



Equazioni di conservazione di massa, energia e quantità di moto

Modelli dei principali elementi dei processi con fluidi incomprimibili



Equazioni di conservazione di massa, energia e quantità di moto



Equazioni di conservazione

Premessa: bilanci statici e dinamici

- **Bilanci statici (a parametri concentrati)**

- Servono a determinare le condizioni di equilibrio di un sistema.
- Sono quindi rappresentati matematicamente da equazioni algebriche.
- Queste equazioni hanno la forma "la sommatoria di qualcosa è uguale a zero".
- Esempi:

- per l'equilibrio meccanico di un corpo rigido vincolato a un punto fisso

$$\begin{cases} \sum(\text{forze}) &= 0 \\ \sum(\text{coppie}) &= 0 \end{cases}$$

- per l'equilibrio termico di un corpo solido

$$\sum(\text{potenze termiche}) = 0$$

- ...



Equazioni di conservazione

Premessa: bilanci statici e dinamici

• Bilanci dinamici (a parametri concentrati)

- Servono a determinare il comportamento dinamico di un sistema (note ovviamente le condizioni iniziali e l'andamento delle variabili d'ingresso o "esogene").
- Sono quindi rappresentati matematicamente da equazioni differenziali.
- Queste equazioni hanno la forma "la sommatoria di qualcosa è uguale alla derivata temporale di qualcos'altro".
- Esempi:
 - per il moto rettilineo di un punto materiale

$$\sum(\text{forze}) = \frac{d}{dt}(\text{quantità di moto})$$

- per l'andamento della temperatura di un corpo solido

$$\sum(\text{potenze termiche}) = \frac{d}{dt}(\text{energia termica})$$

- ...



• Considerazioni generali

- Ogni bilancio statico è la “restrizione a regime” di uno dinamico, ovvero si ottiene da quello ponendo a zero le derivate temporali.
- I bilanci sono quindi “equazioni di conservazione”:
 - nel caso statico lo sono nel senso ovvio che “qualcosa non cambia”, ovvero “si conserva” (avendo derivata temporale nulla),
 - nel caso dinamico lo sono ancora perché dicono che lo stesso qualcosa “si conserva” a patto di mettere nel conto, istante per istante, quanto ne entra e/o ne esce dal sistema per cui il bilancio si scrive, ovvero descrivono l'*accumulo* di quel qualcosa.
- Le equazioni di conservazione si scrivono per un corpo, un insieme di corpi,...
- ...in generale, per un *volume* che ha per *frontiera* una *superficie*. Ad esempio,
 - la derivata temporale della massa contenuta in un volume eguaglia la somma algebrica di tutte le portate massiche che attraversano la superficie frontiera di quel volume,
 - la derivata temporale dell'energia contenuta in un volume eguaglia la somma algebrica di tutte le potenze che attraversano la superficie frontiera di quel volume,
 - ...
 - (ohibò, due equazioni di conservazione che c'interessano le abbiamo già definite).

• Considerazioni generali

- Ci sono poi i sistemi a parametri distribuiti, dove (a differenza del caso a parametri concentrati) le variabili indipendenti, oltre al tempo, sono una o più coordinate spaziali. Ad esempio, la temperatura in un corpo solido esteso è funzione del tempo e del punto, ovvero di una coordinata temporale e tre spaziali.
- Nei modelli a parametri distribuiti compaiono equazioni differenziali alle derivate parziali. Ad esempio, nel corpo *de quo* vi sono flussi termici interni che dipendono dal gradiente della temperatura, ovvero dalle sue tre derivate parziali rispetto alle coordinate spaziali: per capirlo pensate “al finito” che ogni volumetto entro il corpo scambia coi volumetti adiacenti una potenza termica dipendente dal salto di temperatura e quindi immaginate di passare al limite facendo tendere a zero il volume dei volumetti, col che i salti termici (finiti) tra volumetti (finiti) divengono appunto il gradiente della temperatura.
- Naturalmente per risolvere (cioè integrare) tali sistemi occorreranno sia le condizioni iniziali (nel tempo) che quelle al contorno (nello spazio, o più precisamente alla frontiera).

- **Considerazioni generali**

- Fortunatamente, nella modellistica di processo *a fini di controllo* praticamente non servono equazioni alle derivate parziali, se non al più con *una* coordinata spaziale.
- Lo scopo della “nostra” modellistica, infatti, è capire del processo abbastanza per controllarlo, non - ad esempio - scoprire se una zona della parete di una camera di combustione va o meno in crisi termica: per capire cose come questa servono modelli “tridimensionali”, ovvero alle derivate parziali con il tempo e tre coordinate spaziali, ma problemi di questo genere interessano a chi dimensiona e costruisce la camera di combustione, non tanto a chi ne fa il controllo.



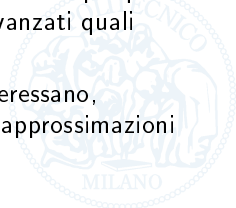
• Considerazioni generali

- Nella modellistica di processo a fini di controllo si distinguono quindi
 - elementi “di accumulo” come i serbatoi, dove è lecito pensare che le caratteristiche del fluido contenuto (ad esempio la temperatura) siano uniformi e dove i moti del fluido medesimo entro l’elemento non interessano perché conta essenzialmente cosa entra e cosa esce: qui bastano pertanto equazioni differenziali ordinarie (con derivate soltanto temporali);
 - elementi “di flusso” e “corti” come le valvole o le pompe, dove i volumi in gioco sono talmente piccoli che non soltanto le distribuzioni spaziali non interessano, ma gli accumuli stessi di massa ed energia sono trascurabili: qui bastano addirittura equazioni algebriche;
 - elementi “di flusso” e “lunghi” (ovvero “con una dimensione prevalente”) come i tubi, dove al più interessa la distribuzione spaziale delle variabili lungo la dimensione prevelente (ossia “lungo il tubo”, si pensi ad esempio ai profili di temperatura in uno scambiatore di calore): qui occorrono equazioni differenziali alle derivate parziali ma bastano due variabili (il tempo e l’ascissa curvilinea del tubo).

Equazioni di conservazione

Premessa

- In questo corso trattiamo soltanto (semplici) modelli, e quindi scriviamo equazioni di bilancio, a parametri concentrati. Vedremo un solo esempio elementare di bilancio dinamico a parametri distribuiti per intuire cosa matematicamente succede.
- Tuttavia è bene che abbiate un'idea del panorama complessivo perché per ogni ingegnere, adottando una terminologia vagamente socratica, è bene non soltanto sapere *bene* quel che si sa, ma anche avere un'idea di “che cosa c'è che non si sa”.
- In concreto, quel che sulla modellistica di processo “non si sa” dopo questo corso, che ha per focus il controllo, si trova in corsi più avanzati quali Controllo dei processi.
- ...e ora passiamo alle equazioni di conservazione che c'interessano, scrivendole nella forma che c'interessa ed evidenziando le approssimazioni introdotte e le relative motivazioni.

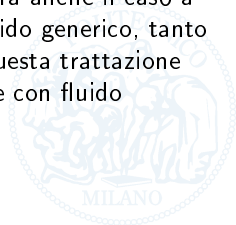


Equazione di conservazione della massa

- Vediamola prima nel caso a parametri concentrati. Detta M la massa contenuta in un volume e dette w_i , $i = 1 \dots N$, le N portate massiche scambiate tra quel volume e l'esterno, assunte con segno positivo se entranti nel volume, l'equazione è

$$\frac{dM(t)}{dt} = \sum_{i=1}^N w_i(t).$$

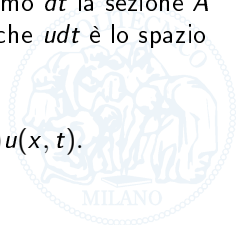
- Soltanto per quest'equazione di conservazione, vediamo ora anche il caso a parametri distribuiti, Proviamo anche a considerare un fluido generico, tanto per vedere che succede perché è molto istruttivo (dopo questa trattazione torneremo a trattare solo il caso a parametri concentrati e con fluido incomprimibile, ovviamente).



Equazione di conservazione della massa

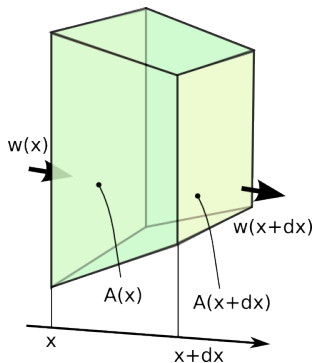
- Consideriamo un condotto. Chiamiamo x l'ascissa curvilinea lungo l'asse del condotto e $A(x)$ la sua sezione, che può quindi non essere uniforme e variare lungo x . Supponiamo però nota a priori la funzione $A(x)$, ossia assumiamo che il condotto sia indeformabile. Chiamiamo poi ρ la densità del fluido (che non assumiamo costante, rilasciando quindi l'ipotesi d'incomprimibilità) e u la sua velocità, che riteniamo uniforme su ogni sezione di condotto ortogonale all'asse (il che equivale a mantenere l'ipotesi di moto turbolento, senza la quale peraltro una sola coordinata spaziale non ci basterebbe).
- Esprimiamo ora la portata massica w che attraversa una sezione di condotto, presa positiva nel verso positivo di x . Nel tempo infinitesimo dt la sezione A è attraversata da un volume di fluido pari ad $Audt$, dato che udt è lo spazio percorso in dt a velocità u . Allora, essendo ρ la densità,

$$w(x, t) = \frac{\rho(x, t)A(x)u(x, t)dt}{dt} = \rho(x, t)A(x)u(x, t).$$



Equazione di conservazione della massa

- Consideriamo ora un tratto di condotto di lunghezza infinitesima dx :



- la massa contenuta, essendo dx infinitesimo, si può calcolare come

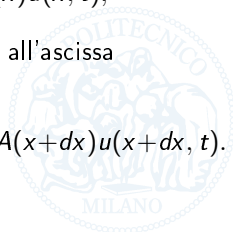
$$M_{dx} = \rho(x, t)A(x)dx,$$

- la portata entrante all'ascissa x è

$$w(x, t) = \rho(x, t)A(x)u(x, t),$$

- e infine la portata uscente all'ascissa $x + dx$ è

$$w(x+dx, t) = \rho(x+dx, t)A(x+dx)u(x+dx, t).$$



Equazione di conservazione della massa

- Quindi l'equazione di conservazione della massa si scrive come

$$\frac{d(\rho(x, t)A(x)dx)}{dt} = \rho(x, t)A(x)u(x, t) - \rho(x + dx, t)A(x + dx)u(x + dx, t)$$

- da cui, separando le variabili,

$$\frac{d(\rho(x, t)A(x))}{dt} = \frac{\rho(x, t)A(x)u(x, t) - \rho(x + dx, t)A(x + dx)u(x + dx, t)}{dx}$$

- e infine, introducendo le necessarie derivate parziali,

$$\frac{\partial(\rho(x, t)A(x))}{\partial t} + \frac{\partial(\rho(x, t)A(x)u(x, t))}{\partial x} = 0$$

o, equivalentemente ma più in sintesi,

$$\frac{\partial \rho A}{\partial t} + \frac{\partial w}{\partial x} = 0.$$



Equazione di conservazione della massa

- Nel caso di fluido incompressibile (ρ costante) l'equazione si riduce ovviamente a $\partial w / \partial x = 0$, ovvero dice che
 - la massa contenuta nel condotto è costante (ovvio, la densità è costante e il volume pure perché il condotto è indeformabile);
 - la portata lungo il condotto è uniforme (ovvio applicando lo stesso ragionamento a qualsiasi tratto di condotto, anche infinitesimo);
 - Au non varia lungo x , quindi la velocità varia in modo inversamente proporzionale alla sezione (conseguenza diretta del fatto che la densità è costante e la portata uniforme).
- Peraltro, capiamo che per un elemento sempre pieno di fluido incompressibile l'equazione di conservazione della massa non dice nulla di utile...



Equazione di conservazione della massa

Considerazioni conclusive dell'*excursus* a parametri distribuiti

- Abbiamo fatto questa breve parentesi
 - per farvi buttare un occhio al di là del contenuto di questo corso,
 - per farvi capire in un caso elementare cosa vuol dire fare un modello a parametri distribuiti,
 - per mostrare come lavora un modellista, ovvero che occorre imparare a capire qual è il problema, quali fenomeni contano, come cambiano le variabili e le equazioni in gioco a seconda delle ipotesi fatte, e così via.
- Come già detto, il “seguito” è in Controllo dei processi e negli altri corsi con significativo contenuto modellistico (che tuttavia si affrontano meglio se i principi sottostanti si capiscono prima possibile).
- D'ora in poi reintroduciamo tutte le ipotesi di questo corso (modelli a parametri concentrati, fluidi incomprimibili e moto turbolento).



Equazione di conservazione dell'energia

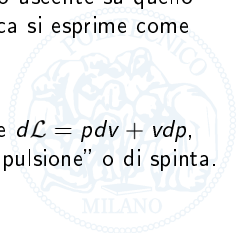
- Consideriamo un volume di fluido che contiene un'energia totale E e dove entrano (o escono se hanno segno negativo) N portate w_i e M potenze termiche Q_j , anch'esse considerate positive se entranti.
- Ovviamente, la derivata temporale di E è la somma delle potenze termiche e dei contributi energetici delle portate, che sono di due tipi coesistenti:
 - apporti (con segno) di energia legati al trasporto di massa, che hanno la forma

$$\text{portata} \times \text{energia specifica del fluido} \quad ([kg/s] \times [J/kg] = [J/s] = [W])$$

- e lavoro (sempre con segno) eseguito dal fluido entrante o uscente su quello contenuto nel volume, che in forma differenziale e specifica si esprime come

$$d\mathcal{L} = d(pv) = d(p/\rho)$$

dove v è il volume specifico e ρ la densità. Nota: è anche $d\mathcal{L} = p dv + v dp$, dove $p dv$ è il "lavoro di compressione" e $v dp$ il "lavoro di pulsione" o di spinta.



Equazione di conservazione dell'energia

- Quindi la grandezza termodinamica che caratterizza l'apporto energetico di una portata di fluido entrante in un volume è l'*entalpia* specifica h del fluido, che si esprime come

$$h = e + \frac{p}{\rho}$$

dove e è l'energia interna specifica.

- Pertanto l'equazione di conservazione dell'energia si scrive

$$\frac{dE(t)}{dt} = \sum_{i=1}^N w_i(t) h_i(t) + \sum_{j=1}^M Q_j(t).$$

- Fortunatamente per fluidi incomprimibili alle pressioni e alle temperature d'interesse per noi l'apporto termico (il termine e) domina enormemente quello di lavoro (il termine p/ρ)². Corrispondentemente è lecito in tali condizioni confondere entalpia ed energia interna specifiche esprimendo ambedue come cT , dove c è il *calore specifico* del fluido (costante nelle nostre ipotesi) e T la temperatura.

²Tuttavia, riuscireste a scrivere l'equazione "esatta", ovvero in entalpia, per un gas ideale: pensateci...

Equazione di conservazione dell'energia

- Quindi per noi è lecito esprimere l'apporto energetico della portata w_i come $w_i c T_i$, dove T_i è la temperatura³ del fluido i -esimo.
- Inoltre in questo corso tratteremo soltanto processi con un solo fluido (niente miscele, cioè) e quindi avremo sempre un solo calore specifico c .
- Tutto ciò premesso, ai nostri fini e con le nostre ipotesi l'equazione di conservazione dell'energia ha la forma

$$c \frac{dT(t)}{dt} = c \sum_{i=1}^N w_i(t) T_i(t) + \sum_{j=1}^M Q_j(t).$$

dove T è la temperatura nel volume considerato, assunta ovviamente uniforme in coerenza con l'approccio a parametri concentrati.

- Anche in questo caso, il “seguito” è in corsi come Controllo dei processi.

³Qualcuno potrebbe chiedersi in che scala. In tal caso cerchi di darsi una risposta e se non la trova pazienti: ripareremo della cosa e la chiariremo.

Equazione di conservazione della quantità di moto

- L'equazione di conservazione della quantità di moto è per noi d'interesse essenzialmente negli elementi “di trasporto” e “lunghi”, ovvero nei condotti.
- Consideriamo quindi un condotto e scriviamo che la derivata temporale della quantità di moto del fluido è la somma delle forze agenti sul fluido medesimo, ovvero
 - forze di pressione agli estremi,
 - gravità
 - e attrito sulla superficie laterale),

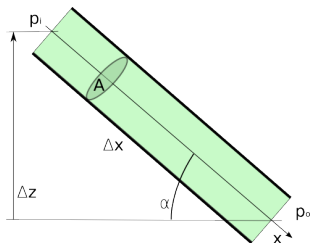
il tutto ai nostri fini proiettato sull'asse x del condotto.



Equazione di conservazione della quantità di moto

- Per mantenere la complessità della trattazione a un livello adatto al corso, consideriamo un condotto a sezione A costante⁴ (e ricordiamo sempre che trattiamo un fluido incompressibile).
- Con queste ipotesi otteniamo

$$M \frac{du(t)}{dt} = Ap_i(t) - Ap_o(t) + Mg \sin(\alpha) - f_a(t).$$



- L'angolo α è tale che $\sin(\alpha) = \Delta z / \Delta x$, Δz essendo la differenza tra le quote iniziale e finale.
- La forza d'attrito f_a , che si oppone sempre al moto, si esprime (detta A_ℓ la superficie laterale) come

$$f_a = K_f A_\ell \rho u |u|$$

⁴Spunto di riflessione: che succede se la sezione non è costante?

Equazione di conservazione della quantità di moto

- La quantità K_f è detta *coefficiente d'attrito*, dipende dalle caratteristiche del contatto fluido/parete ed è valutata sulla base di correlazioni empiriche tratte da esperimenti: si vedano le dispense per ulteriori commenti sulla sua espressione.
- L'equazione contiene ovviamente un termine d'inerzia, che tuttavia nei modelli per il controllo di processo si può in generale trascurare dato che i fenomeni "idraulici" sono enormemente più rapidi di quelli termodinamici, che sono quanto in genere interessa.
- In altri termini, dato che le quantità termiche (temperature, etc.) si propagano essenzialmente con la velocità del fluido mentre quelle idrauliche (pressioni e portate) si propagano con la velocità del suono nel fluido, è lecito pensare che "l'idraulica sia sempre a regime".
- Quindi ai nostri fini la conservazione della quantità di moto si riduce all'equazione algebrica

$$A(p_i(t) - p_o(t)) + Mg \frac{\Delta z}{\Delta x} - K_f A_\ell \rho u(t)|u(t)| = 0.$$

Equazione di conservazione della quantità di moto

- Conclusione: indicando

- con A la sezione (costante, si ricordi) del condotto,
- con L la sua lunghezza
- e con ω il suo perimetro interno,

semplificando un po' la notazione e ricordando che $w = \rho A u$, si ha

$$A(p_i - p_o) + \rho A L g \frac{\Delta z}{L} - K_f \omega L \rho u |u| = 0$$

$$p_i - p_o = K_f \frac{\omega L}{\rho A^3} w |w| - \rho g \Delta z.$$

- Se poi il condotto è installato in modo che w abbia lo stesso segno (con la convenzione di considerarla positiva se va dall'estremo a pressione maggiore a quello a pressione minore) in tutte le condizioni operative d'interesse, si può scrivere

$$p_i - p_o = K_f \frac{\omega L}{\rho A^3} w^2 - \rho g \Delta z.$$

che spesso riassumeremo in

$$p_i - p_o = \frac{K_T}{\rho} w^2 - \rho g \Delta z.$$



Modelli dei principali elementi dei processi con fluidi incomprimibili



Modelli dei principali elementi di processo

Premessa

- Vedremo ora come usare le equazioni di conservazione di

- massa,
- energia
- e quantità di moto,

scritte pensando a elementi di processo con un fluido incompressibile in moto turbolento e adottando un approccio a parametri concentrati, per scrivere i modelli dei principali elementi di tali processi, ovvero

- valvole,
 - pompe,
 - tubi
 - e serbatoi a pelo libero.
- Inizieremo anche a vedere come combinare tali modelli per scrivere il modello completo di un processo (argomento su cui ritorneremo più avanti).

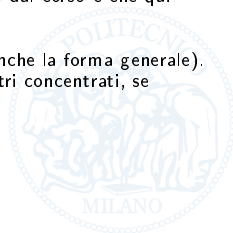


Modelli dei principali elementi di processo

Premessa

Procuriamoci anzitutto un prospetto delle equazioni che ci servono.

Valvola	M	Non si scrive, cioè “si banalizza” (fluido incompressibile). NB questa si banalizza sempre (accumulo comunque trascurabile).
	E	Si banalizza (accumulo e scambi trascurabili). NB anche questa si banalizza sempre (vedi sopra).
	QM	Si usa e conduce alla legge di efflusso che scriveremo. NB nelle valvole intervengono fenomeni che esulano dal corso, ne ripareremo (a livello di cenni) tra poco.
Pompa	M	Come per la valvola (a meno di pompe “grandi” e comprimibilità rilevanti).
	E	Come per la valvola (vedi sopra).
	QM	Si usa e conduce alla caratteristica che scriveremo. NB anche nelle pompe intervengono fenomeni che esulano dal corso e che qui ometteremo del tutto.
Tubo	M	Si banalizza per fluidi incompressibili (qui abbiamo visto anche la forma generale).
	E	Si usa se vi sono scambi rilevanti (noi lo faremo a parametri concentrati, se serve divideremo un tubo in tratti).
	QM	Si usa e conduce alla legge di efflusso che scriveremo.
Serbatoio	M	Si usa (ovvio, l’accumulo è il fatto principale).
	E	Si usa (se interessano i fenomeni termici).
	QM	Si banalizza (i moti interni non sono per noi rilevanti).



Modelli dei principali elementi di processo

Premessa

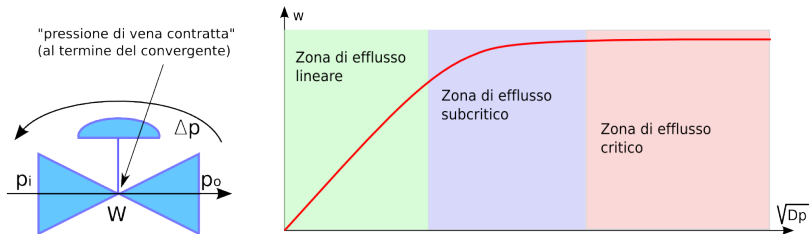
- Ora procederemo a scrivere e commentare i modelli.

- **Nota importante**

D'ora in avanti manterremo sempre le ipotesi di fluido incomprimibile in moto turbolento e modelli scrivibili a parametri concentrati, ma per brevità non le ripeteremo più: ricordatevi però che le assumeremo sempre vere e quindi i nostri modelli avranno validità limitata al contesto, pur abbastanza ampio, di questo corso. Il seguito ormai sapete bene dov'è.



Modello della valvola



- Sperimentalmente, ad apertura fissata, si rileva quanto sopra.
 - Nella zona lineare, la sola che c'interessa (e dove peraltro una valvola di solito va fatta lavorare), la valvola si comporta come una resistenza variabile con l'apertura (la curva "si alza" aprendo e "si abbassa" chiudendo). Si noti che la legge è quadratica, coerentemente con l'equazione di conservazione della quantità di moto per un condotto: qui la valvola si comporta proprio come un condotto a sezione variabile.
 - Nelle altre zone intervengono fenomeni che si spiegano soltanto con un'analisi della trasformazione termodinamica del fluido che esula del tutto da questo corso: noi ci accontentiamo di sapere che l'esperienza mostra che le cose vanno come illustrato.

Modello della valvola

- Detta allora $x \in [0, 1]$ la “posizione” della valvola, il modello è per noi

$$w = C_{v_{max}} \Phi(x) \sqrt{\Delta p}$$

dove

- $C_{v_{max}}$ è il *coefficiente d'efflusso* della valvola, corrispondente alla massima apertura, ovvero alla massima area del convergente;
- $\Phi(x)$, $\Phi(0) = 0$, $\Phi(1) = 1$, è la *caratteristica intrinseca* della valvola, ovvero una funzione che dice come l'apertura della valvola varia, andando da zero al massimo, quando la posizione (che si chiama così perché di fatto è proprio la posizione dello stelo o del volantino) varia tra l'estremo “tutto chiuso” (0) e quello “tutto aperto” (1).
- Un altro parametro caratteristico, detta p_{vc} la pressione di vena contratta, è il *coefficiente di recupero*

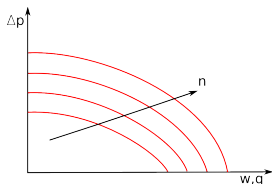
$$F = \frac{p_i - p_{vc}}{p_i - p_o}.$$

- Nota: con un fluido incompressibile il modello è del tutto analogo se si usa la portata volumetrica q invece di quella massica w (cambiano ovviamente valore e dimensioni di $C_{v_{max}}$).

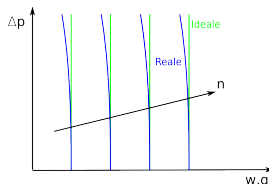
Modello della valvola

- Per *dimensionare* una valvola, argomento su cui torneremo e faremo esempi, occorre allora
 - scegliere Cv_{max} in modo che, in tutte le condizioni di lavoro previste, la valvola “ce la faccia” a far passare la portata richiesta;
 - dato per scontato che il ruolo della valvola sia quello di modulare una certa portata \bar{w} (di solito, ma non sempre, quella che attraversa lei stessa), scegliere la caratteristica intrinseca $\Phi(x)$ in modo che la *caratteristica installata* $\bar{w}(x)$ sia (il più possibile) lineare;
 - scegliere F in modo che, in tutte le condizioni di lavoro previste, p_{vc} non divenga troppo bassa (se questo avvenisse il fluido alla fine del convergente potrebbe uscire dalle condizioni di incomprimibilità o addirittura vaporizzare).
- Inoltre è bene far lavorare sempre le valvole tra il 25% e il 75% di apertura circa, perché quando sono quasi del tutto aperte o chiuse le imperfezioni nella loro costruzione cominciano a contare e l'azione di modulazione della portata si fa imprecisa. Se serve si possono montare in serie e/o parallelo due o più valvole di taglia diversa: ad esempio per modulare una portata fino a zero tipicamente si mette una valvola grande in parallelo a una piccola e, vicino a portata zero, si chiude del tutto la grande e si lavora con la piccola.

Modello della pompa



Pompa centrifuga

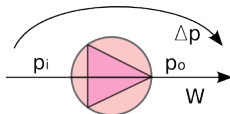


Pompa volumetrica

- Sperimentalmente si rileva quanto sopra:

- una pompa centrifuga si comporta come un generatore di (aumento di) pressione ideale in serie a una resistenza (quadratica, al solito, quindi proveniente dagli attriti che compaiono nella conservazione della quantità di moto);
- una pompa volumetrica (si ricordi sempre che con gli incompressibili $w = \rho q$) si comporta come un generatore di portata ideale o quasi, nel senso che a salti di pressione molto alti una piccola resistenza si avverte per l'effetto congiunto della comprimibilità del fluido e della deformabilità della pompa (noi comunque non consideriamo tali effetti).

Modello della pompa



- Quindi per una pompa centrifuga (fissato il fluido) il modello è

$$\Delta p = H_0(n) - H_1(n)w^2$$

dove n è il numero di giri e, assegnato un *numero di giri nominale* n_0 cui corrispondono $H_0 = \overline{H}_0$ e $H_1 = \overline{H}_1$, si ha tipicamente

$$H_0(n) = \overline{H}_0 \frac{n}{n_0}, \quad H_1(n) = \overline{H}_1 \frac{n}{n_0}.$$

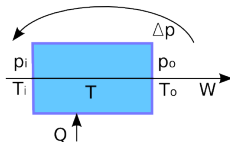
- Per una pompa volumetrica invece si ha semplicemente

$$w = Kn$$

dove K è un parametro caratteristico della pompa (tralasciamo qui la differenza tra i casi a flusso continuo e pulsante)



Modello del tubo



- Per valvole e pompe i fenomeni termici sono spesso (e ai nostri fini sempre) trascurabili.
- Nel caso dei tubi, almeno a volte, non è così (pensiamo per esempio agli scambiatori di calore).
- Dovremo allora tener conto sia dell'“idraulica” che della “termica”, ovvero scrivere sia la conservazione della massa (equazione algebrica dato che per noi l'idraulica è “infinitamente rapida”) che quella dell'energia (e questa sarà invece un'equazione differenziale, ordinaria perché operiamo a parametri concentrati).
- Inoltre, valvole e pompe sono costruite e installate per funzionare con la portata in un solo verso e quindi a noi non interessa scrivere modelli che tengano conto dell'“inversione di portata”. Per questo abbiamo tranquillamente scritto w^2 mentre, per tener conto del fatto che l'attrito si oppone sempre al moto, avremmo dovuto scrivere per l'esattezza $\bar{w}|w|$.

Modello del tubo

con “inciso” sull’inversione di portata

- Richiamiamo l’equazione di conservazione della quantità di moto per un tubo a sezione costante (ci limitiamo a questo caso, come già detto):

$$p_i - p_o = K_f \frac{\omega L}{A} \rho u |u| - \rho g \Delta z.$$

- Ricordando che $w = \rho A u$, quindi,

$$\Delta p = K_f \frac{\omega L}{A} \rho \frac{w |w|}{\rho^2 A^2} - \rho g \Delta z,$$

$$\Delta p = K_f \frac{\omega L w |w|}{\rho A^3} - \rho g \Delta z.$$

- Infine, inglobando in un solo coefficiente K_T tutte le caratteristiche del tubo ma lasciando per chiarezza espressa la dipendenza da ρ , si ha

$$\Delta p = \frac{K_T}{\rho} w |w| - \rho g \Delta z$$

che fornisce la “parte idraulica” del modello.



Modello del tubo

con “inciso” sull’inversione di portata

- Passiamo ora alla “parte termica”. Introducendo la nota semplificazione $h = cT$, se $w > 0$ risulta subito

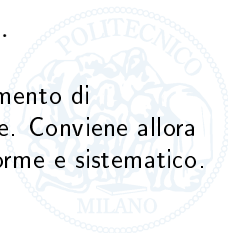
$$\rho V c \frac{dT}{dt} = cw(T_i - T_o) + Q, \quad T_o = T,$$

dove V è il volume (ovviamente costante) del tubo.

- Se invece $w < 0$ sarà

$$\rho V c \frac{dT}{dt} = cw(T_i - T_o) + Q, \quad T_i = T.$$

- Decisamente un modello scritto così non è chiaro e al momento di connetterlo con altri è molto facile che si generi confusione. Conviene allora adottare (e non soltanto per i tubi) un approccio più uniforme e sistematico.



Modello del tubo

con “inciso” sull’inversione di portata

- Allora d’ora in poi *tutti* gli elementi (non solo i tubi anche se cominciamo da loro)
 - saranno considerati dei volumi di controllo,
 - avranno portate di scambio con altri volumi e il verso di tali portate sarà positivo se esse sono entranti nel volume,
 - avranno potenze termiche di scambio con altri volumi e il verso di tali potenze sarà positivo se esse sono entranti nel volume,
- Applichiamo ciò al tubo:

$$w_i + w_o = 0$$

$$p_i - p_o = \frac{K_T}{\rho} w_i |w_i| - \rho g \Delta z$$

$$\rho V c \frac{dT}{dt} = c w_i T_i + c w_o T_o + Q$$

if $w_i > 0$ then

$$T_o - T = 0$$

else

$$T_i - T = 0$$



Modello del tubo

con “inciso” sull’inversione di portata

- Questo modello contiene delle *equazioni condizionali* (e per inciso non tratta il caso in cui w resti nulla a lungo perché ai nostri fini non interessa)⁵.
- Ricordiamo che nel modello ci sono **equazioni**, non **assegnamenti**! Chi non ha ancora compreso la differenza colmi subito questa lacuna perché altrimenti non capirà nulla né di modellistica e nemmeno di software per la medesima, anche se sa tutti i linguaggi di programmazione e di specifica del mondo.
- Proprio per evidenziare questo fatto ed eliminare potenziali equivoci le equazioni entro il costrutto *if* sono scritte facendole terminare con “=0”: il costrutto *if* nei modelli con equazioni condizionali NON significa “se questo è vero si fa questo calcolo” ma “se questo è vero deve valere quest’equazione”.
- Per inciso, ci occorreranno ambienti (software) di simulazione capaci di trattare le equazioni condizionali. Ne vedremo brevemente uno (OpenModelica, che avrete a disposizione) ma non potremo qui approfondire l’argomento. Chi è interessato a questo importante mondo è invitato, alla LS, a seguire il corso di Tecniche e strumenti di simulazione.

⁵Spunto di riflessione: perché non lo tratta? Cosa manca? Come si potrebbe fare per ovviare a tale mancanza?

Modello del serbatoio a pelo libero

con “inciso” sull’inversione di portata

- Consideriamo un serbatoio a pelo libero (per semplicità cilindrico), di area di base A . Dobbiamo scrivere, stanti le premesse, le equazioni di conservazione di massa ed energia.
- Esprimiamo prima la massa M e l’energia E contenute e poi le loro derivate. Indichiamo con l il livello; il significato degli altri simboli usati è il solito. Facendo l’ipotesi che il serbatoio non tracimi⁶ otteniamo

$$M = \rho A l$$

$$E = M c T$$

$$\frac{dM}{dt} = \sum_{i=1}^N w_i$$

$$\frac{dE}{dt} = \sum_{i=1}^N w_i c T_i + \sum_{j=1}^M Q_j$$

$$\text{for } i = 1 : N$$

$$\text{if } w_i < 0 \text{ then } T_i - T = 0$$

- Ovviamente il significato di *for* $i=1:N$ qui è “si scrivono N equazioni”...

⁶Spunto di riflessione: e se no?



Modelli con le semplificazioni che adotteremo nel corso

- Il nostro scopo è capire come si scrive un modello e come lo si controlla.
- Non dobbiamo necessariamente rappresentare tutti i fenomeni (per quello ci sono i corsi di modellistica).
- Allora teniamo soltanto i fenomeni che ci consentono di scrivere modelli abbastanza semplici da poterli tradurre in Modelica e simulare senza problemi.
- A tal fine scriveremo (e simuleremo) soltanto modelli di componenti di processo rappresentanti fenomeni *idraulici*⁷.
- Già che ci siamo, i modelli dei componenti li scriviamo direttamente in Modelica.
- Assumiamo la convenzione che le portate siano positive quando entrano nell'elemento; come vedremo questo ci aiuterà molto nello scrivere modelli di processo come “aggregati” dei modelli dei componenti.

⁷Chi è curioso si chiedi il perché e provi a rispondere, eventualmente ne discutiamo.

Modelli con le semplificazioni che adotteremo

Connettori

Anzitutto, definiamo i tre *connector* che ci servono.

Useremo il connector *Flange*, per le flange degli elementi dove transita massa (e anche energia, ma abbiamo detto che rappresentiamo soltanto l'idraulica, quindi c'importano solo portate e pressioni).

```
connector Flange
    Real p      "pressure [Pa]";
    flow Real w "mass flowrate [kg/s]";
end Flange;
```

Useremo i connector *SignalIn* e *SignalOut* per le connessioni (orientate!) relative a comandi e misure da e verso il sistema di controllo; questo ci permetterà di realizzare le relative connessioni con *connect* che funziona tra connector, non tra variabili.

```
connector SignalIn      | connector SignalOut
    input Real signal;   |      output Real signal;
end SignalIn;            | end SignalOut;
```

Modelli con le semplificazioni che adotteremo

Valvola (con caratteristica lineare)

```
model ValveWithLinearChar
  Flange    inlet;
  Flange    outlet;
  SignalIn  cmd;
  Real      x "valve opening command [# , 0-1]";
  parameter Real Cvmax = 0.01 "maximum Cv (x=1) [(kg/s)/sqrt(Pa)]";
equation
  x = min(1,max(cmd.signal,0));
  0 = inlet.w+outlet.w;
  0 = inlet.w-Cvmax*x*sqrtReg(inlet.p-outlet.p);
end ValveWithLinearChar;
```

NOTA: la funzione sqrtReg è un'approssimante di $\sqrt{|x|}/\text{sign}(x)$ definita $\forall x$ e con derivata continua e finita, il che evita problemi numerici in simulazione (nei corsi di metodi numerici s'impara questo e altro); $\text{sqrtReg}(x)$ è definita come

$$\frac{x}{\sqrt{\sqrt{x^2 + \delta^2}}}$$

dove δ è, in sintesi, la distanza da $x = 0$ per cui $\text{sqrtReg}(x)$ praticamente si confonde con $\sqrt{|x|}/\text{sign}(x)$.

Modelli con le semplificazioni che adotteremo

Pompa centrifuga

```
model CentrifugalPump
  Flange inlet;
  Flange outlet;
  SignalIn cmd;
  Real n "pump command [# , 0-1], 1 meaning rpm=nominal rpm";
  parameter Real H0bar = 1e5 "zero-flow head [Pa]";
  parameter Real H1bar = 50 "head reduction coeff. [Pa/(kg/s)^2]";
equation
  n = min(1,max(cmd.signal,0));
  0 = inlet.w+outlet.w;
  0 = outlet.p-inlet.p-(H0bar*n-H1bar*n*inlet.w^2);
end CentrifugalPump;
```



Modelli con le semplificazioni che adotteremo

Pompa volumetrica

```
model VolumetricPump
  Flange inlet;
  Flange outlet;
  SignalIn cmd;
  Real n "pump command [#, 0-1], 1 meaning rpm=nominal rpm";
  parameter Real n0 = 1000 "nominal rpm [#]";
  parameter Real w0 = 1 "mass flowrate at nominal rpm [kg/s]";
equation
  n = min(1,max(cmd.signal,0));
  0 = inlet.w+outlet.w;
  inlet.w = w0*n;
end VolumetricPump;
```



Modelli con le semplificazioni che adotteremo

Tubo

```
model Duct
  Flange inlet;
  Flange outlet;
  parameter Real KT = 0.001 "friction coeff. [Pa/(kg/s)^2]";
  parameter Real V = 1 "duct volume [m^3]";
  parameter Real Dz = 0 "in-out vertical dist. [m]";
  parameter Real ro = 1000 "fluid density [kg/m^3]";
equation
  0 = inlet.w+outlet.w;
  0 = inlet.p-outlet.p-(KT*inlet.w^2-9.81*ro*Dz);
end Duct;
```



Modelli con le semplificazioni che adotteremo

Serbatoio con due flange (sul fondo)

```
model Tank2flanges
  Flange flange1;
  Flange flange2;
  Real l(start=l_start);
  Real p;
  Real M;
  parameter Real A      = 1 "tank base area [m^2]";
  parameter Real ro     = 1000 "fluid density [kg/m^3]";
  parameter Real l_start = 0 "initial fluid level [m]";
equation
  M = ro*A*l;
  der(M) = if M>=0 then flange1.w+flange2.w else 0;
  p = ro*9.81*l;
  p = flange1.p;
  p = flange2.p;
end Tank2flanges;
```



Modelli con le semplificazioni che adotteremo

...e adesso proviamo a usare quel che abbiamo imparato per scrivere un modello di processo semplice ma non banalissimo: vedrete che con l'approccio sistematico che ci siamo dati è più facile di quel che sembra.



ExampleHydro01

Il processo da simulare

Una pompa volumetrica P a giri variabili preleva acqua a pressione atmosferica e la manda a un serbatoio a pelo libero Tank1. Da esso due valvole V12 e V13 portano a due altri serbatoi a pelo libero Tank2 e Tank3. Da questi due altre valvole (V2o e V3o, rispettivamente) scaricano in atmosfera. Gli ingressi del sistema sono i comandi a pompa e valvole.

Disegnate qui sotto uno schema del processo, così vi allenate a capire una descrizione a parole (tra 1 min lo facciamo insieme).



ExampleHydro01

Codice Modelica (1/2)

Scriviamo ora il modello in Modelica, assumendo dati quelli dei componenti. Definiamo inoltre i valori dei parametri fisici e gli andamenti degli ingressi. Lasciamo al default lo stato iniziale (ossia poniamo i livelli iniziali a zero).

```
model ExampleHydro01
  FixedP atm(p=0);
  VolumetricPump Pump(w0=0.1);
  Tank3flanges Tank1(A=0.01);
  Tank2flanges Tank2(A=0.01);
  Tank2flanges Tank3(A=0.01);
  ValveWithLinearChar V12(Cvmax=2e-3);
  ValveWithLinearChar V13(Cvmax=2e-3);
  ValveWithLinearChar V2o(Cvmax=2e-3);
  ValveWithLinearChar V3o(Cvmax=2e-3);
```



ExampleHydro01

Codice Modelica (2/2)

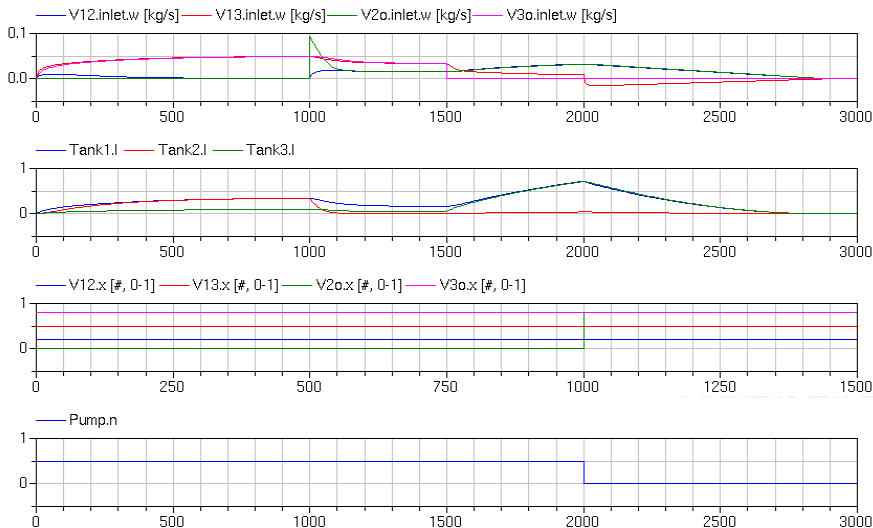
equation

```
connect(atm.flange,Pump.inlet);
connect(Pump.outlet,Tank1.flange1);
connect(Tank1.flange2,V12.inlet);
connect(Tank1.flange3,V13.inlet);
connect(V12.outlet,Tank2.flange1);
connect(V13.outlet,Tank3.flange1);
connect(Tank2.flange2,V2o.inlet);
connect(V2o.outlet,atm.flange);
connect(Tank3.flange2,V3o.inlet);
connect(V3o.outlet,atm.flange);
Pump.cmd.signal = if time<2000 then 0.5 else 0;
V12.cmd.signal  = 0.2;
V13.cmd.signal  = 0.5;
V2o.cmd.signal  = if time<1000 then 0 else 0.8;
V3o.cmd.signal  = if time<1500 then 0.8 else 0;
end ExampleHydro01;
```



ExampleHydro01

Risultato della simulazione (per 3000 s)



- In appendice alla lezione si riporta, perché vi diate un'occhiata, il codice “flat” del modello `ExampleHydro01`, ovvero quello generato dal traduttore (Dymola oppure OpenModelica) dopo aver risolto tutte le `connect`.
- Se leggendolo lo capite avete capito un punto chiave di come funziona l'astrazione modellistica dei linguaggi di modellazione (non di programmazione!) Object-oriented come Modelica.
- Sempre se avete capito, non dovrebbe spaventarvi l'idea di scrivere “a mano” un modello anche complesso, purché sappiate suddividerlo in componenti separati da connessioni fisiche, scrivere i modelli dei primi e astrarre come terminali (`connector`) le seconde. Il modello sarà semplicemente fatto dalle equazioni costitutive dei componenti e da quelle di legame.
- Chiaramente facendo così si ottengono molte equazioni di cui una gran parte è banale (sono essenzialmente quelle di legame che vengono dalle `connect`). Per fortuna il lavoro di manipolazione e semplificazione è fatto dal traduttore.
- La ricerca su modellistica e simulazione Object-oriented è un campo stimolante per informatici “duri” ed è di grande interesse per l'industria. Chi non è spaventato dall'idea di lavorarci, a suo tempo lo segnali.

Appendice

Codice Modelica flat di ExampleHydro01 (1/7)

```
model ExampleHydro01
parameter Real source.p = 0 "flange fixed pressure [Pa]";
parameter Real sink.p = 0 "flange fixed pressure [Pa]";
parameter Real Pump.n0 = 1000 "nominal rpm [#]";
parameter Real Pump.w0 = 0.1 "mass flowrate at nominal rpm, i.e., n=1 [kg/s]";
parameter Real Tank1.A = 0.01 "tank base area [m^2]";
parameter Real Tank1.ro = 1000 "fluid density [kg/m^3]";
parameter Real Tank1.l_start = 0 "initial fluid level [m]";
parameter Real Tank2.A = 0.01 "tank base area [m^2]";
parameter Real Tank2.ro = 1000 "fluid density [kg/m^3]";
parameter Real Tank2.l_start = 0 "initial fluid level [m]";
parameter Real Tank3.A = 0.01 "tank base area [m^2]";
parameter Real Tank3.ro = 1000 "fluid density [kg/m^3]";
parameter Real Tank3.l_start = 0 "initial fluid level [m]";
parameter Real V12.Cvmax = 0.002 "maximum Cv (x=1) [(kg/s)/sqrt(Pa)]";
parameter Real V13.Cvmax = 0.002 "maximum Cv (x=1) [(kg/s)/sqrt(Pa)]";
parameter Real V2o.Cvmax = 0.002 "maximum Cv (x=1) [(kg/s)/sqrt(Pa)]";
parameter Real V3o.Cvmax = 0.002 "maximum Cv (x=1) [(kg/s)/sqrt(Pa)]";
Real source.flange.p "pressure at flange [Pa]";
Real source.flange.w "mass flowrate entering flange [kg/s]";
Real sink.flange.p "pressure at flange [Pa]";
Real sink.flange.w "mass flowrate entering flange [kg/s]";
```



Appendice

Codice Modelica flat di ExampleHydro01 (2/7)

```
Real Pump.inlet.p "pressure at flange [Pa]";
Real Pump.inlet.w "mass flowrate entering flange [kg/s]";
Real Pump.outlet.p "pressure at flange [Pa]";
Real Pump.outlet.w "mass flowrate entering flange [kg/s]";
Real Pump.cmd.signal;
Real Pump.n "pump command [#, 0-1], 1 meaning rpm=nominal rpm";
Real Tank1.flange1.p "pressure at flange [Pa]";
Real Tank1.flange1.w "mass flowrate entering flange [kg/s]";
Real Tank1.flange2.p "pressure at flange [Pa]";
Real Tank1.flange2.w "mass flowrate entering flange [kg/s]";
Real Tank1.flange3.p "pressure at flange [Pa]";
Real Tank1.flange3.w "mass flowrate entering flange [kg/s]";
Real Tank1.l(start = Tank1.l_start);
Real Tank1.p;
Real Tank1.M;
Real Tank2.flange1.p "pressure at flange [Pa]";
Real Tank2.flange1.w "mass flowrate entering flange [kg/s]";
Real Tank2.flange2.p "pressure at flange [Pa]";
Real Tank2.flange2.w "mass flowrate entering flange [kg/s]";
Real Tank2.l(start = Tank2.l_start);
Real Tank2.p;
Real Tank2.M;
```



Appendice

Codice Modelica flat di ExampleHydro01 (3/7)

```
Real Tank3.flange1.p "pressure at flange [Pa]";
Real Tank3.flange1.w "mass flowrate entering flange [kg/s]";
Real Tank3.flange2.p "pressure at flange [Pa]";
Real Tank3.flange2.w "mass flowrate entering flange [kg/s]";
Real Tank3.l(start = Tank3.l_start);
Real Tank3.p;
Real Tank3.M;
Real V12.inlet.p "pressure at flange [Pa]";
Real V12.inlet.w "mass flowrate entering flange [kg/s]";
Real V12.outlet.p "pressure at flange [Pa]";
Real V12.outlet.w "mass flowrate entering flange [kg/s]";
Real V12.cmd.signal;
Real V12.x "valve opening command [#, 0-1]";
Real V13.inlet.p "pressure at flange [Pa]";
Real V13.inlet.w "mass flowrate entering flange [kg/s]";
Real V13.outlet.p "pressure at flange [Pa]";
Real V13.outlet.w "mass flowrate entering flange [kg/s]";
Real V13.cmd.signal;
Real V13.x "valve opening command [#, 0-1]";
Real V2o.inlet.p "pressure at flange [Pa]";
Real V2o.inlet.w "mass flowrate entering flange [kg/s]";
Real V2o.outlet.p "pressure at flange [Pa]";
```



Appendice

Codice Modelica flat di ExampleHydro01 (4/7)

```
Real V2o.outlet.w "mass flowrate entering flange [kg/s]";
Real V2o.cmd.signal;
Real V2o.x "valve opening command [#, 0-1]";
Real V3o.inlet.p "pressure at flange [Pa]";
Real V3o.inlet.w "mass flowrate entering flange [kg/s]";
Real V3o.outlet.p "pressure at flange [Pa]";
Real V3o.outlet.w "mass flowrate entering flange [kg/s]";
Real V3o.cmd.signal;
Real V3o.x "valve opening command [#, 0-1]";

function sqrtReg
  input Real x;
  input Real delta := 0.01 "Range of significant deviation from sqrt(x)";
  output Real y;
algorithm
  y := x/sqrt(sqrt(x*x+delta*delta));
end sqrtReg;
```



Appendice

Codice Modelica flat di ExampleHydro01 (5/7)

```
equation
source.flange.p = source.p;
sink.flange.p = sink.p;
Pump.n = min(1, max(Pump.cmd.signal, 0));
0 = Pump.inlet.w+Pump.outlet.w;
Pump.inlet.w = Pump.w0*Pump.n;
Tank1.M = Tank1.ro*Tank1.A*Tank1.l;
der(Tank1.M) = (if Tank1.M >= 0 then Tank1.flange1.w+Tank1.flange2.w+Tank1.flange3.w else 0);
Tank1.p = 9.81*Tank1.ro*Tank1.l;
Tank1.p = Tank1.flange1.p;
Tank1.p = Tank1.flange2.p;
Tank1.p = Tank1.flange3.p;
Tank2.M = Tank2.ro*Tank2.A*Tank2.l;
der(Tank2.M) = (if Tank2.M >= 0 then Tank2.flange1.w+Tank2.flange2.w else 0);
Tank2.p = 9.81*Tank2.ro*Tank2.l;
Tank2.p = Tank2.flange1.p;
Tank2.p = Tank2.flange2.p;
Tank3.M = Tank3.ro*Tank3.A*Tank3.l;
der(Tank3.M) = (if Tank3.M >= 0 then Tank3.flange1.w+Tank3.flange2.w else 0);
Tank3.p = 9.81*Tank3.ro*Tank3.l;
Tank3.p = Tank3.flange1.p;
Tank3.p = Tank3.flange2.p;
V12.x = min(1, max(V12.cmd.signal, 0));
```



Appendice

Codice Modelica flat di ExampleHydro01 (6/7)

```
0 = V12.inlet.w+V12.outlet.w;  
0 = V12.inlet.w-V12.Cvmax*V12.x*sqrtReg(V12.inlet.p-V12.outlet.p,0.01);  
V13.x = min(1, max(V13.cmd.signal, 0));  
0 = V13.inlet.w+V13.outlet.w;  
0 = V13.inlet.w-V13.Cvmax*V13.x*sqrtReg(V13.inlet.p-V13.outlet.p,0.01);  
V2o.x = min(1, max(V2o.cmd.signal, 0));  
0 = V2o.inlet.w+V2o.outlet.w;  
0 = V2o.inlet.w-V2o.Cvmax*V2o.x*sqrtReg(V2o.inlet.p-V2o.outlet.p,0.01);  
V3o.x = min(1, max(V3o.cmd.signal, 0));  
0 = V3o.inlet.w+V3o.outlet.w;  
0 = V3o.inlet.w-V3o.Cvmax*V3o.x*sqrtReg(V3o.inlet.p-V3o.outlet.p,0.01);  
Pump.cmd.signal = (if time < 2000 then 0.5 else 0);  
V12.cmd.signal = 0.2;  
V13.cmd.signal = 0.5;  
V2o.cmd.signal = (if time < 1000 then 0 else 0.8);  
V3o.cmd.signal = (if time < 1500 then 0.8 else 0);  
source.flange.p = Pump.inlet.p;  
Pump.inlet.w+source.flange.w = 0;  
Tank1.flange1.p = Pump.outlet.p;  
Pump.outlet.w+Tank1.flange1.w = 0;  
V12.inlet.p = Tank1.flange2.p;  
Tank1.flange2.w+V12.inlet.w = 0;  
V13.inlet.p = Tank1.flange3.p;
```



Appendice

Codice Modelica flat di ExampleHydro01 (7/7)

```
Tank1.flange3.w+V13.inlet.w = 0;  
V12.outlet.p = Tank2.flange1.p;  
Tank2.flange1.w+V12.outlet.w = 0;  
V2o.inlet.p = Tank2.flange2.p;  
Tank2.flange2.w+V2o.inlet.w = 0;  
V13.outlet.p = Tank3.flange1.p;  
Tank3.flange1.w+V13.outlet.w = 0;  
V3o.inlet.p = Tank3.flange2.p;  
Tank3.flange2.w+V3o.inlet.w = 0;  
V3o.outlet.p = V2o.outlet.p;  
sink.flange.p = V2o.outlet.p;  
V2o.outlet.w+V3o.outlet.w+sink.flange.w = 0;  
end ExampleHydro01;
```

DAE having 57 scalar unknowns and 57 scalar equations.

Per i curiosi che sono giunti fin qui: notato com'è definita in Modelica una function? Di fatto è possibile mischiare modellazione a-causale (con equazioni e connettori non orientati) e algoritmica (con programmi contenenti assegnamenti), il che sarà estremamente utile quando descriveremo i regolatori, sia come sistemi dinamici che - appunto - come algoritmi. Ne parleremo a lezione, comunque.

Lezione 4 (4L)

Il controllo del moto: equazioni elettromeccaniche e modello del motore DC a magneti permanenti

Dimensionamento di organi di controllo



Il controllo del moto



Il controllo del moto

Premesse

- Il concetto di 'controllo del moto' è stato già introdotto.
- Lo scopo di questa parte del corso è capire come lo si affronta a livello di scrittura dei modelli.
- La sintesi dei regolatori verrà qui solo accennata e sarà ripresa dopo aver visto le strutture di controllo (che evidentemente servono); in tutto il corso, per coerenza col livello, si limiterà la complessità della trattazione mantenendo il focus sul controllo di un singolo asse.
- Per affrontare problemi più complessi servono nozioni geometriche, cinematiche (e anche dinamiche) che si trovano nei corsi specializzati, come ad esempio quelli di robotica.



Il controllo del moto

Struttura della trattazione

- Anzitutto considereremo un tipo di motore elettrico, ovvero quello in corrente continua (DC) a magneti permanenti: questo ci permetterà d'introdurre il concetto di fenomeni (quindi equazioni, quindi modelli) elettromeccanici, primo esempio (per noi) di modelli e quindi simulazione 'multidomain', dove coesistono equazioni provenienti da domini fisici diversi e, ai fini della descrizione degli oggetti d'interesse, inscindibili.
- Quindi vedremo altri (pochi e semplici) modelli di oggetti meccanici, mantenendo un livello di approfondimento sui fenomeni descritti adatto al corso.
- Infine scriveremo qualche semplice modello di catena (elettro)meccanica: il problema del controllo del moto in simili catene (essenzialmente, controllo di posizione e velocità sarà trattato in seguito).



Equazioni elettromeccaniche e modello del motore DC a magneti permanenti



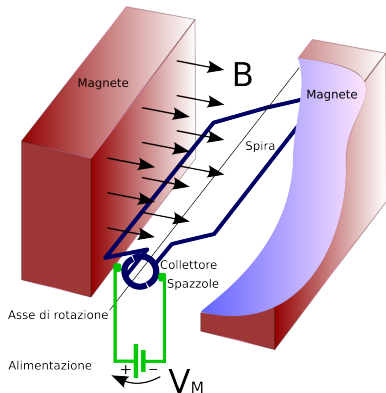
Equazioni elettromeccaniche

Descrizione dei fenomeni che avvengono nel motore DCMP

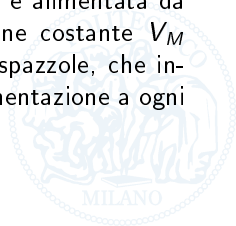
- Per descrivere gli elementi di nostro interesse (visto il motore DC a magneti permanenti - 'DCMP' d'ora in poi per brevità - e alcuni altri elementi sapremo praticamente tutto quel che ci serve) dobbiamo rappresentare, al livello di questo corso, fenomeni
 - elettrici di tipo resistivo e induttivo (negli avvolgimenti),
 - elettromagnetici (dovuti al moto del rotore entro il campo magnetico statorico),
 - meccanici (inerzie, elasticità e attriti).
- Vedremo anche come tali fenomeni si combinano in modo tale da rendere piuttosto innaturale la descrizione dei sistemi elettromeccanici con modelli "block-oriented", ovvero fatti da sistemi *orientati* (à la Simulink/Scicos);
- con l'approccio a-causale (à la Modelica) che abbiamo adottato, invece, le cose sono decisamente più semplici e soprattutto comprensibili.
- Procediamo.

Il motore DCMP

Schema di principio



Per spiegare il funzionamento del motore DCMP partiamo dal comportamento di una spira conduttrice (caratterizzata quindi da una resistenza R_M e un'induttanza L_M) rettangolare vincolata a ruotare su un asse, immersa in un campo magnetico costante B (prodotto da una coppia di magneti permanenti con i poli opposti affacciati) e alimentata da un generatore di tensione costante V_M tramite un collettore a spazzole, che inverte la polarità dell'alimentazione a ogni mezzo giro della spira.



Il motore DCMP

Principio di funzionamento

- Una carica elettrica q in moto con velocità \vec{u} entro un campo magnetico \vec{B} è soggetta alla forza di Lorentz $\vec{F} = q\vec{u} \wedge \vec{B}$.
- Consideriamo ora un tratto infinitesimo di conduttore, di area A e lungo dx , x essendo l'ascissa (eventualmente curvilinea) del conduttore stesso. Detta ρ_q la densità di carica [C/m^3] e i la corrente, risulta immediatamente che tra corrente i , velocità u delle cariche e carica dq contenuta nell'elemento infinitesimo di conduttore valgono le relazioni

$$i = \rho_q Au, \quad dq = \rho_q A dx$$

- Detto ora $\vec{\tau}$ il versore tangente all'ascissa curvilinea x , si ha che sull'elemento infinitesimo di conduttore agisce la forza infinitesima

$$d\vec{F} = \rho_q A dx \frac{1}{\rho_q A} \vec{\tau} \wedge \vec{B} = i \vec{\tau} \wedge \vec{B} dx$$



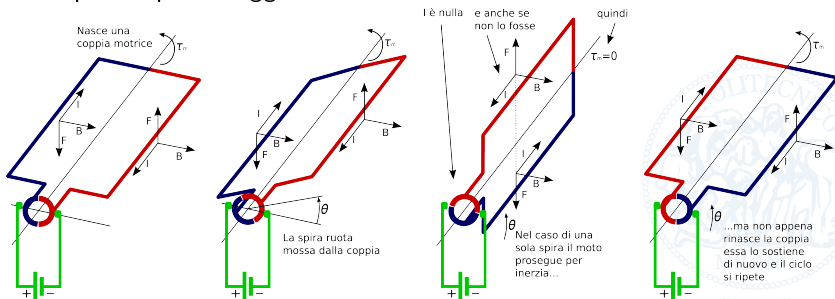
Il motore DCMP

Principio di funzionamento

- Se ora si considera un conduttore esteso *rettilineo* Γ di lunghezza L , percorso da corrente \vec{I} (la direzione del vettore \vec{I} essendo data dal conduttore stesso) e immerso in un campo magnetico B , esso risulta complessivamente soggetto alla forza

$$\vec{F} = \int_{\Gamma} d\vec{F} = \int_0^L \vec{I} \wedge \vec{B} dx = L \vec{I} \wedge \vec{B}$$

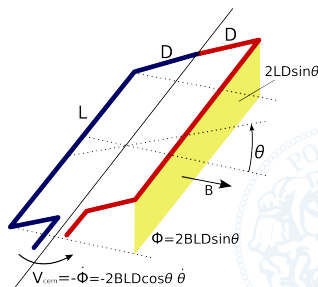
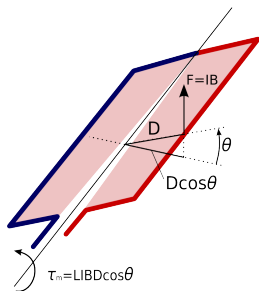
- Quindi per la spira in oggetto



Il motore DCMP

Principio di funzionamento

- Legge di Lenz: ai capi di una spira che concatena un flusso magnetico⁸ Φ variabile nasce una tensione $V = -d\Phi/dt$.
- Quindi non soltanto la corrente nella spira genera coppia, ma per effetto della rotazione nasce ai capi della spira stessa una forza controelettromotrice, ovvero una tensione V_{cem} che si oppone alla corrente nella spira.



⁸Il flusso magnetico è dato dal campo per la superficie (ortogonale al campo) della spira che lo concatena.

Il motore DCMF

Principio di funzionamento

- In un motore 'vero' il rotore é assimilabile a un cilindro ricoperto di spire con la superficie passante per l'asse del cilindro, elettricamente tutte in serie in quanto prodotte da un solo avvolgimento (a volte ci sono piú sistemi di spire, ossia avvolgimenti, ma per noi questo é inessenziale).
- Quindi ci sono sempre spire che generano coppia e non c'è di fatto bisogno che in alcun tratto il moto proceda per inerzia (di fatto anche la geometria del campo magnetico é in generale piú complessa del nostro schema di principio, ma neppure questo é essenziale per noi in questo corso).
- Supponendo che ogni spira copra un angolo infinitesimo $d\varphi$ del rotore, si ha quindi (fissati geometria e campo magnetico)

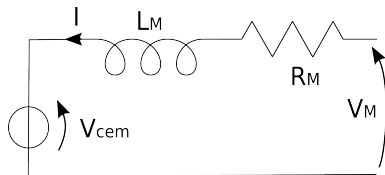
$$\begin{aligned}\tau_m &= \int_{-\pi/2}^{\pi/2} L I B D \cos \varphi d\varphi = \dots = k_\tau I, \\ V_{cem} &= \int_{-\pi/2}^{\pi/2} 2 B L D \cos \varphi \dot{\theta} d\varphi = \dots = k_\omega \dot{\theta},\end{aligned}$$

dove $\dot{\theta}$ é evidentemente la velocità angolare del rotore, indicata anche con ω (dove la scelta del simbolo k_ω). Maggiori dettagli (e un po' di geometrie piú realistiche) sui testi.

Il motore DCMP

Modello

- Il circuito elettrico che descrive il motore DCMP è



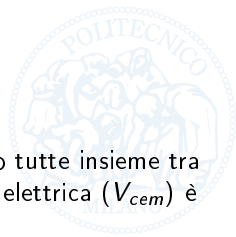
dove (con i versi indicati la costante k_ω ha segno positivo)

$$V_{cem} = k_\omega \dot{\theta}, \quad \tau_m = k_\tau I.$$

- Risulta poi ovviamente

$$V_M = V_{cem} + (R_M + sL_M)I.$$

- Come si vede le equazioni che del modello, che scriveremo tutte insieme tra un po', sono *elettromeccaniche*, in quanto una grandezza elettrica (V_{cem}) è legata a una meccanica ($\dot{\theta}$).



Il motore DCMP

Modello

- Il motore ha anche un comportamento meccanico, naturalmente. Detti J_M il suo momento d'inerzia (riferito all'asse del rotore) e D_M il suo coefficiente d'attrito dinamico, ovvero la costante di proporzionalità tra la coppia d'attrito, sempre opposta al moto, e la velocità angolare (non trattiamo qui fenomeni legati all'attrito statico perché sono di una complessità che esula dal corso) si ha

$$J_M \ddot{\theta} = \tau_M - D_M \dot{\theta}.$$

- Quindi, mettendo tutto insieme, il modello del motore DCMP è

$$\begin{aligned} J_M \ddot{\theta} &= \tau_M - D_M \dot{\theta} \\ \tau_M &= k_\tau I \\ V_M &= V_{cem} + R_M I + L_M \dot{I} \\ V_{cem} &= k_\omega \dot{\theta} \end{aligned}$$

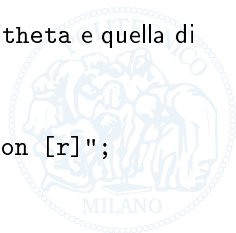


Modello Modelica del motore DCMP

Premessa

- Per scrivere modelli di elementi in cui avvengono fenomeni meccanici occorrono evidentemente terminali meccanici.
- Qui ci limitiamo a definire il *connector* MechJoint, rappresentante una giunto meccanico che connette due elementi rotanti (alberi) in modo da trasmettere il moto.
- Connettendo insieme due alberi con un giunto (che pensiamo rigido, dato che le eventuali elasticità saranno descritte entro gli elementi da esso connessi)
 - le due posizioni angolari si eguagliano,
 - mentre l'effetto di ognuno dei due elementi sull'altro si può descrivere con una coppia, uguale e opposta a quella che si dovrebbe introdurre per descrivere l'effetto dell'altro elemento sul primo considerato.
- Quindi il *connector* MechJoint avrà la variabile di sforzo theta e quella di flusso tau e sarà definito come

```
connector MechJoint
    Real theta      "angular position [r]";
    flow Real tau    "torque [Nm]";
end MechJoint;
```



Modello Modelica del motore DCM

```
model DCMotorMP
  Pin a,b;
  MechJoint shaft;
  Real theta,thetadot,tauM,VM,I;
  parameter Real RM      = 20    "winding resistance [ohm]";
  parameter Real LM      = 1e-7  "winding inductance [H]";
  parameter Real JM      = 1e-4  "inertia [kgm^2]";
  parameter Real DM      = 0.01  "angular friction coeff. [Nms/r]";
  parameter Real ktau    = 0.1   "torque/current ratio [Nm/A]";
  parameter Real komega  = 0.1   "Vcem/speed ratio [Vs/r]";

  equation
    theta          = shaft.theta;
    I              = a.i;
    VM             = a.v-b.v;
    a.i+b.i        = 0;
    der(theta)     = thetadot;
    JM*der(thetadot) = tauM-DM*thetadot+shaft.tau;
    tauM           = ktau*I;
    VM             = komega*thetadot+RM*I+LM*der(I);
end DCMotorMP;
```



Modelli di altri elementi meccanici



Altri modelli meccanici

Premessa

- La libreria `Modelica.Mechanics` e le sue sottolibrerie contengono già i modelli che ci occorrono. Noi qui li scriviamo da zero per evidenti fini didattici, visto che siamo a Ingegneria, ma se in futuro ve ne serviranno altri andate a guardare là.
- Tuttavia ricordatevi sempre che usando per la simulazione modelli di cui non si sono capiti *a fondo*
 - i fenomeni fisici descritti,
 - le equazioni,
 - il significato dei parametri
 - e soprattutto le ipotesi sottostanti e i conseguenti limiti di validità

la probabilità di ottenere 'numeri del lotto' è elevatissima. Non esiste nessun sistema che metta in grado di simulare un sistema e valutare criticamente i risultati chi non capisce di modellistica e non conosce abbastanza la 'fisica' di quel sistema; la presenza dell'avverbio 'abbastanza' viene dal fatto che a volte simulare un sistema aiuta a capire meglio come funzione, ma bisogna - appunto - essere certi di saper valutare i risultati della simulazione.

- Detto ciò, scriviamo un paio di modelli che ci serviranno.

Altri modelli meccanici

Carico e giunto elastico

- Carico meccanico (rotazionale):
oggetto di momento d'inerzia J_L e coefficiente d'attrito rotazionale D_L , dotato di una flangia meccanica per connetterlo ad altri elementi.
- Giunto elastico (rotazionale):
oggetto con due flange senza inerzia né attrito propri, caratterizzato da una relazione lineare tra la coppia e lo scostamento angolare tra le due flange, dove E_J è la costante di proporzionalità tra essi; in sintesi, il giunto in questione è l'analogo rotazionale di una molla lineare.
- Provate a scrivere i modelli di questi due elementi direttamente in Modelica: tra qualche minuto lo facciamo insieme.



Modelli Modelica di carico e giunto elastico

```
model MechLoad
  MechJoint shaft;
  Real theta,thetadot;
  parameter Real JL = 5e-4 "inertia [kgm^2]";
  parameter Real DL = 0.05 "angular friction coeff. [Nms/r]";
equation
  theta = shaft.theta;
  der(theta) = thetadot;
  JL*der(thetadot) = shaft.tau-DL*thetadot;
end MechLoad;

model ElasticJoint
  MechJoint flangea,flangeb;
  Real dtheta;
  parameter Real EJ = 2 "angular elasticity [Nm/r]";
equation
  dtheta = flangea.theta-flangeb.theta;
  0 = flangea.tau+flangeb.tau;
  flangea.tau = EJ*dtheta;
end ElasticJoint;
```

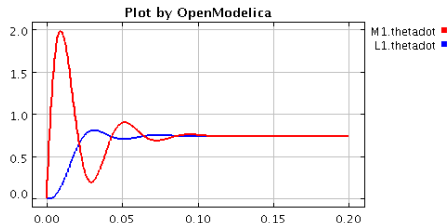


Semplice simulatore di catena meccanica

Costruiamo e simuliamo una semplice catena meccanica (elettromeccanica, a voler essere precisi...) per provare i modelli scritti:

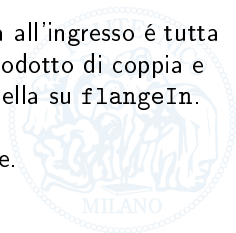
```
model ExampleMech01
  IdealVoltageGenerator VG1;
  Ground                G;
  DCMotorMP             M1;
  ElasticJoint          J1;
  MechLoad              L1;

  equation
    connect(VG1.b,G.a);
    connect(VG1.a,M1.a);
    connect(M1.b,G.a);
    connect(M1.shaft,J1.flangea);
    connect(J1.flangeb,L1.shaft);
    VG1.Vg = 9;
end ExampleMech01;
```



Modello del riduttore ideale

- Costruiamo un ultimo modello, ovvero quello di un riduttore ideale (cioè rigido e senza perdite).
- Si tratta evidentemente di un oggetto con due MechJoint, in cui le due velocità angolari stanno tra di loro nel rapporto (fisso, perché il riduttore è rigido) $1 : n$, dove n è appunto detto 'rapporto di riduzione'. Per chiarezza definiamo i nomi in modo da evidenziare il verso di montaggio dell'elemento, cioè chiamiamo `flangeIn` la flangia attaccata al motore e `flangeOut` quella attaccata al carico, intendendo (si tratta appunto di un riduttore) che `flangeOut` fa un giro ogni n giri di `flangeIn`.
- Essendo il riduttore privo di perdite, la potenza meccanica all'ingresso è tutta trasferita in uscita. Dato che la potenza meccanica è il prodotto di coppia e velocità angolare, la coppia su `flangeOut` sarà n volte quella su `flangeIn`.
- Scrivete voi il modello. Tra un minuto lo facciamo insieme.



Modello Modelica del riduttore ideale

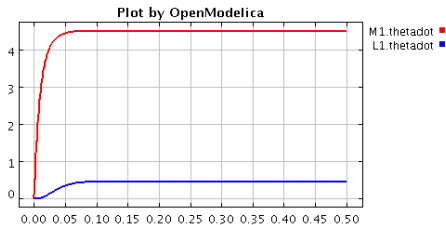
```
model IdealGearbox
  MechJoint flangeIn, flangeOut;
  parameter Real n = 10 "gearbox ratio [#]";
equation
  flangeIn.theta = n*flangeOut.theta;
  flangeIn.tau    = flangeOut.tau/n;
end IdealGearbox;
```



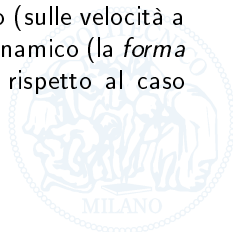
Un altro semplice simulatore di catena meccanica

Inseriamo un riduttore 10:1 nella catena già simulata e vediamo l'effetto:

```
model ExampleMech02
  IdealVoltageGenerator VG1;
  DCmotorMP M1;
  IdealGearbox GB1;
  ElasticJoint J1;
  MechLoad L1;
  Ground G;
equation
  connect(VG1.b,G.a);
  connect(VG1.a,M1.a);
  connect(M1.b,G.a);
  connect(M1.shaft,GB1.flangeIn);
  connect(GB1.flangeOut,J1.flangea);
  connect(J1.flangeb,L1.shaft);
  VG1.Vg = 9;
end ExampleMech02;
```



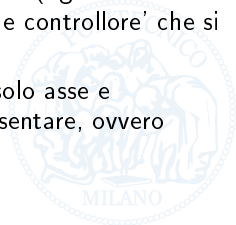
Si vede un effetto statico (sulle velocità a regime) ma anche uno dinamico (la *forma* della risposta è diversa rispetto al caso precedente): perché?



Il controllo di un asse motorizzato

Cenni introduttivi

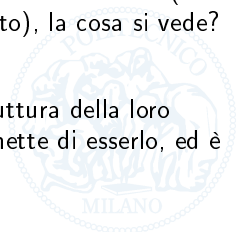
- Supponiamo di aver ridotto il controllo del moto dell'organo di nostro interesse a quello di un certo numero di assi mossi ognuno da un motore (per noi DCMP): questo è sempre possibile, almeno finché il moto è appunto generato da motori rotativi.
- Supponiamo anche che tali assi siano indipendenti. Questo molto spesso non è vero, perché se ad esempio un braccio robotico muove un giunto (pensate al vostro gomito per analogia) questo moto disturba inerzialmente gli altri giunti (pensate, sempre per analogia, alla spalla) e richiede quindi un'azione su di essi, sia che vi siano tanti controllori tutti indipendenti (ogni controllore vede soltanto in suo giunto) sia che vi sia un 'unico grande controllore' che si occupa di tutti i giunti in modo coordinato.
- Noi, però, in questo corso ci limitiamo al controllo di un solo asse e consideriamo le due situazioni tipiche che si possono presentare, ovvero
 - il controllo di posizione
 - e il controllo di velocità.



Il controllo di un asse motorizzato

Cenni introduttivi

- Facciamoci subito un'idea di com'è strutturata la dinamica del sistema da controllare:
 - il controllore imprime al motore una tensione
 - che diventa una corrente a meno della forza controlettromotrice;
 - la corrente produce una coppia motrice
 - che, diminuita di coppia resistente e d'attrito (totali) accelera l'inerzia (totale):
 - ne risulta una velocità
 - il cui integrale è la posizione.
- Ora, se diciamo che ci sono due retroazioni di velocità interne al sistema (una tramite la forza controlettromotrice e una tramite l'attrito), la cosa si vede? Se no discutiamone e capiamo...
- Allenatevi a guardare i sistemi e descrivere a parole la struttura della loro dinamica: all'inizio è difficile ma con un po' d'esercizio smette di esserlo, ed è tremendamente utile (non solo per il controllo).

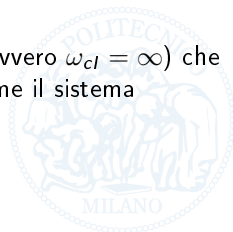


Il controllo di un asse motorizzato

Cenni introduttivi

- Pensiamo quindi per prima cosa a come si potrebbe fare il controllo di velocità.
- Anzitutto, controllando la corrente, posso far sí
 - che la coppia motrice risenta delle dinamiche elettriche date da resistenze e induttanza meno di quanto farebbe senza un tale controllo
 - e anche che tale coppia non risenta della forza controlettromotrice V_{cem} , che agisce su di essa come un disturbo (in andata).
- Quindi per prima cosa posso misurare la corrente I e fare in modo che, fino a una data frequenza ω_{cl} , essa segua il suo riferimento I^o .
- Questo fa sí, supponendo il controllo di corrente ideale (ovvero $\omega_{cl} = \infty$) che il motore con attorno l'anello di corrente' si comporti come il sistema

$$J_M \ddot{\theta} = k_t I^o - D_M \dot{\theta}.$$



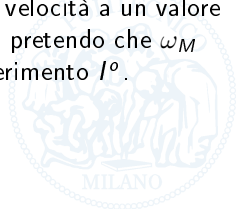
Il controllo di un asse motorizzato

Cenni introduttivi

- Ora, la dinamica dal riferimento di corrente I^o alla velocità $\dot{\theta}$ è evidentemente descritta da un sistema del prim'ordine asintoticamente stabile, dato che ponendo per leggibilità $\omega_M := \dot{\theta}$ la relazione precedente diviene

$$\dot{\omega}_M = -\frac{D_M}{J_M}\omega_M + \frac{k_\tau}{J_M}I^o.$$

- Il controllo di velocità è quindi a questo punto relativamente semplice. Basta ad esempio un PI, pur di porre la banda $\omega_{c\omega}$ dell'anello di velocità a un valore molto minore di ω_{cl} , in modo che alle frequenze alle quali pretendo che ω_M segua il suo riferimento ω_M^o sia vero che I segue il suo riferimento I^o .



Il controllo di un asse motorizzato

Cenni introduttivi

- Pensiamo ora al controllo di posizione.
- Supponendo ideali i controlli di corrente e velocità, ovvero $\omega_{c\omega} = \omega_{cI} = \infty$ è ovvio che la dinamica dal riferimento di velocità ω_M^o alla posizione θ è un semplice integratore.
- Quindi, per controllare θ basta chiudere attorno al sistema con ingresso ω_M^o e uscita $\omega_M = \dot{\theta}$ un anello con un controllore proporzionale per avere errore nullo a transitorio esaurito.
- Se la banda ω_θ di tale anello, quella ω_ω dell'anello di posizione e quella ω_I dell'anello di corrente sono tali che

$$\omega_\theta \ll \omega_\omega \ll \omega_I$$

ogni anello richiede che il suo riferimento sia inseguito dalla sua uscita fino a frequenze alle quali l'anello più interno (chiuso) ha funzione di trasferimento (circa) pari a 1.



Il controllo di un asse motorizzato

Cenni introduttivi

- Conclusioni:

- il controllo del motore si fa chiudendo tre anelli uno attorno all'altro:
 - prima uno di corrente,
 - poi uno di velocità,
 - e infine, se serve, uno di posizione;
 - ovviamente così servono tre misure, il che può essere problematico (ci sono quindi anche schemi di controllo semplificati);
 - con scelte opportune delle bande dei tre anelli si può pervenire a una taratura dei regolatori più semplice di quanto lo sarebbe se si tentasse di fare tutto con un solo regolatore, dato che ogni anello 'non vede' le dinamiche controllate da quello più interno.
-
- Riprenderemo l'argomento dopo aver visto le 'strutture di controllo', il che ci permetterà di usare una terminologia rigorosa al posto di una approssimativa come il termine 'tre anelli uno attorno all'altro'. A quel punto il procedimento per la sintesi dei regolatori degli anelli sarà semplice da capire.

- Tradurre il modello del motore DCMP in uno schema a blocchi e 'vedere' in tale schema la connessione tra fenomeni elettrici e meccanici che ci ha fatto dire che essi non sono 'separabili' (lo schema sarà discusso in classe più avanti, al momento di parlare di controllo del motore, ma provare a farlo da soli è molto istruttivo).
- Modificare l'esempio ExampleMech02 in modo che il giunto elastico sia 'a monte' del riduttore, ossia tra il motore e il riduttore. Simulare il modello, vedere cosa cambia rispetto all'esempio visto prima e cercare di spiegare il perché dei cambiamenti in modo rigoroso.
- Scrivere in OpenModelica il modello di un sistema in cui un motore DCMP tira, con una fune rigida e assunta sempre tesa avvolta su una puleggia, una massa che scorre con attrito su una guida orizzontale ed è vincolata a un punto fisso tramite una molla con elasticità lineare situata dalla parte opposta al motore (occorrerà scrivere il modello di qualche altro componente e strutturare il tutto in modo ordinato). Simulare il modello e riflettere sul ruolo dei vari parametri fisici.

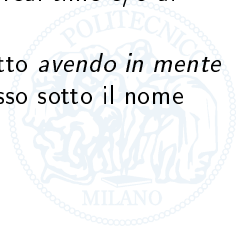
Dimensionamento di organi di controllo



Dimensionamento di organi di controllo

Premesse

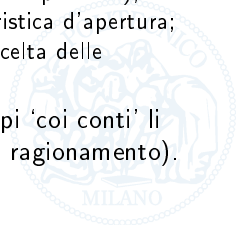
- Evidentemente, gli organi di controllo (principalmente sensori e attuatori) vanno 'dimensionati', ovvero 'scelti in modo che ce la facciano in modo affidabile e che non costino troppo'.
- A rigore vanno dimensionati anche i regolatori (numero e potenza delle CPU, caratteristiche della rete, etc.) ma in questo corso non ne parliamo, perché non ne abbiamo il tempo e anche perché questa è la parte della materia meglio coperta in altri corsi non necessariamente orientati in modo prevalente al controllo (ad esempio quelli relativi ai sistemi operativi real-time e/o ai sistemi embedded).
- A rigore va dimensionato anche il processo e questo va fatto *avendo in mente che lo si vorrà controllare in un certo modo*, il che va spesso sotto il nome 'process-control codesign'.



Dimensionamento di organi di controllo

Premesse

- Dato che tutto va dimensionato, non si può dare un insieme esaustivo di metodi.
- Occorre invece capire come si ragiona per dimensionare un organo di controllo (ci limitiamo a questi, abbiamo detto).
- Coerentemente con gli obiettivi del corso, vedremo alcuni esempi contando che da essi si comprenda il modo di ragionare testé nominato.
- Gli esempi che vedremo sono
 - il dimensionamento di una valvola regolante (per fluidi incompressibili), in termini di scelta del coefficiente d'efflusso e della caratteristica d'apertura;
 - il dimensionamento di un motore elettrico, in termini di scelta delle caratteristiche di coppia e inerzia.
- Vedremo ora il solo ragionamento, come anticipato: esempi 'coi conti' li faremo a esercitazione (ovvero quando avrete assimilato il ragionamento).



Dimensionamento di una valvola regolante

Scelta del coefficiente d'efflusso

- Consideriamo una valvola inserita in un generico circuito idraulico.
- Dai dati di progetto del circuito si è in grado di stabilire, eventualmente con qualche ragionamento,
 - quali sono i valori nominali del salto di pressione tra monte e valle della valvola e della portata (massica o volumetrica) che essa deve far passare.
 - oppure un 'caso pessimo' dal punto di vista della capacità di efflusso, ovvero la massima portata che deve passare e il minimo salto di pressione che può cadere sulla valvola.
- In ambo i casi si ha una coppia (salto di) pressione/portata ($\overline{\Delta p}, \overline{w}$ oppure $\overline{\Delta p}, \overline{q}$) nominale o 'di progetto'.
- Per scegliere Cv_{max} si calcola allora il coefficiente d'efflusso nominale come

$$Cv_{nom} = \frac{\overline{w}}{\sqrt{\overline{\Delta p}}} \quad \text{oppure} \quad Cv_{nom} = \frac{\overline{q}}{\sqrt{\overline{\Delta p}}}$$

e poi lo si aumenta convenzionalmente del 30% (circa), ottenendo

$$Cv_{max} = Cv_{nom}/0.7.$$

Dimensionamento di una valvola regolante

Scelta del coefficiente d'efflusso

- Determinato Cv_{max} si cerca sui cataloghi una valvola
 - con un valore di Cv_{max} vicino,
 - in grado di reggere i salti di pressione richiesti,
 - adatta al fluido da trattare
 - e con un valore del coefficiente di recupero che non renda troppo bassa la pressione di vene contratta.
- Scelto così il coefficiente d'efflusso, resta da determinare la caratteristica intrinseca $\Phi(x)$ della valvola.



Dimensionamento di una valvola regolante

Scelta della caratteristica intrinseca

- Determinato Cv_{max} (limitiamoci al caso con portata massica, nel caso di fluido incomprimibile il caso con portata volumetrica è identico) si sa che

$$w = Cv_{max} \Phi(x) \sqrt{p_m - p_v}$$

dove p_m e p_v sono le pressioni a monte e a valle della valvola.

- Chiamiamo ora $w(x)$ la *caratteristica installata*, ovvero la relazione tra x e la portata che la valvola deve modulare.
- Di solito questa è la portata nella valvola stessa, ma non sempre è così (vedremo a esercitazione un esempio con pompa volumetrica e valvola di ricircolo, che è un caso abbastanza tipico).
- Per ora supponiamo comunque che la portata da modulare sia quella nella valvola medesima, dato che i concetti da introdurre si vedono tutti lo stesso.
- Il nostro scopo è scegliere $\Phi(x)$ in modo che $w(x)$ sia lineare, ovvero $w(x) = kx$.

Dimensionamento di una valvola regolante

Scelta della caratteristica intrinseca

- Se p_m e p_v sono impresse, e in particolare non dipendono da w , la caratteristica $w(x)$ ha evidentemente (a salto di pressione dato) la stessa forma di $\Phi(x)$, quindi $\Phi(x)$ lineare produce $w(x)$ lineare. Se invece p_m e p_v dipendono da w , la forma di $w(x)$ non è la stessa di $\Phi(x)$.
- In tal caso per determinare $\Phi(x)$
 - si esprimono le dipendenze di p_m e p_v da w , dalle condizioni al contorno e dagli altri elementi del circuito, costanti che indichiamo collettivamente con θ , scrivendo

$$p_m = p_m(w, \theta), \quad p_v = p_v(w, \theta);$$

- si scrive

$$w = C_{v_{max}} \Phi(x) \sqrt{p_m(w, \theta) - p_v(w, \theta)}$$

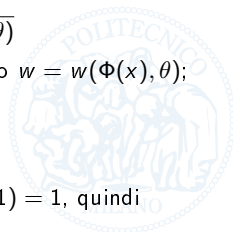
e si risolve rispetto a w , ottenendo un'espressione del tipo $w = w(\Phi(x), \theta)$;

- si pone

$$w(\Phi(x), \theta) = kx$$

e si risolve rispetto a $\Phi(x)$;

- infine per determinare k si ricorda che per definizione $\Phi(1) = 1$, quindi ponendo $x = 1$ risulta semplicemente $k = w(1, \theta)$.



Dimensionamento di una valvola regolante

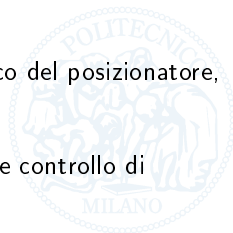
Complementi

- Scelta la valvola c'è anche da scegliere il posizionatore, ovvero l'organo che muove materialmente lo stelo secondo il comando.
- I posizionatori hanno ovviamente una loro dinamica, che non descriviamo per ragioni di tempo. Tuttavia non è opportuno trascurarla completamente, quindi facciamo quel che si fa nelle applicazioni dove di tale dinamica basta un modello di massima, ovvero supponiamo che la posizione x della valvola dipenda dalla posizione \bar{x} richiesta dal sistema di controllo secondo la legge

$$X(s) = \frac{1}{1 + sT_p} \bar{X}(s)$$

dove la costante di tempo T_p è un parametro caratteristico del posizionatore, da scegliersi coerentemente con gli obiettivi del controllo.

- Vedremo qualche esempio di dimensionamento di valvole e controllo di portata a esercitazione.



Dimensionamento di un motore elettrico

Premesse

- Il problema ha un aspetto meccanico e uno elettrico.
- L'aspetto meccanico si riconduce a 'riportare all'albero motore' (vedremo tra poco che significa) le caratteristiche del sistema da muovere, scegliendo poi un motore che sia in gradi di farlo.
- L'aspetto elettrico, dato quel che il motore deve fare meccanicamente, si riconduce a dimensionare il 'drive' del motore stesso in modo che possa fornire l'alimentazione richiesta.
- Dato che lo scopo qui è insegnare come si ragiona per dimensionare, non fare una casistica esaustiva che non avrebbe alcun senso, ci limitiamo a considerare l'aspetto meccanico.



Dimensionamento di un motore elettrico

Scelta delle caratteristiche meccaniche del motore

- Il sistema da muovere è in generale composto da un cinematismo che trasmette il moto del motore a un carico, sottoposto a volte a coppie esterne.
- Se quindi 'ci mettiamo nei panni del motore' tale sistema si tradurrà per noi in un momento d'inerzia totale (o 'equivalente all'albero motore') del carico J_L e in un coefficiente d'attrito pure totale (o 'equivalente all'albero motore') del carico D_L , dove 'L' sta per 'load'. Inoltre, le coppie esterne applicate al carico possono anch'esse riportarsi all'albero motore: indichiamo qui con τ_L il risultato.
- Il calcolo di J_L e D_L e la caratterizzazione di τ_L possono essere un compito anche molto complesso e non vi entriamo: vedremo soltanto un esempio abbastanza semplice a esercitazione, giusto per capire di cose si tratta.
- Dati comunque J_L e D_L e detti J_M e D_M inerzia e attrito propri del motore, il sistema complessivo è retto (supponendo opportunamente caratterizzata la coppia τ_L) dall'equazione

$$(J_M + J_L)\ddot{\theta}_M = \tau_M - \tau_L - (D_M + D_L)\dot{\theta}_M$$

dove θ_M è l'angolo motore.

Dimensionamento di un motore elettrico

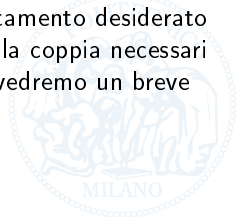
Scelta delle caratteristiche meccaniche del motore

- Riconducendo il problema al controllo dell'angolo motore, la conoscenza dei requisiti di tale controllo e delle caratteristiche del carico permette di determinare congiuntamente, al limite per tentativi

- la coppia motrice τ_M necessaria,
- l'inerzia e l'attrito del motore,

in modo da ottenere transitori di velocità e/o posizione con le caratteristiche richieste.

- La disponibilità di modelli a-causali è molto utile perché tali modelli possono essere invertiti, ovvero simulati imponendo loro il comportamento desiderato (ad esempio la posizione) e ricavandone gli andamenti della coppia necessari all'ottenimento di tale comportamento. Anche di questo vedremo un breve esempio a esercitazione.



Dimensionamento di organi di controllo

Scrittura di modelli per il controllo di processo e del moto

Questa esercitazione è svolta usando la lavagna;
non ci sono slide guida.



Lezione 6 (4L)

**Principali strutture di controllo:
feedforward, cascata, disaccoppiamento, predittore di Smith**

Breve introduzione a Scilab



Principali strutture di controllo



Principali strutture di controllo

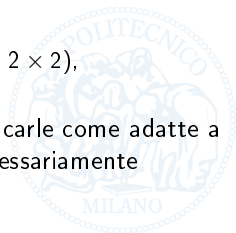
Premessa

- Finora avete visto sistemi di controllo ad anello singolo.
- E' evidente che i sistemi di controllo "veri" sono molto più complessi e articolati, non foss'altro perché in un impianto le variabili da controllare sono tante e l'impianto è "interagente", nel senso che non è vero che agendo su una variabile di controllo s'influenza una sola variabile controllata.
- Quindi gli schemi di controllo in generale contengono molti blocchi, combinati in vari modi.
- La costruzione di tale insieme di blocchi interconnessi è detta "strutturazione" del controllo e spesso è la parte più difficile e critica del problema.
- A volte qualcuno dice che una volta strutturato lo schema di controllo il lavoro è praticamente fatto perchè poi "si tratta solo di mettere i numeri nei blocchi" (ovvero di *parametrizzarli*).
- Detto così è un po' esagerato ma non è del tutto falso: se lo schema è stato strutturato bene, parametrizzare i blocchi è certamente molto più facile, mentre se la strutturazione è inadatta al problema la parametrizzazione può diventare quasi impossibile.

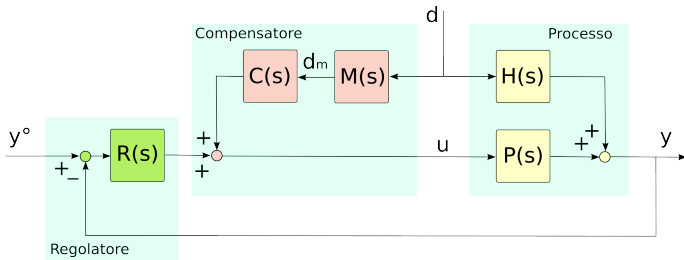
Principali strutture di controllo

Premessa

- Nello strutturare uno schema di controllo, la fisica dei fenomeni e le dinamiche che ne conseguono tendono a far sì che i blocchi si combinino in “modi tipici”; un paragone forse un po’ azzardato ma suggestivo è che se il blocco fosse la singola lettera e lo schema complessivo la frase, allora quei “modi tipici di combinare i blocchi” sarebbero le parole.
- I “modi tipici” di cui sopra si dicono *strutture di controllo*. Conoscerle e saperle usare è molto importante.
- Qui vedremo alcune delle principali strutture di controllo, ovvero
 - la compensazione in anello aperto,
 - il controllo in cascata,
 - il controllo multivariabile con disaccoppiamento (nel caso 2×2),
 - il predittore di Smith.
- Impareremo anche a cosa esse servono, cercando di qualificarle come adatte a un problema con certe caratteristiche *dinamiche* (non necessariamente fisiche).



Compensazione in anello aperto (feedforward)



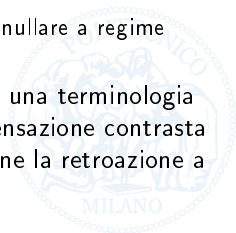
- Quando c'è un disturbo in andata $d(t)$ e se ne può ottenere una misura $d_m(t)$ tramite un *misuratore* $M(s)$, ha senso tentare di annullare - o quantomeno ridurre - la sua influenza su y con un *compensatore in anello aperto* $C(s)$.
- A tale scopo occorre calcolare $C(s)$ in modo che la funzione di trasferimento da d a y sia nulla.
- Fatelo (avete 1 minuto) e poi ne discutiamo.

Compensazione in anello aperto

- Risulta

$$\frac{H(s) + M(s)C(s)P(s)}{1 + R(s)P(s)} = 0, \quad \Rightarrow C(s) = -\frac{H(s)}{M(s)P(s)}$$

- Prima osservazione: $R(s)$ non c'entra. Ovviamente la retroazione aiuta a rigettare d ma lo fa soltanto quando i suoi effetti si sono manifestati su y , mentre la compensazione agisce *subito*. Pensando allora ad esempio a un disturbo costante,
 - la compensazione agisce prontamente ma funziona soltanto se il modello è preciso (opera in anello aperto),
 - mentre la retroazione non agisce prontamente ma può annullare a regime l'effetto del disturbo anche a fronte di errori di modello.
- In generale, retroazione e compensazione cooperano. Con una terminologia non del tutto rigorosa ma espressiva si dice che “la compensazione contrasta il disturbo in modo rapido ma approssimato e poi interviene la retroazione a sistemare le cose a regime”.



Compensazione in anello aperto

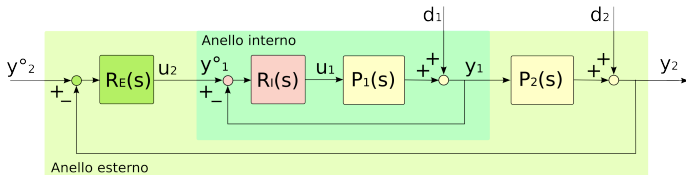
- Seconda osservazione: il compensatore trovato è “ideale”, e infatti scriveremo

$$C_{ID}(s) = -\frac{H(s)}{M(s)P(s)},$$

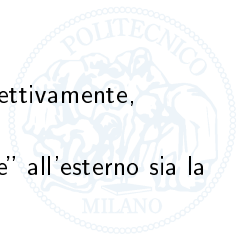
perché

- potrebbe avere più zeri che poli
 - e/o potrebbe avere poli nel semipiano destro.
- In tali casi occorre ottenere dal compensatore ideale $C_{ID}(s)$ un compensatore reale $C_R(s)$
 - togliendo zeri e/o aggiungendo poli
 - e comunque *omettendo* eventuali poli nel semipiano destro.
- Facendo così si avrà un compensatore “buono fino a una certa frequenza”, quella dove $C_R(j\omega)$ inizia a differire in modo significativo da $C_{ID}(j\omega)$, considerando *sia il modulo che la fase*.
- Vedremo più avanti degli esempi (ma non pensate di capire con quelli: se qualcosa non è chiaro ditelo adesso).

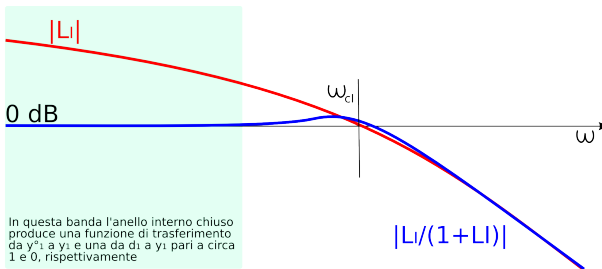
Controllo in cascata (cascade control)



- Quando nel sistema di controllo c'è una variabile y_1 che
 - risponde al segnale di controllo u_1 più rapidamente (in senso dinamico) di quanto faccia la variabile controllata y_2
 - e si può misurare (il che, sappiate e ricordate, costa)allora può convenire usare lo schema illustrato.
- Un caso tipico è quello in cui P_1 e P_2 rappresentano, rispettivamente, l'attuatore e il processo vero e proprio.
- Il principio è avere un anello interno veloce, che “nasconde” all'esterno sia la dinamica di P_1 che il disturbo d_1 .

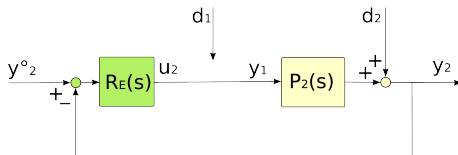


Controllo in cascata



- La figura illustra il principio di funzionamento dello schema in cascata, ponendo $L_I = R_I P_1$ e indicando con ω_{cI} e ω_{cE} le pulsazioni critiche degli anelli interno ed esterno
- Ai fini pratici occorre una separazione minima di mezza decade (meglio una) tra i due anelli, ovvero che ω_{cI} sia almeno 5 (meglio 10) volte più grande di ω_{cE} . In tale ipotesi ha senso calcolare ω_{cE} usando come funzione di trasferimento d'anello esterno $R_E P_2$

Controllo in cascata



- Infatti, nel caso ideale, l'anello interno chiuso produce la situazione qui mostrata.
- Per la sintesi
 - prima si determina R_I basandosi sul solo P_1
 - e quindi si sintetizza R_E basandosi sul solo P_2 , mantenendo la separazione di banda richiesta.
- E' ora utile analizzare lo schema un po' più a fondo. Calcolate le 6 funzioni di trasferimento dagli ingressi y_2^o , d_1 e d_2 alla variabile controllata y_2
 - con lo schema in cascata idealizzato (T_{ci} , H_{ci} , K_{ci}),
 - e con quello reale (T_{cr} , H_{cr} , K_{cr}),
 - e senza cascata, ossia con un solo regolatore R agente su u_1 (T_{nc} , H_{nc} , K_{nc}).
- Avete 5 minuti e poi ne discutiamo.

- Cascata con anello interno idealizzato:

$$T_{ci}(s) := \left. \frac{Y(s)}{Y^o(s)} \right|_{\text{a.i. ideale}} = \frac{R_E(s)P_2(s)}{1+R_E(s)P_2(s)}$$

$$H_{ci}(s) := \left. \frac{Y(s)}{D_1(s)} \right|_{\text{a.i. ideale}} = 0$$

$$K_{ci}(s) := \left. \frac{Y(s)}{D_2(s)} \right|_{\text{a.i. ideale}} = \frac{1}{1+R_E(s)P_2(s)}$$

- Cascata con anello interno reale:

$$T_{cr}(s) := \left. \frac{Y(s)}{Y^o(s)} \right|_{\text{a.i. reale}} = \frac{R_E(s)R_I(s)P_1(s)P_2(s)}{1+R_I(s)P_1(s)(1+R_E(s)P_2(s))}$$

$$H_{cr}(s) := \left. \frac{Y(s)}{D_1(s)} \right|_{\text{a.i. reale}} = \frac{P_2(s)}{1+R_I(s)P_1(s)(1+R_E(s)P_2(s))}$$

$$K_{cr}(s) := \left. \frac{Y(s)}{D_2(s)} \right|_{\text{a.i. reale}} = \frac{1+R_I(s)P_1(s)}{1+R_I(s)P_1(s)(1+R_E(s)P_2(s))}$$

Controllo in cascata (cascade control)

- Osservazione importante:

$$|R_I P_1| \rightarrow \infty \quad \Rightarrow \quad \begin{cases} T_{cr} \rightarrow T_{ci} \\ H_{cr} \rightarrow H_{ci} \\ K_{cr} \rightarrow K_{ci} \end{cases}$$

- Senza il controllo in cascata, ovvero singolo anello con regolatore $R(s)$:

$$T_{nc}(s) := \left. \frac{Y(s)}{Y^o(s)} \right|_{\text{single loop}} = \frac{R(s)P_1(s)P_2(s)}{1+R(s)P_1(s)P_2(s)}$$

$$H_{nc}(s) := \left. \frac{Y(s)}{D_1(s)} \right|_{\text{single loop}} = \frac{P_2(s)}{1+R(s)P_1(s)P_2(s)}$$

$$K_{nc}(s) := \left. \frac{Y(s)}{D_2(s)} \right|_{\text{single loop}} = \frac{1}{1+R(s)P_1(s)P_2(s)}$$

- Anche per il controllo in cascata vedremo a suo tempo degli esempi.

Controllo multivariabile con disaccoppiamento (decoupling)

- Iniziamo descrivendo un *processo* LTI multivariabile. Ci limitiamo al caso 2×2 (2 ingressi e 2 uscite) per semplicità di trattazione; l'estensione al caso $n \times n$ è comunque ovvia.
- Un processo LTI MIMO (Multiple Input, Multiple Output) si descrive con una *matrice di trasferimento*, quadrata se - come nel nostro caso - ci sono tanti ingressi quante uscite. L'elemento (i, j) di tale matrice è la funzione di trasferimento dall'ingresso j -esimo all'uscita i -esima.
- Ad esempio, un processo con 2 ingressi u_1, u_2 e 2 uscite y_1, y_2 (il nostro caso) si descrive come

$$\begin{bmatrix} Y_1(s) \\ Y_2(s) \end{bmatrix} = \begin{bmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{bmatrix} \begin{bmatrix} U_1(s) \\ U_2(s) \end{bmatrix}$$

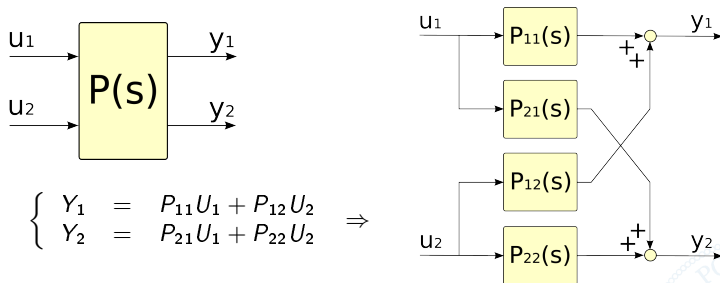
dove la matrice di trasferimento è

$$P(s) = \begin{bmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{bmatrix}$$



Controllo multivariabile con disaccoppiamento

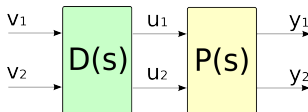
- Vediamo per maggior chiarezza la cosa in termini di schema a blocchi:



- Osservate bene lo schema, confrontatelo con l'espressione della matrice di trasferimento e accertatevi di aver capito *bene*, perché altrimenti il seguito vi risulterebbe del tutto incomprensibile.

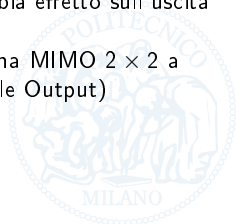
Controllo multivariabile con disaccoppiamento

- Il controllo multivariabile con disaccoppiamento affronta il problema in due passi:
 - prima si antepone a $P(s)$ una matrice di trasferimento 2×2 $D(s)$, che rappresenta il sistema detto “disaccoppiatore”, ovvero si costruisce il sistema



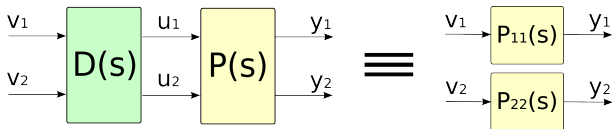
dove $D(s)$ si determina in modo che l'ingresso v_1 non abbia effetto sull'uscita y_2 e, simmetricamente, l'ingresso v_2 non ne abbia su y_1 ;

- quindi, avendo ridotto il problema di controllare un sistema MIMO 2×2 a quello di controllare due sistemi SISO (Single Input, Single Output) indipendenti, si procede con i metodi noti.



Controllo multivariabile con disaccoppiamento

- Procediamo ora alla determinazione di $D(s)$.
- Vogliamo ottenere quanto indicato in figura



ovvero, ponendo $Y' = [Y_1 \ Y_2]$, $U' = [U_1 \ U_2]$ e $V' = [V_1 \ V_2]$, che

$$Y = PU = \overset{9}{\widehat{PD}} V = \begin{bmatrix} P_{11}(s) & 0 \\ 0 & P_{22}(s) \end{bmatrix} V$$

- Allora $D(s)$ si determina, moltiplicando a sinistra per $P^{-1}(s)$ i fattori matrici dell'ultima eguaglianza, come

$$D(s) = P^{-1}(s) \begin{bmatrix} P_{11}(s) & 0 \\ 0 & P_{22}(s) \end{bmatrix}$$

⁹ATTENZIONE: il prodotto di matrici non commuta, quindi PD e *non* DP !

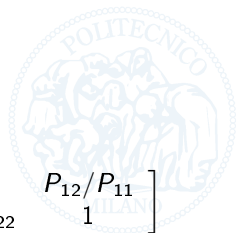
Controllo multivariabile con disaccoppiamento

- Interpretiamo il risultato:

$$D = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}^{-1} \begin{bmatrix} P_{11} & 0 \\ 0 & P_{22} \end{bmatrix}$$

quindi

$$\begin{aligned} D^{-1} &= \left(\begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}^{-1} \begin{bmatrix} P_{11} & 0 \\ 0 & P_{22} \end{bmatrix} \right)^{-1} \\ &= \begin{bmatrix} P_{11} & 0 \\ 0 & P_{22} \end{bmatrix}^{-1} \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \\ &= \frac{1}{P_{11}P_{22}} \begin{bmatrix} P_{22} & 0 \\ 0 & P_{11} \end{bmatrix} \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \\ &= \frac{1}{P_{11}P_{22}} \begin{bmatrix} P_{11}P_{22} & P_{12}P_{22} \\ P_{11}P_{21} & P_{11}P_{22} \end{bmatrix} = \begin{bmatrix} 1 & P_{12}/P_{11} \\ P_{21}/P_{22} & 1 \end{bmatrix} \end{aligned}$$



Controllo multivariabile con disaccoppiamento

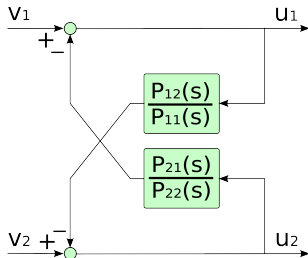
- Allora, dato che $U = DV$, è anche $V = D^{-1}U$, ovvero

$$\begin{bmatrix} V_1(s) \\ V_2(s) \end{bmatrix} = \begin{bmatrix} 1 & \frac{P_{12}}{P_{11}} \\ \frac{P_{21}}{P_{22}} & 1 \end{bmatrix} \begin{bmatrix} U_1(s) \\ U_2(s) \end{bmatrix} \Rightarrow \begin{cases} V_1 = U_1 + \frac{P_{12}}{P_{11}} U_2 \\ V_2 = \frac{P_{21}}{P_{22}} U_1 + U_2 \end{cases}$$

- Si ha quindi immediatamente

$$\begin{cases} U_1 = V_1 - \frac{P_{12}}{P_{11}} U_2 \\ U_2 = V_2 - \frac{P_{21}}{P_{22}} U_1 \end{cases}$$

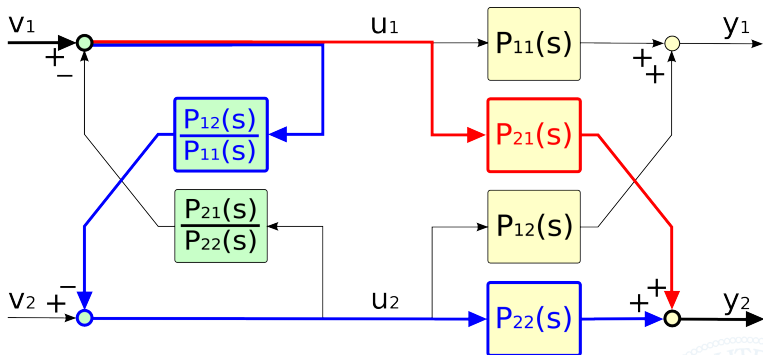
e il disaccoppiatore è pertanto descritto dallo schema a blocchi



che spiega anche perché spesso lo si chiama “disaccoppiatore all’indietro”.



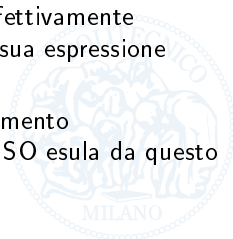
Controllo multivariabile con disaccoppiamento



- Lo schema illustra il funzionamento del disaccoppiatore all'indietro, evidenziando come esso annulli il "cammino netto" di segnale tra v_1 e y_2 (la somma di quelli rosso e blu).
- Osservatelo attentamente e ritrovatevi i concetti che abbiamo esposto. Fate domande se necessario.

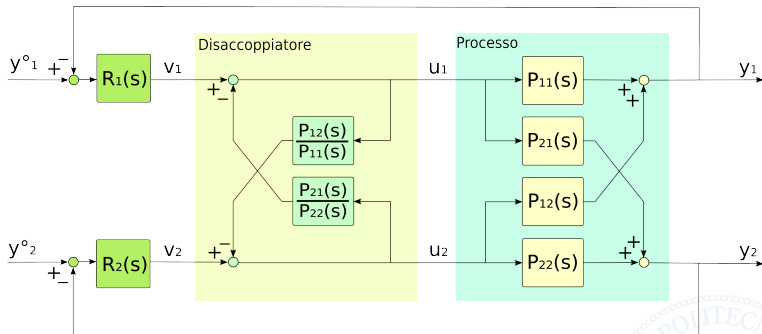
Controllo multivariabile con disaccoppiamento

- Anche qui può presentarsi il problema di realizzabilità visto per la compensazione in anello aperto. Infatti i blocchi presenti nel disaccoppiatore
 - potrebbero avere più zeri che poli
 - e/o potrebbero avere poli nel semipiano destro.
- In tali casi il disaccoppiamento ideale non è possibile e occorrerà rendere realizzabili i blocchi che non lo fossero
 - togliendo zeri e/o aggiungendo poli
 - e comunque *omettendo* eventuali poli nel semipiano destro.
- Facendo così si avrà un disaccoppiamento “buono fino a una certa frequenza”, quella dove almeno uno dei blocchi del disaccoppiatore effettivamente implementato inizia a differire in modo significativo dalla sua espressione ideale, considerando *sia il modulo che la fase*.
- Una trattazione approfondita degli effetti del “disaccoppiamento approssimato” sulla successiva sintesi dei due regolatori SISO esula da questo corso.



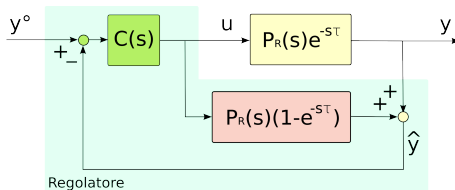
Controllo multivariabile con disaccoppiamento

- Schema completo del sistema di controllo



- I regolatori $R_1(s)$ e $R_2(s)$ si progettano coi metodi noti, come se si avesse a che fare con due sistemi da controllare indipendenti, aventi funzione di trasferimento rispettivamente $P_{11}(s)$ e $P_{22}(s)$.
- Anche sul disaccoppiamento vedremo più avanti degli esempi.

Predittore di Smith (Smith predictor)



- Serve quando il processo contiene un ritardo così grande che per avere un margine di fase accettabile bisogna limitare la banda di controllo in un modo che invece accettabile non è.
- Sfrutta il fatto che, nello schema mostrato, risulta

$$\frac{\hat{Y}(s)}{U(s)} = P_R(s)$$

dove la funzione di trasferimento $P_R(s)$ si assume razionale fratta.

- Il blocco $C(s)$ si può quindi sintetizzare, coi metodi noti, tenendo conto della sola “dinamica razionale” del processo.



- Dallo schema, infatti, risulta subito

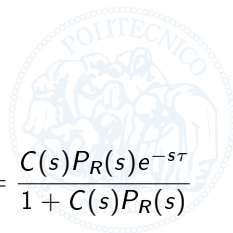
$$\frac{\hat{Y}(s)}{Y^o(s)} = \frac{C(s)P_R(s)}{1 + C(s)P_R(s)}$$

- Inoltre, il “regolatore” indicato nello schema stesso è evidentemente equivalente a uno in retroazione con funzione di trasferimento

$$R(s) = \frac{C(s)}{1 + C(s)P_R(s)(1 - e^{-s\tau})}$$

per cui

$$\begin{aligned} \frac{Y(s)}{Y^o(s)} &= \frac{R(s)P_R(s)e^{-s\tau}}{1 + R(s)P_R(s)e^{-s\tau}} \\ &= \frac{\frac{C(s)}{1 + C(s)P_R(s)(1 - e^{-s\tau})}P_R(s)e^{-s\tau}}{1 + \frac{C(s)}{1 + C(s)P_R(s)(1 - e^{-s\tau})}P_R(s)e^{-s\tau}} = \dots = \frac{C(s)P_R(s)e^{-s\tau}}{1 + C(s)P_R(s)} \end{aligned}$$



- In conclusione, allora,

$$\frac{Y(s)}{Y^o(s)} = \frac{\hat{Y}(s)}{Y^o(s)} e^{-s\tau}$$

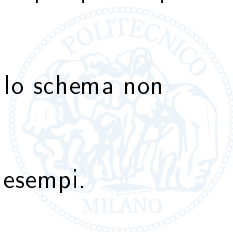
ovvero, sintetizzando $C(s)$ sulla base di $P_R(s)$ e usando lo schema a predittore di Smith, il comportamento di y che si ottiene è quello di \hat{y} , semplicemente ritardato di τ .

- Naturalmente occorre

- che il modello sia *preciso* (più di quanto serve per la sintesi di un “normale” regolatore in retroazione), specialmente per il ritardo (tanto più quanto questo domina la dinamica razionale)
- e che non vi siano disturbi (significativi),

perché se non è così \hat{y} non è una buona predizione di y e lo schema non funziona.

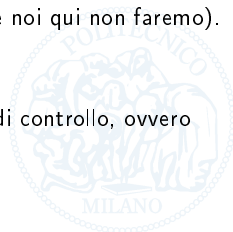
- Anche sul predittore di Smith vedremo a suo tempo degli esempi.



Breve introduzione a Scilab



- Scilab (<http://www.scilab.org>) è un pacchetto di calcolo orientato ad applicazioni scientifiche e ingegneristiche, sviluppato presso INRIA ed ENPC (Francia) e oggi curato dallo “Scilab Consortium”, che lo rende disponibile in forma open source.
- In estrema sintesi, Scilab “è simile a MATLAB” (e ha anche un package di nome Scicos che “è simile a Simulink”) ma
 - è gratuito
 - e soprattutto open, il che vuol dire che chi è interessato può studiarne il funzionamento attraverso i sorgenti (cosa che ovviamente noi qui non faremo).
- Ora
 - vedremo in breve i “fondamentali” di Scilab
 - e poi lo useremo per fare qualche simulazione di sistemi di controllo, ovvero l’uso che a noi interessa di più.



Comandi fondamentali (per noi): matrici, vettori, autovalori, definizione di funzioni di trasferimento

```

scilab-4.0  scilab-4.0 Scilab Consortium (Inria, Enpc)
File Control Demos Graphic Window 0 Help Editor

-->A=[1 2;3 4];
-->b=[1 0];
-->c=[0;1];
-->b*A*c
ans =
    2.
-->c'*inv(A)*b'
ans =
    1.5
-->[M,D]=spec(A)
D =
    5.3722813    0
    0    -0.3722813
M =
    0.4159736    -0.8245648
    0.9093767    0.5657675
-->inv(M)*A*M
ans =
    5.3722813    -8.528E-16
    -3.089E-16    -0.3722813
-->

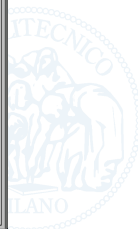
```

```

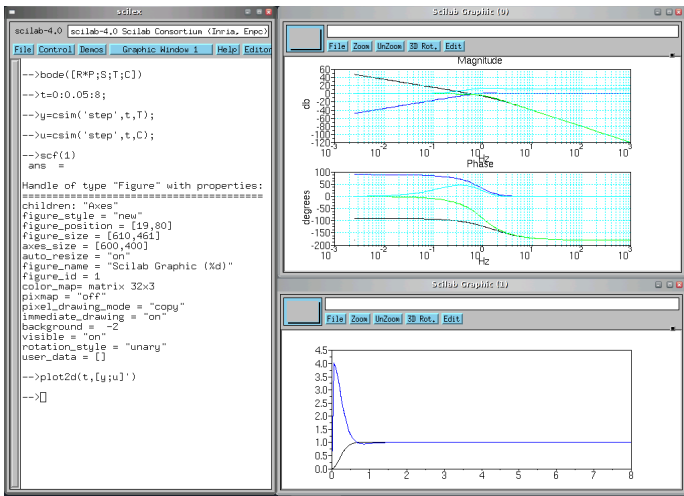
scilab-4.0  scilab-4.0 Scilab Consortium (Inria, Enpc)
File Control Demos Graphic Window 0 Help Editor

-->s=poly(0,"s");
-->P=syslin('c',1/((1+s)*(1+0.1*s)))
P =
    1
    -----
    1 + 1.1s + 0.1s2
-->R=syslin('c',4*(1+1/s))
R =
    4 + 4s
    -----
    s
-->T=R*P/(1+R*P)
T =
    40
    -----
    40 + 10s + s2
-->S=1/(1+R*P);
-->C=R/(1+R*P);
-->

```



Comandi fondamentali (per noi): diagrammi di Bode e risposte a scalino

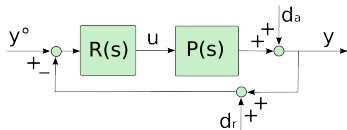


- Per quel che vi servisse in seguito, c'è un ottimo sistema di help.
- Qualcosa di più comunque vedremo adesso, facendo un semplice esempio relativi alla sintesi del controllore in retroazione e alla simulazione dello schema di controllo risultante.
- La prossima volta, dopo aver fatto esercizi sulle strutture di controllo, useremo Scilab per simularle e discuteremo brevemente i risultati ottenuti.
- Quindi, per la prossima volta ci si aspetta che possediate la manualità minima su Scilab (per installarlo e capire quel che vi serve a questo livello bastano due ore).



Esempio introduttivo a Scilab

Sintesi del controllore in retroazione



- Dato lo schema di controllo in figura, dove

$$P(s) = \frac{1 - 0.2s}{(1 + 5s)(1 + s)^2},$$

determinare $R(s)$ in modo che il sistema in anello chiuso sia asintoticamente stabile e che

- l'errore a regime per $y^o(t) = sca(t)$, $d_a(t) = d_r(t) = 0$ sia nullo,
- la pulsazione critica ω_c sia almeno pari a 0.5 r/s,
- il margine di fase φ_m sia di almeno 35° ,
- un disturbo $d_a(t) = D_a \sin(\omega_a t)$, $|D_a| \leq 10$, $\omega_a \leq 0.02\pi$, produca asintoticamente su $y(t)$ un effetto di ampiezza non maggiore di 1,
- un disturbo $d_r(t)$ sinusoidale con ampiezza minore o uguale a 1 e frequenza non minore di 3 Hz produca asintoticamente su $y(t)$ un effetto di ampiezza non maggiore di 0.01.

Esempio introduttivo a Scilab

Sintesi del controllore in retroazione

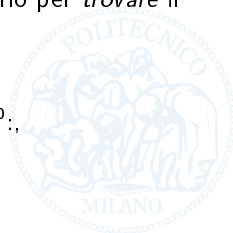
- Eseguendo la sintesi coi metodi noti un possibile risultato è

$$R(s) = \frac{(1 + 5s)(1 + s)}{s(1 + 0.1s)}$$

il che produce

$$L(s) = R(s)P(s) = \frac{1 - 0.2s}{s(1 + s)(1 + 0.1s)}$$

- Usiamo Scilab per verificare il risultato (come potrete usarlo per *trovare* il risultato per tentativi, peraltro). Occorre che
 - $\omega_c \geq 0.5$ r/s,
 - $\varphi_m \geq 35^\circ$,
 - $|L(j\omega)|_{db} \geq 20$ per $\omega \leq 0.02\pi$ r/s, cioè per $f \leq 0.01$ Hz¹⁰.,
 - $|L(j\omega)|_{db} \leq -40$ per $f \geq 3$ Hz.



¹⁰ATTENZIONE: Scilab usa nativamente gli Hz anziché i r/s!

Esempio introduttivo a Scilab

Sintesi del controllore in retroazione

- Anzitutto definiamo P , R e L :

```
s = poly(0,'s');  
P = syslin('c',(1-0.2*s)/((1+5*s)*(1+s)^2));  
R = syslin('c',(1+5*s)*(1+s)/(s*(1+0.1*s)));  
L = R*P;
```

- Quindi calcoliamo ω_c e φ_m :

```
[pc,fc] = p_margin(L);  
wc      = 2*%pi*fc;  
pm      = pc+180;
```

Otteniamo $\omega_c = 0.79$ r/s e $\varphi_m = 38.12^\circ$, che va bene.

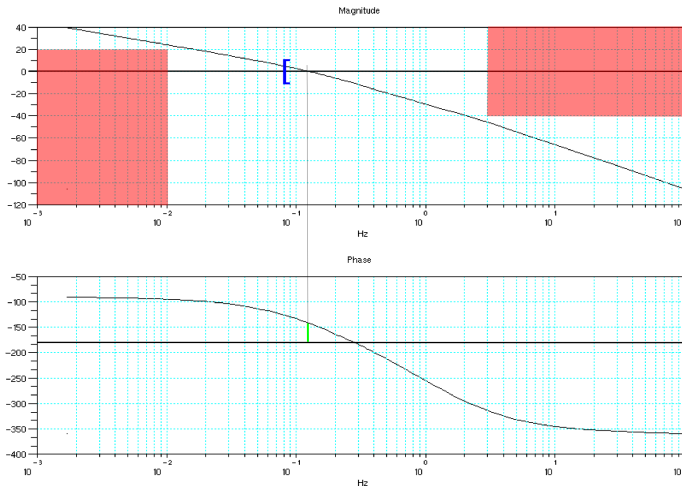
ATTENZIONE: il comando `p_margin` restituisce la *frequenza* critica in Hz e soprattutto, nonostante il nome, la *fase critica* e non il margine di fase!



Esempio introduttivo a Scilab

Sintesi del controllore in retroazione

- Verifichiamo ora il resto della sintesi sui diagrammi di Bode di $L(j\omega)$:



Esempio introduttivo a Scilab

Sintesi del controllore in retroazione

- Calcoliamo le FdT d'interesse per simulare il comportamento del sistema:

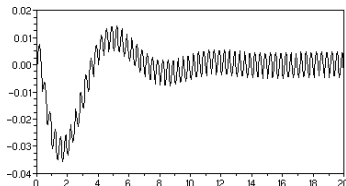
$$Y_o2Y = L/(1+L);$$

$$D_a2Y = 1/(1+L);$$

$$D_r2Y = -L/(1+L);$$

- Definiamo opportuni vettori per tempo e ingressi (guardiamo soltanto la risposta di y a d_r - ovviamente nel caso limite - e lasciamo il resto come esercizio agli interessati), simuliamo e plottiamo:

```
t = 0:0.05:10;  
dr = sin(2*pi*3*t);  
ydr = csim(dr,t,Dr2Y);  
plot2d(t,ydr);
```



Messa a punto di schemi con strutture di controllo

Simulazione in Scilab di strutture di controllo

Questa esercitazione è svolta usando la lavagna e la carta semilogaritmica (per la prima parte) e in modo prevalentemente interattivo con Scilab (per la seconda parte); per la prima parte non ci sono slide guida.



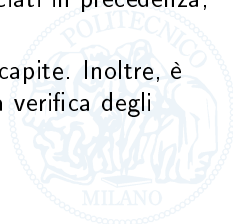
Simulazione in Scilab di strutture di controllo



Simulazione in Scilab di strutture di controllo

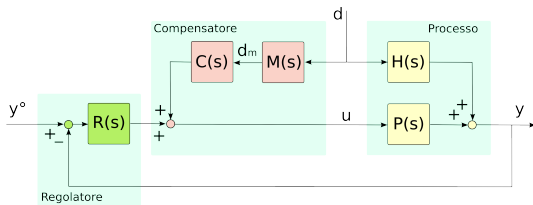
premessa

- Conoscete i “fondamentali” di Scilab.
- Ora faremo alcuni esempi di simulazione relativi alle strutture di controllo.
- In dettaglio, vedremo
 - la compensazione in anello aperto,
 - il controllo in cascata,
 - il controllo 2×2 con disaccoppiamento,
 - il controllo con predittore di Smith.
- Lo scopo è confermare la comprensione dei concetti enunciati in precedenza, non imparare comandi che sono documentati sui manuali.
- Cercate di seguire con attenzione e fate domande se non capite. Inoltre, è bene che facciate in seguito *vostre* simulazioni in Scilab, a verifica degli esercizi che farete da soli studiando.



Simulazione in Scilab di strutture di controllo

Compensazione in anello aperto



- Dati

$$P(s) = \frac{1}{(1 + 10s)^2}, \quad H(s) = \frac{1}{1 + 0.1s}, \quad R(s) = \frac{1 + 10s}{10s}, \quad M(s) = \frac{1}{1 + 0.1s}$$

determinare il blocco $C(s)$, se necessario, in modo da compensare asintoticamente (idealmente se possibile e se no indicando e giustificando le approssimazioni introdotte) l'effetto su $y(t)$ di un disturbo $d(t)$ sinusoidale con frequenza non maggiore di 1 Hz.

Simulazione in Scilab di strutture di controllo

Compensazione in anello aperto

- Risulta

$$L(s) = R(s)P(s) = \frac{1}{10s} \Rightarrow \omega_c = 0.1r/s = \frac{0.1}{2\pi} \text{ Hz} \cong 0.016 \text{ Hz}$$

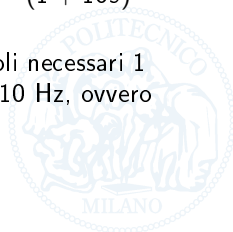
quindi la frequenza del disturbo può trovarsi vicino o addirittura oltre il limite superiore della banda di controllo, rendendo la retroazione inefficace a contrastarlo. La compensazione è pertanto necessaria.

- Calcolando il compensatore si ha

$$C_{id}(s) = -\frac{H(s)}{M(s)P(s)} = -\frac{(1 + 0.1s)(1 + 10s)^2}{(1 + 0.1s)} = -(1 + 10s)^2$$

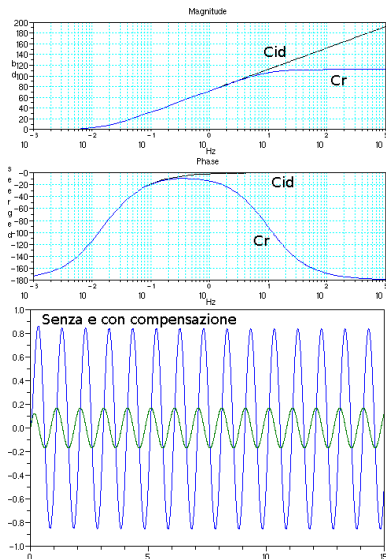
evidentemente non realizzabile. Introduciamo allora i 2 poli necessari 1 decade oltre la massima frequenza del disturbo, ovvero a 10 Hz, ovvero (attenzione!) a 20π r/s:

$$C_r(s) = -\frac{(1 + 10s)^2}{(1 + \frac{s}{20\pi})^2}$$



Simulazione in Scilab di strutture di controllo

Compensazione in anello aperto



Diagrammi di Bode dei compensatori ideale e reale (sopra) e risposta di y a $d(t) = \sin(t)$ (sotto):

```
s = poly(0,'s');
P = syslin('c',1/(1+10*s)^2);
H = syslin('c',1/(1+0.1*s));
R = syslin('c',(1+10*s)/(10*s));
M = syslin('c',1/(1+0.1*s));
fdHz = 1;

Cid = -H/(M*P);
Cp = syslin('c',1/(1+s/(2*pi*10*fdHz))^2);
Cr = Cid*Cp;

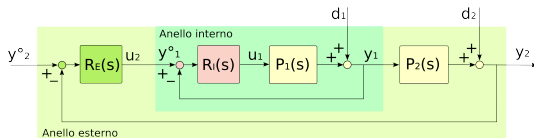
DYnc = H/(1+R*P);
DYc = (H+M*Cr*P)/(1+R*P);
t = 0:0.005:15;
d = sin(2*pi*fdHz*t);
ync = csim(d,t,DYnc);
yc = csim(d,t,DYc);

scf(0); bode([Cid;Cr]);
scf(1); plot(t,[ync;yc]);
```



Simulazione in Scilab di strutture di controllo

Controllo in cascata



- Dati

$$P_1(s) = \frac{2}{1+s}, \quad P_2(s) = \frac{0.8}{(1+5s)(1+20s)}$$

mettere a punto uno schema in cascata che garantisca per l'anello esterno una costante di tempo non superiore a 2.5 s e una separazione in banda tra l'anello esterno e l'interno di una decade.

- Quindi, confrontare il sistema di controllo in cascata con uno ad anello singolo, sintetizzato in modo da garantire le stesse prestazioni relativamente a y_2 .

Simulazione in Scilab di strutture di controllo

Controllo in cascata

- L'anello esterno dovrà avere una banda $\omega_{cE} = 1/2.5 = 0.4$ r/s. Quindi, l'anello interno dovrà avere $\omega_{cI} = 10\omega_{cE} = 4$ r/s.
- Il regolatore $R_I(s)$ dell'anello interno si sintetizza basandosi su $P_1(s)$; una possibile soluzione è

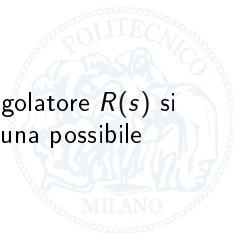
$$R_I(s) = 2 \frac{1+s}{s}$$

- Similmente, assumendo ideale l'anello interno (ossia a patto che $\omega_{cE} \ll 10\omega_{cI}$), il regolatore $R_E(s)$ dell'anello esterno si sintetizza basandosi sul solo $P_2(s)$; una possibile soluzione è

$$R_E(s) = 0.5 \frac{(1+5s)(1+20s)}{s(1+0.1s)}$$

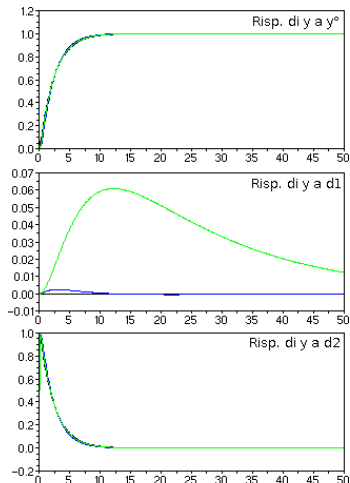
- Quanto al sistema di controllo in anello singolo, dove il regolatore $R(s)$ si sintetizza basandosi su $P_1(s)P_2(s)$, volendo $\omega_c = 0.4$ r/s una possibile soluzione è

$$R(s) = 0.5s \frac{(1+s)(1+5s)(1+20s)}{s(1+0.1s)^2}$$



Simulazione in Scilab di strutture di controllo

Controllo in cascata



Cascata con a.i. ideale (nero), cascata con a.i. reale (blu), controllo ad anello singolo (verde)

```
s = poly(0,'s');
P1 = syslin('c',2/(1+s));
P2 = syslin('c',0.8/((1+5*s)*(1+20*s)));
Ri = syslin('c',2*(1+s)/s);
Re = syslin('c',0.5*(1+5*s)*(1+20*s)/(s*(1+0.1*s)));
R = syslin('c',0.25*((1+s)*(1+5*s)*(1+20*s)/(s*(1+0.1*s)^2)));
t = 0:0.1:50;
```

```
Yo2Yci = Re*P2/(1+Re*P2);
D22yci = 1/(1+Re*P2);
Yo2Ycr = Re*Ri*P1*P2/(1+Ri*P1*(1+Re*P2));
D12Ycr = P2/(1+Ri*P1*(1+Re*P2));
D22Ycr = (1+Ri*P1)/(1+Ri*P1*(1+Re*P2));
Yo2Ync = R*P1*P2/(1+R*P1*P2);
D12Ync = P2/(1+R*P1*P2);
D22ync = 1/(1+R*P1*P2);
```

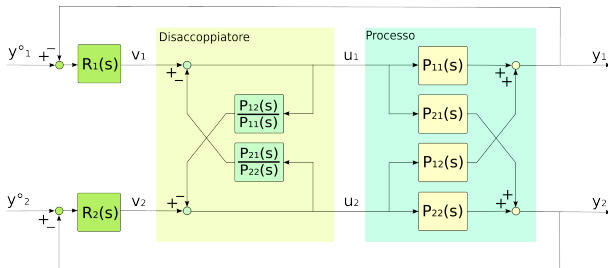
```
yYoci = csim('step',t,Yo2Yci);
yD1ci = zeros(t);
yD2ci = csim('step',t,D22yci);
yYocr = csim('step',t,Yo2Ycr);
yD1cr = csim('step',t,D12Ycr);
yD2cr = csim('step',t,D22Ycr);
yYonc = csim('step',t,Yo2Ync);
yD1nc = csim('step',t,D12Ync);
yD2nc = csim('step',t,D22ync);
```

```
scf(0);
subplot(311);plot2d(t,[yYoci;yYocr;yYonc]');
subplot(312);plot2d(t,[yD1ci;yD1cr;yD1nc]');
subplot(313);plot2d(t,[yD2ci;yD2cr;yD2nc]');
```



Simulazione in Scilab di strutture di controllo

Controllo 2×2 con disaccoppiamento



- Dato il processo 2×2 con matrice di trasferimento

$$P(s) = \begin{bmatrix} \frac{1}{1+5s} & \frac{0.5}{1+4s} \\ \frac{0.5}{1+3s} & \frac{1}{1+2s} \end{bmatrix}$$

mettere a punto per esso un sistema di controllo con disaccoppiamento all'indietro e due regolatori SISO tale da garantire, per ambedue gli anelli, una costante di tempo in anello chiuso di 1 s.



Simulazione in Scilab di strutture di controllo

Controllo 2×2 con disaccoppiamento

- Risulta

$$\begin{aligned} D(s) &= \begin{bmatrix} 1 & \frac{P_{12}(s)}{P_{11}(s)} \\ \frac{P_{21}(s)}{P_{22}(s)} & 1 \end{bmatrix}^{-1} \\ &= \begin{bmatrix} 1 & \frac{0.5(1+5s)}{1+4s} \\ \frac{0.5(1+2s)}{1+3s} & 1 \end{bmatrix}^{-1} \end{aligned}$$

dove ambedue i blocchi

$$\frac{P_{12}(s)}{P_{11}(s)} \quad \text{e} \quad \frac{P_{21}(s)}{P_{22}(s)}$$

sono fisicamente realizzabili.



Simulazione in Scilab di strutture di controllo

Controllo 2×2 con disaccoppiamento

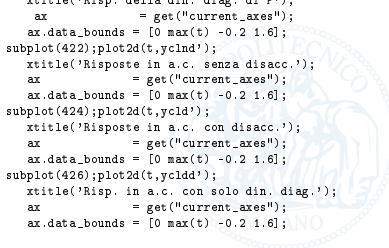
Codice Scilab per la simulazione

```
s = poly(0,'s');
P11 = syslin('c',1.0/(1+5*s)); // FdT del processo
P12 = syslin('c',0.5/(1+4*s));
P21 = syslin('c',0.5/(1+3*s));
P22 = syslin('c',1.0/(1+2*s));
P = [P11,P12;P21,P22]; // MdT del processo
iD12 = P12/P11; // Elementi di D-1
iD21 = P21/P22;
D = inv([1,iD12;iD21,1]); // Disaccoppiatore
R1 = syslin('c',1/P11*1/s); // Regolatori
R2 = syslin('c',1/P22*1/s);
R = [R1,0;0,R2]; // MdT (diag.) del regolatore
Pdd = [P11,0;0,P22];
T = inv(eye(2,2)+P*D*R)*P*D*R; // Anello chiuso con decoupling,
Tdd = inv(eye(2,2)+Pdd*R)*Pdd*R; // con la sola din. diag. di P
Tnd = inv(eye(2,2)+P*R)*P*R; // e senza decoupling

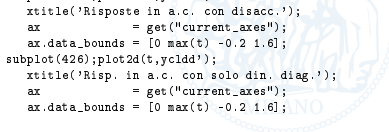
N2 = 500;
Ts = 0.05;
t = (0:2*N2-1)*Ts;
u = [ones(1,2*N2);zeros(1,N2),ones(1,N2)];

yolnd = csim(u,t,P);
yold = csim(u,t,P*D);
yoldd = csim(u,t,Pdd);
yclnd = csim(u,t,T);
ycldd = csim(u,t,Tdd);
yclnd = csim(u,t,Tnd);
```

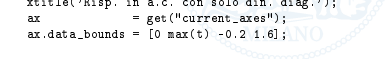
```
scf(0); clf;
subplot(421);plot2d(t,u');
    xtitle('Ingressi');
    ax = get("current_axes");
    ax.data_bounds = [0 max(t) -0.2 1.2];
subplot(423);plot2d(t,yolnd');
    xtitle('Risposte di P');
    ax = get("current_axes");
    ax.data_bounds = [0 max(t) -0.2 1.6];
subplot(425);plot2d(t,yold');
    xtitle('Risposte di PD');
    ax = get("current_axes");
    ax.data_bounds = [0 max(t) -0.2 1.6];
subplot(427);plot2d(t,yoldd');
    xtitle('Risp. della din. diag. di P');
    ax = get("current_axes");
    ax.data_bounds = [0 max(t) -0.2 1.6];
subplot(422);plot2d(t,yclnd');
    xtitle('Risposte in a.c. senza disacc.');
```



```
    ax = get("current_axes");
    ax.data_bounds = [0 max(t) -0.2 1.6];
subplot(424);plot2d(t,ycldd');
    xtitle('Risposte in a.c. con disacc.');
```



```
    ax = get("current_axes");
    ax.data_bounds = [0 max(t) -0.2 1.6];
subplot(426);plot2d(t,yclnd');
    xtitle('Risp. in a.c. con solo din. diag.');
```

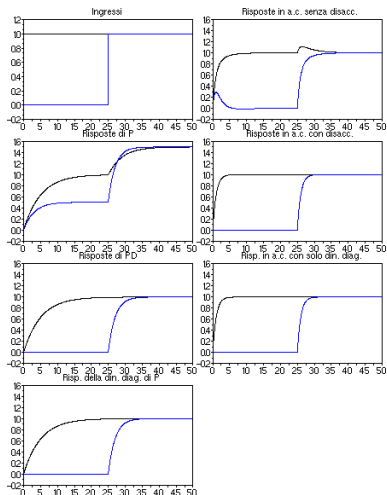


```
    ax = get("current_axes");
    ax.data_bounds = [0 max(t) -0.2 1.6];
```

Simulazione in Scilab di strutture di controllo

Controllo 2×2 con disaccoppiamento

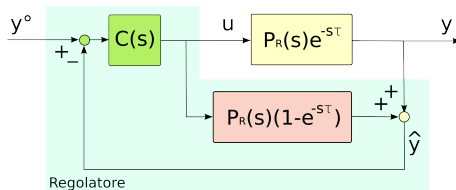
Risultati della simulazione



- Si noti l'efficacia del disaccoppiamento: nella colonna di sinistra si confrontano le risposte del processo con e senza disaccoppiatore e quelle della sola dinamica diagonale.
- A destra sono invece riportati i risultati del controllo in anello chiuso, anche qui nei tre casi col processo con e senza disaccoppiatore e con la sola dinamica diagonale (ovviamente senza disaccoppiatore).

Simulazione in Scilab di strutture di controllo

Controllo con predittore di Smith



- Dato

$$P(s) = P_R(s)e^{-sT} = \frac{e^{-8s}}{(1+s)(1+0.2s)}$$

mettere a punto uno schema di controllo con predittore di Smith, se necessario, in modo che l'errore a transitorio esaurito per una variazione a scalino di y^o sia nullo e che il tempo di assestamento della risposta di y a uno scalino di y^o non superi i 10 s.

Simulazione in Scilab di strutture di controllo

Controllo con predittore di Smith

- Anzitutto vediamo se il predittore di Smith serve: un tempo di assestamento di 10 s corrisponde a una costante di tempo dominante in anello chiuso di circa $10/5 = 2$ s, ovvero a $\omega_c = 1/2$ r/s. La fase di $P(j\omega)$ alla frequenza critica necessaria è allora

$$\arg^{\circ}(P(j0.5)) = -\tan^{-1}(0.5) - \tan^{-1}(0.2 \cdot 0.5) - 8 \cdot 0.5 \frac{180^{\circ}}{\pi} \cong -270^{\circ}$$

- Questo richiederebbe, se si volesser usare uno schema in retroazione senza predittore, che il regolatore di tale schema fornisse, al limite superiore della banda di controllo, un anticipo ben maggiore di 90° . Ciò è al di là di ogni ragionevolezza¹¹, quindi il predittore serve.
- Procediamo quindi a determinare il blocco $C(s)$, ovvero la sola incognita dello schema con predittore di Smith.

¹¹Spunto di riflessione (e ripasso): perché?

Simulazione in Scilab di strutture di controllo

Controllo con predittore di Smith

- Il tempo di assestamento di \hat{y} , che precede y di un tempo pari al ritardo, è da rendere uguale al tempo di assestamento richiesto per y meno - appunto - il ritardo.
- Qui vogliamo che y si assesti in 10 s, quindi \hat{y} deve farlo in $10 - 8 = 2$ s al massimo.
- Allora $C(s)$ deve dare, relativamente al sistema in anello chiuso composto da lui e dal solo $P_R(s)$, errore nullo a transitorio esaurito per una variazione a scalino del riferimento e $\omega_c \geq 5/2$.
- Scegliendo per semplicità proprio $\omega_c = 5/2$ e usando i metodi noti, una possibile soluzione è

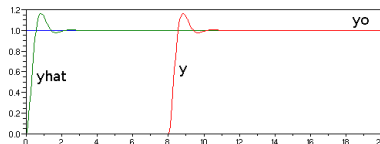
$$C(s) = \frac{1+s}{0.25s}$$



Simulazione in Scilab di strutture di controllo

Controllo con predittore di Smith

- Come si vede dalle risposte a uno scalino di y^o , le specifiche sono rispettate.



- Il codice Scilab usato è riportato in appendice a questa lezione, a uso di chi è interessato.
- La questione è che gli operatori aritmetici overloaded in Scilab per trattare i sistemi dinamici operano soltanto su funzioni di trasferimento razionali¹². Quindi per trattare blocchi con ritardo un modo (accettando di vincolarsi a un'integrazione a passo fisso e prefissato sia per il processo che per il regolatore) è operare a tempo discreto, dove anche il ritardo ha una rappresentazione razionale.

¹²La stessa cosa è vera in MATLAB, per la cronaca.

Simulazione in Scilab di strutture di controllo

APPENDICE: codice Scilab per la simulazione dello schema con predittore di Smith

```
s = poly(0,'s');
z = poly(0,'z');
Ts = 0.05; // Passo di campionamento
Prc = syslin('c',1/((1+s)*(1+0.2*s))); // Parte razionale di P(s)
Ptau = 8; // Ritardo di P(s)
Cc = syslin('c',(1+s)/(0.25*s)); // Blocco C(s)
t = 0:Ts:20; // Vettore dei tempi
yo = ones(t); // Scalino di yo

// Discretizzazione delle FdT Prd(s) e Cc(s): per farla in Scilab
// si trasforma da 'State Space' (A,b,c,d) a 'Transfer Function'
// | il risultato della discretizzazione a passo Ts
// | | della rappresentazione State Space della FdT di partenza
// | | |
// ---+-- --+-- ---+--
Prd = ss2tf(dscr(tf2ss(Prc),Ts));
Cd = ss2tf(dscr(tf2ss(Cc),Ts));
Pdd = syslin('d',1/z^floor(Ptau/Ts)); // Rappresentazione a tempo discreto del ritardo
Pd = Prd*Pdd; // e quindi di Prc(s)*e^(-s*Ptau)
Pred = Prd*(1-Pdd); // e di Prc(s)*(1-e^(-s*Ptau))

Trd = Cd*Prd/(1+Cd*Prd); // FdT da yo a y^ (non compare il ritardo)
Rd = Cd/(1+Cd*Pred); // FdT del regolatore con predittore
Td = Rd*Pd/(1+Rd*Pd); // FdT da yo a y, calcolata sullo schema con predittore
yhat = dsimul(tf2ss(Trd),yo); // Simulazione di Trd per avere y^
y = dsimul(tf2ss(Td),yo); // Simulazione di Td per avere y^

scf(0); plot(t,[yo;yhat,y]);
```



Cenni al condizionamento dei segnali

Architettura hardware e software dei sistemi di controllo

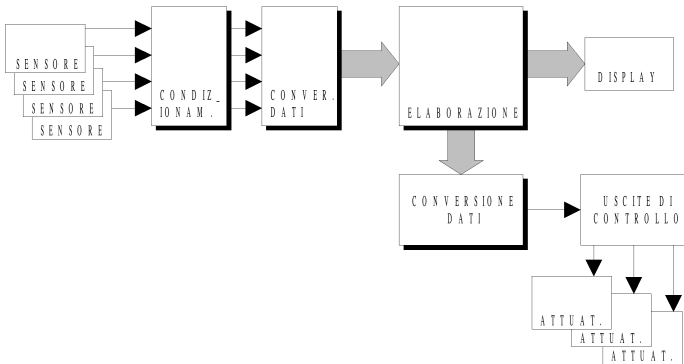


Cenni al condizionamento dei segnali nei sistemi di controllo



Catena di misura, controllo e attuazione

Schema sintetico

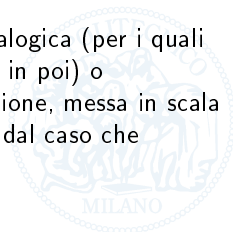


Vedremo ora le principali *funzioni* di condizionamento: sui componenti a esse dedicati daremo qualche cenno nella sezione 'architettura hw/sw', mentre il funzionamento dettagliato dei circuiti in essi contenuti lo conoscete dai corsi di Elettronica.

Catena di misura, controllo e attuazione

Principali funzioni di condizionamento nella catena di misura

- Le principali funzioni di condizionamento nella catena di misura sono
 - trasduzione,
 - amplificazione di segnale e filtraggio,
 - messa in scala.
 - trasmissione analogica,
 - conversione analogico/digitale (A/D),
 - trasmissione digitale.
- A esse 'segue' (nel flusso di segnale) il calcolo del controllo, che assumiamo fatto da uno o più elaboratori digitali.
- Non trattiamo qui controllori realizzati con elettronica analogica (per i quali peraltro basta omettere le funzioni dalla conversione A/D in poi) o pneumatica, per i quali si conservano i concetti di trasduzione, messa in scala e trasmissione ma la loro realizzazione è del tutto diversa dal caso che interessa a noi in questo corso.



Catena di misura, controllo e attuazione

Principali funzioni di condizionamento nella catena di attuazione

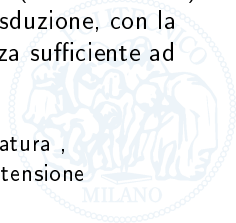
- Le principali funzioni di condizionamento nella catena di attuazione sono
 - trasmissione digitale,
 - conversione digitale/analogica (D/A),
 - trasmissione analogica,
 - amplificazione di potenza,
 - attuazione.
- Anche qui, ci limitiamo al caso in cui il calcolo del controllo è fatto da uno o più elaboratori digitali.



Definizione delle operazioni di condizionamento

Trasduzione e attuazione

- Per 'trasduzione' s'intende la trasformazione di una grandezza fisica in una diversa che le corrisponde secondo una legge nota.
- Esempi:
 - una termocoppia trasduce temperatura in tensione,
 - un sensore di posizione a potenziometro alimentato con tensione costante ai capi fissi trasduce posizione in tensione al cursore (oppure, se vogliamo essere pignoli, trasduce posizione in resistenza e il circuito che gli sta attorno trasduce questa in tensione),
 - ...
- Il segnale trasdotto (per noi, sempre elettrico) è in generale di piccola potenza e rumoroso, quindi va amplificato, filtrato e messo in scala (come vedremo).
- Per 'attuazione' s'intende l'operazione simmetrica alla trasduzione, con la differenza che il segnale ricevuto dall'attuatore è di potenza sufficiente ad azionarlo.
- Esempi:
 - un motore elettrico attua coppia ricevendo corrente d'armatura ,
 - un riscaldatore elettrico attua potenza termica ricevendo tensione d'alimentazione,
 - ...



Trattamento del segnale dai trasduttori

Amplificazione, filtraggio e messa in scala

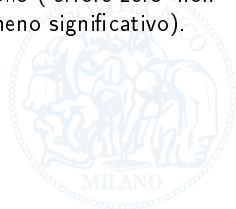
- Queste tre operazioni avvengono di solito insieme, ossia nello stesso circuito elettronico.
- Il segnale (debole) proveniente dal trasduttore viene amplificato e messo in scala, ossia fatto variare in un range standard per la trasmissione analogica.
- i tipici range sono 0–10 V, 0–5 V, 1–5 V, -10–10 V e -5–5 V per la trasmissione in tensione, 4–20 mA per la trasmissione in corrente.
- Ogni insieme trasduttore-amplificatore (a volte detto sensore o trasmettitore) ha quindi una scala (ad esempio, 4–20 mA per 0–100 °C). Occorre sceglierne uno adeguato al range che le variabili d'interesse hanno nel sistema da controllare. riconoscibili, il che ha influenza sulla precisione del controllo ('errore zero' non può significare nulla di meglio che 'errore minore del bit meno significativo').
- Nel processo di amplificazione è tipicamente incluso un filtraggio (analogico, ovviamente) di tipo passa-basso, per ripulire il segnale dal rumore di misura sempre presente. In generale la dinamica del filtro passa-basso può essere trascurata nel progetto del controllo, ma se la relativa frequenza di taglio non è molto maggiore della banda dell'anello interessato può essere necessario tenerne conto.

Trasmissione analogica

Trasmissione dai sensori e verso gli attuatori

- La trasmissione analogica avviene in tensione o in corrente. I segnali trasmessi sono già stati messi in scala.
- Nella trasmissione in tensione i ricevitori sono in parallelo.
- Se sono troppi il parallelo delle loro impedenze, che si pone in serie all'impedenza interna del trasmettitore e a quella della linea di trasmissione, può attenuare (quindi alterare) il segnale. Occorre fare attenzione a tale possibile effetto (vedremo qualche esercizio numerico a suo tempo).
- nella trasmissione in corrente i ricevitori sono in serie.
- La corrente è certamente la stessa in tutti, a patto però che il trasmettitore sia in grado d'imprimerla alla serie di tutte le impedenze dei ricevitori e della linea (anche su questo faremo a suo tempo qualche esercizio numerico).
- La trasmissione 4–20 mA (o l'equivalente in tensione 1–5 V) si dicono 'a zero vivo' perché hanno il vantaggio che se si legge zero si sa che la linea di trasmissione non funziona (vedremo anche qualche semplice esercizio su cosa si legge in vari punti di una linea se essa s'interrompe o si cortocircuita).
- NOTA: gli esercizi di cui sopra sono banali applicazioni di quel che sapete dal corso di Elettrotecnica (quindi chi non maneggia bene partitori *et similia* provveda).

- Come funziona un convertitore A/D lo sapete già.
- Ai fini del condizionamento dei segnali a noi interessano
 - il tempo di conversione (è un ritardo, di solito non conta nulla ma sapere che c'è è cosa buona),
 - la scala, ovvero i valori minimi e massimi della grandezza analogica (tensione o corrente) in ingresso,
 - la risoluzione, ovvero il numero N di bit, ovvero il numero 2^N di valori diversi riconoscibili, il che ha influenza sulla precisione del controllo ('errore zero' non può significare nulla di meglio che errore minore del bit meno significativo).



- Anche come funziona un convertitore D/A lo sapete già.
- Ai fini del condizionamento dei segnali a noi interessano
 - il tempo di conversione (vedi sopra),
 - la scala, ovvero i valori minimi e massimi della grandezza analogica (tensione o corrente) in uscita,
 - la risoluzione, ovvero il numero N di bit, ovvero il numero 2^N di valori diversi producibili (altro effetto di quantizzazione che influisce sulla precisione del controllo).
- Sugli effetti della quantizzazione si potrebbe discutere molto a lungo. Noi daremo soltanto qualche cenno nella sezione relativa all'architettura dei sistemi di controllo.



Amplificazione di potenza

- Può avvenire in molti modi, ma tipicamente consiste nell'usare il segnale di comando all'attuatore per modulare la potenza proveniente da una sorgente esterna, sia in modo continuo che ad esempio alimentandola con un'onda quadra di periodo piccolo rispetto alle dinamiche del sistema da controllare e il cui duty cycle va da 0 a 100 quando il comando va dal minimo al massimo (comando 'a divisione di tempo' o PWM, Pulse Width Modulation).
- Anche gli attuatori hanno quindi una scala (ad esempio, 0–1 kW per 0–10 V) e vanno dimensionati bene e insieme ai sensori, facendo in modo che per i range delle variabili d'interesse si usi tutta la scala, a vantaggio della precisione nelle conversioni.
- Ai fini di questo corso non ci servono altri dettagli.
- Sul condizionamento dei segnali c'è un capitolo nella dispensa, dove si trovano tutti i dettagli omessi qui (e le risposte alle principali curiosità che potreste avere).



Architettura hardware e software dei sistemi di controllo



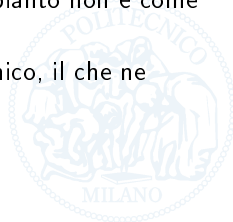
Architettura hardware e software dei sistemi di controllo

Premessa

- I concetti di 'architettura' di un elaboratore e di una rete vi sono noti.
- Anche nei sistemi di controllo tali concetti hanno luogo, dovendo però essere adeguati e a volte completati per via delle particolarità che un sistema di controllo ha rispetto a un elaboratore o una rete qualsiasi.
- Molto in sintesi si possono raggruppare tali particolarità in tre categorie:
 - esigenze specifiche di componenti hardware e programmi per il controllo,
 - necessità di componenti *ad hoc*,
 - necessità d'integrazione con sistemi informativi più ampi.
- Daremo ora alcuni cenni su tali questioni. Non possiamo ovviamente esaurirle, quindi ci poniamo due soli obiettivi:
 - che usiate concetti 'informatici' già appresi sapendo applicarli a un campo molto particolare quale i sistemi di controllo
 - e che conosciate (almeno per nome e per il panorama odierno) i principali standard tecnologici e normativi coinvolti.
- Ricordate che questa materia si evolve di continuo. Non avrebbe senso qui spiegarvi l'esistente in dettaglio: dovete diventare capaci di capirlo voi, oggi e tutte le volte che in futuro cambierà.

Esigenze specifiche del controllo

- Determinismo temporale (particolarmente critico),
- tolleranza ai guasti e ai malfunzionamenti (sia del software che dell'hardware),
- grandi quantità di dati e segnali da gestire con diverse scale temporali, d'importanza e di criticità,
- interfaccia operatore (che coinvolge anche l'hardware) adatta a personale con grado di cultura ed addestramento molto variabile,
- semplicità e comodità nella costruzione e nella (ri)configurazione del sistema (mettere le mani su un sistema di controllo dentro un impianto non è come farlo su una rete di PC in un ufficio),
- possibilità di costruire e gestire il sistema in modo gerarchico, il che ne migliora l'organizzazione e quindi la manutenibilità.



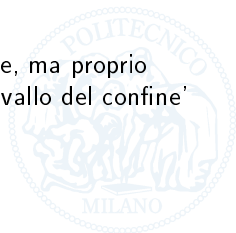
Necessità di componenti *ad hoc*

- Occorre interfacciarsi con il 'mondo fisico', misurando ed attuando grandezze eterogenee.
- I segnali da gestire sono di tipo diverso, provengono da fonti diverse e vanno trattati in modo diverso, essendo poi naturalmente trasmessi in modo standard.
- Gli ambienti in cui si opera sono molto diversi, e quindi i componenti si trovano nelle condizioni operative più svariate.
- Tuttavia, i componenti devono essere 'interoperabili', ovvero devono poter coesistere e cooperare in un sistema le cui parti provengono da produttori diversi.



Necessità d'integrazione con sistemi più ampi

- I sistemi di controllo sono sempre più spesso visti come una parte del sistema informatico (hw/sw) della fabbrica o dell'impianto.
- Quindi, il sistema di controllo in senso stretto (ovvero quel che fa seguire a certe variabili certi andamenti e/o fa sì che certi sistemi abbiano certi comportamenti) deve sempre più spesso integrarsi con
 - il monitoraggio dell'impianto (ai fini più svariati),
 - la gestione della produzione,
 - la gestione degli approvvigionamenti,
 - la logistica,
 - ...
- Si noti che, sempre più spesso, non si tratta d'interfacciare, ma proprio d'integrare i sistemi: vi sono funzionalità che stanno 'a cavallo del confine' del sistema di controllo vero e proprio.

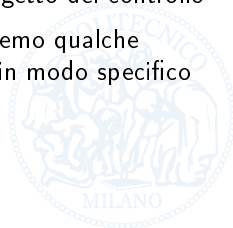


Architettura hardware e software dei sistemi di controllo

Gli argomenti che ora toccheremo

- Principali componenti hardware dei sistemi di controllo,
- sistemi di controllo distribuiti (DCS, Distributed Control Systems),
- bus di campo (Field Bus),
- aspetti normativi (standard IEC),
- strumenti per la configurazione e la programmazione dei sistemi di controllo,
- influenza dell'architettura di un sistema distribuito sul progetto del controllo

Ricordiamo che questa è soltanto una breve rassegna: indicheremo qualche riferimento per approfondimenti futuri, utile a chi si occuperà in modo specifico della materia.



Principali componenti hardware dei sistemi di controllo

Regolatori industriali (single- e multi-loop)

- Costituiscono la maggioranza dei regolatori impiegati nei sistemi di controllo industriali (circa il 64% secondo uno studio condotto nel 1998).
- Ve ne sono di
 - 'general purpose', tipicamente con legge di controllo PID e segnali di I/O standard (es. 0-10V, 4-20mA,...);
 - 'specializzati' per quanto riguarda
 - il numero di loop gestiti,
 - la legge di controllo (raramente con struttura molto variabile),
 - il tipo d'ingressi (termocoppia, encoder,...),
 - il numero di uscite (es. organo riscaldante e refrigerante), tipicamente gestite in split-range,
 - il tipo di uscite (PWM, a divisione di tempo,...).



Principali componenti hardware dei sistemi di controllo

Regolatori industriali (single- e multi-loop)

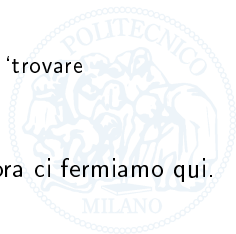
- Funzionalità principali:

- calcolo della legge di controllo;
- funzionamento in automatico, manuale o tracking;
- HMI (Human-Machine Interface) 'minima', spesso soltanto pochi tasti e un display a LCD o LED;
- allarmi.

- Funzionalità accessorie:

- HMI evoluta (grafici,...);
- comunicazione (seriale, industrial ethernet,...);
- programmazione di set point e parametri;
- autotuning ('sintonia automatica', ovvero programmi per 'trovare automaticamente dei buoni parametri'),
- ...

Sui regolatori industriali c'è una sezione del corso, quindi per ora ci fermiamo qui.



Principali componenti hardware dei sistemi di controllo

Hardware di acquisizione, condizionamento, trasmissione ed attuazione

- La tipica catena di acquisizione consta di
 - una parte analogica, che si sostanzia fisicamente in una linea di segnale composta da trasduzione, amplificazione, filtraggio, messa in scala, trasmissione;
 - una conversione analogico-digitale;
 - una parte digitale, che avviene attraverso la rete di processo.
- Si tende sempre di più a ridurre la parte analogica, essenzialmente per ragioni di
 - immunità ai disturbi,
 - economia di cablaggio,
 - riconfigurabilità delle connessioni.



Principali componenti hardware dei sistemi di controllo

Hardware di acquisizione, condizionamento, trasmissione ed attuazione

- La tipica catena di attuazione consta di
 - una parte digitale, che inizia dal dispositivo che genera il segnale di comando in forma numerica e trasmette questo segnale, lungo la rete di processo, al punto di conversione;
 - una conversione digitale-analogico;
 - una parte analogica, che si sostanzia fisicamente in una linea di segnale composta da trasmissione, amplificazione di potenza, trasduzione.
- Anche qui si tende sempre di più a ridurre la parte analogica, per le stesse ragioni evidenziate parlando della catena di acquisizione.

Del condizionamento dei segnali abbiamo già detto quanto c'interessa qui nell'apposita sezione.

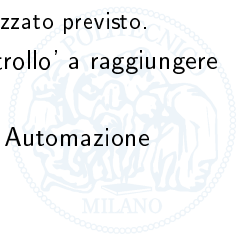


Principali componenti hardware dei sistemi di controllo

Il PLC (Programmable Logic Controller)

- Vediamo alcune definizioni secondo lo standard IEC 61131.
 - *PLC (Programmable Logic Controller)*: sistema elettronico a funzionamento digitale, destinato all'uso in ambito industriale, che utilizza una memoria programmabile per l'archiviazione interna di istruzioni orientate all'utilizzazione per l'implementazione di funzioni specifiche, come quelle logiche, di sequenziamento, di temporizzazione, di conteggio e di calcolo aritmetico, per controllare, mediante ingressi ed uscite sia digitali che analogici, vari tipi di macchine e processi.
 - *Sistema PLC*: configurazione realizzata dall'utilizzatore, formata da un PLC e dalle periferiche associate, necessaria al sistema automatizzato previsto.
- Storicamente, il PLC è stato il primo 'hw dedicato al controllo' a raggiungere una diffusione davvero capillare nell'industria.

Di PLC, IEC 61131 *et similia* si parla estesamente nel corso di Automazione Industriale, quindi ora non andiamo oltre.



Sistemi di controllo distribuiti (DCS)

Concetto di DCS e cenni storici

- Idea fondamentale: distribuire nell'impianto le funzioni di calcolo del controllo a livello di loop, controllo logico, monitoraggio, gestione degli allarmi, trend, log, reporting.
- Implementazione: rete di diversi dispositivi, alcuni dei quali dotati di 'intelligenza' (master) e altri asserviti ai primi (slave).
- I DCS sono di fatto reti multilivello di dispositivi funzionalmente eterogenei.
- I DCS esistono all'incirca dagli anni 70-80 (si può discutere su quale evento ne detti l'inizio, ma per noi è inessenziale).
- Le principali motivazioni del loro grande successo sono
 - la minore necessità di cablaggi (non c'è più un unico punto cui fanno capo tutti i segnali);
 - la distribuzione del carico computazionale;
 - la facilità di costruire sistemi ridondanti;
 - la possibilità di dotare unità di processo del loro controllo già 'in fabbrica'.



Sistemi di controllo distribuiti (DCS)

I DCS nel panorama attuale dell'informatica industriale e dell'automazione

- Molte grandi aziende sono state fautrici del paradigma DCS: Allen-Bradley, Bailey, Hartmann & Braun, ABB Automation, Foxboro, Honeywell, National Instruments, Siemens,...
- Tuttavia, fino a relativamente pochi anni fa (ovvero fino alla diffusione dei bus di campo) l'interoperabilità tra DCS diversi era un problema non banale.
- L'idea di 'distribuire' il controllo è tuttora di successo e si avverte una sempre maggiore necessità d'integrazione tra sistemi diversi, soprattutto da parte dell'utenza che gestisce grandi impianti.
- Grande impulso al concetto di 'controllo distribuito' proviene dalla standardizzazione delle comunicazioni tramite bus di campo (field bus, di cui si dirà più avanti).
- Vi sono tuttora problemi operativi dati dalla difficoltà del tipico personale che installa i sistemi di controllo a capire e gestire correttamente le caratteristiche peculiari di un sistema di controllo con comunicazioni 'via rete' (uno studio del 1994 indicava che circa il 20% dei malfunzionamenti dei sistemi di controllo di processo sono da attribuirsi ad una gestione non corretta dei ritardi di comunicazione).

Sistemi di controllo distribuiti (DCS)

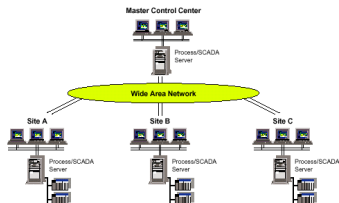
Tipica struttura di un sistema di controllo basato su DCS

- Si tratta di una rete multilivello (si vedrà bene negli esempi che seguono) composta, con una terminologia generale e abbastanza condivisa e andando dai livelli superiori agli inferiori (ovvero 'dalla sala controllo all'impianto'), da
 - 'Data highway' d'impianto,
 - Moduli di comunicazione,
 - Moduli di controllo,
 - Bus locale di I/O,
 - Moduli di I/O.
- Vi sono coinvolti diversi standard: CANopen, ControlNet, DeviceNet, Ethernet, Fieldbus, Interbus, Profibus,...
- Vi sono anche diverse problematiche: sincronizzazione e determinismo scelta tra polling, token passing e report by exception, topologia di rete, presenza di protocolli eterogenei,...

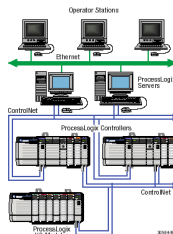


Sistemi di controllo distribuiti (DCS)

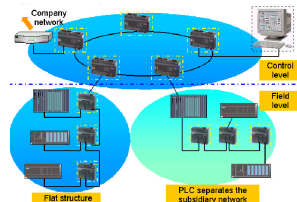
Alcuni esempi



Honeywell
PlantScape



Allen Bradley
ProcessLogix



Siemens
SIMATIC-net

Bus di campo (Field Bus)

Concetti generali e principali standard

- Già nei DCS si è introdotta l'idea di 'intelligenza distribuita' nei sistemi di controllo, limitandola però alle sole CPU e/o ai moduli di acquisizione remota e realizzando la connessione di sensori ed attuatori ancora con le classiche tecniche analogiche.
- L'idea di 'bus di campo' è in un certo senso un'evoluzione della precedente: tutti i componenti di un sistema di controllo, dal PC per l'interfaccia operatore al sensore o all'attuatore, sono pensati come dispositivi in rete. In questo modo, concettualmente, un solo 'cavo' (o, appunto, bus) percorre tutto l'impianto e connette tutti i dispositivi.
- Non è possibile (e non sarebbe nemmeno utile) una trattazione esaustiva dell'esistente. Ci limitiamo quindi a menzionare in sintesi tre standard significativi:
 - PROFIBUS (European Fieldbus Standard IEN 50170), di grande rilevanza industriale e ampiamente supportato;
 - CAN (Controlled Area Network), piuttosto 'datato' ma di grande importanza soprattutto nel settore veicolare;
 - OPC (OLE for Process Control), importante per il suo ruolo nell'integrazione del PC nei sistemi di controllo distribuiti.

Bus di campo (Field Bus)

Ruolo dei dispositivi nella tipica architettura Field Bus

- Tradizionalmente, il ruolo di vari dispositivi di un sistema di controllo era univocamente definito dal tipo di dispositivo stesso (ad esempio, un sensore misura e trasmette un valore).
- Con l'adozione dei bus di campo, il fatto di avere ovunque interfacce 'di rete' e, potenzialmente, 'intelligenza', ha fatto e fa sì che l'elaborazione (in senso lato, ovvero comprendendo in essa - ad esempio - anche aspetti di condizionamento dei segnali) sia ancor più distribuita. Ad esempio, vi sono attuatori che segnalano in anticipo possibili malfunzionamenti sulla base di misure che fanno su sé stessi.
- Questo ha certamente dei vantaggi (scarica le CPU 'centrali', permette una gestione più economica,...) ma anche dei potenziali svantaggi, il primo dei quali è che se si rompe il cavo di un sensore tradizionale si perde un segnale, se si rompe il bus si perde un pezzo d'impianto.
- L'adozione del field bus costringe a ripensare il modo in cui si costruiscono i sistemi di controllo, dovendo ad esempio dimensionare 'cose nuove' come la rete.

Bus di campo (Field Bus)

Vantaggi e svantaggi dell'adozione dei Field Bus

- Vantaggi principali:
Cablaggio più semplice, (ri)configurazione in buona parte via software, diagnostica evoluta, manutenzione soltanto in caso di necessità, integrazione con reti eterogenee, interoperabilità, intercambiabilità, standardizzazione,...
- Svantaggi principali:
Necessità di competenze (almeno in fase di messa in servizio) non sempre disponibili, necessità (nel caso si migri un impianto esistente) di sostituire parte dei dispositivi, maggior costo di hardware e software...
- L'adozione dei field bus incontra ancor oggi nell'industria resistenze significative, ma si tratta senza dubbio - insieme ad altre - della tecnologia 'del futuro' nell'architettura dei sistemi di controllo.



Aspetti normativi (standard IEC)

- Data la crescente complessità delle architetture per il controllo, sono evidentemente necessari degli standard sia per la comunicazione che per la specifica e l'implementazione del software di controllo.
- Per questo esistono opportune normative, delle quali elenchiamo le due principali:
 - IEC 61131, che definisce uno standard per i sistemi di controllo logico occupandosi delle specifiche dei dispositivi, dei linguaggi di programmazione, dei protocolli per la comunicazione,...
 - Della normativa IEC61131 si parla nel corso di Automazione Industriale.
 - IEC 61499, che riguarda specificamente i sistemi di controllo distribuiti e li descrive con uno schema a tre livelli:
 - *system model*, dove si vede quali applicazioni vi sono, quali dispositivi queste interessano e come esse comunicano tra loro, col processo ed eventualmente con livelli superiori del sistema;
 - *device model*, dove si vede quali interfacce di comunicazione (analogiche e/o digitali) e quali risorse (CPU, memoria,...) ogni dispositivo mette a disposizione e da quale/i applicazione/i le risorse sono usate;
 - *resource and application model(s)*, dove si vedono - entro ogni dispositivo - le 'local applications' (applicazioni o parti di esse allocate sul dispositivo), quali interfacce (di comunicazione verso il processo o altri dispositivi) le applicazioni hanno e di quali blocchi funzionali (Function Blocks) esse sono composte.

Strumenti per la configurazione e la programmazione

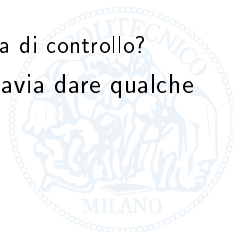
Rassegna minima dei tipi di strumenti usati

- La costruzione di un sistema di controllo comprendente diversi dispositivi si articola, in sintesi, nelle seguenti fasi:
 - installazione e collegamento fisico dei dispositivi;
 - 'configurazione' del sistema, ovvero creazione
 - delle connessioni logiche tra i dispositivi (cioè della rete)
 - e del 'database d'impianto' (cioè dell'insieme di tutti gli oggetti quali variabili, parametri e così via, ognuno identificato da un tag univoco);
 - 'programmazione' del sistema, ovvero
 - creazione delle applicazioni di controllo,
 - loro deployment, eventualmente distribuito, sui dispositivi che dovranno eseguirle
 - e creazione delle interfacce operatore (pannelli, sinottici e così via).
- Vi sono strumenti software per ognuna di queste operazioni (tranne, ovviamente, la prima).
- Sono sempre più diffusi ambienti che permettono di eseguire tutte queste operazioni in modo unitario, consistente e guidato, anche - grazie agli standard - in sistemi fisicamente eterogenei.

Influenza dell'architettura sul progetto del controllo

Premessa

- Nel progetto di un sistema di controllo si ragiona solitamente sui modelli (dell'apparato da controllare e, non lo si dimentichi mai, del regolatore).
- Questo definisce la situazione 'ideale' in cui il comportamento del regolatore fisicamente implementato corrisponde perfettamente a quello del suo modello.
- Tuttavia, ciò non è mai del tutto vero. Dobbiamo allora porci delle domande.
 - Quali sono i fenomeni che fanno sì che un regolatore non si comporti come il suo modello?
 - Quanto sono importanti?
 - Si possono prevenire con un progetto accurato del sistema di controllo?
- Tutto ciò è trattato in dettaglio in corsi più avanzati, tuttavia dare qualche cenno al riguardo ci serve anche qui.



Influenza dell'architettura sul progetto del controllo

- Un regolatore 'non è uguale alla sua funzione di trasferimento $R(s)$ ' per diverse ragioni:
 - discretizzazione temporale, campionamento e tenuta (cose note),
 - quantizzazione dei segnali (cosa abbastanza nota),
 - precisione finita della macchina (cosa molto meno nota e spesso sottovalutata),
 - problemi di comunicazione sulla rete di processo, che si manifestano essenzialmente come imperfezioni nella sincronizzazione dei dispositivi e disuniformità temporali nel passo di controllo,
 - ...
- Analizziamo in breve le principali.



Influenza dell'architettura sul progetto del controllo

- Discretizzazione temporale, campionamento e tenuta
 - Problema principale: scelta incorretta del tempo di campionamento.
 - Effetti: aliasing, eccessiva riduzione del grado di stabilità del loop, alterazione delle caratteristiche di stabilità del regolatore (a seconda del metodo di discretizzazione), errata rappresentazione delle dinamiche veloci del regolatore,...
 - Come si affronta: il passo di campionamento va scelto con oculatezza e non sempre lo si fa (la casistica industriale lo mostra).
- Quantizzazione dei segnali
 - Problema principale: scelta incorretta dei convertitori, soprattutto in termini di risoluzione.
 - Effetti: insorgenza di cicli limite, piccole e rapide fluttuazioni del segnale di controllo, impossibilità d'implementare certe parametrizzazioni del regolatore (ad esempio una data banda proporzionale in un PI),...
 - Come si affronta: curando il codice e dimensionando bene la precisione dei convertitori.

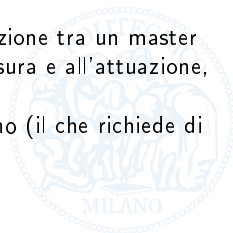
Influenza dell'architettura sul progetto del controllo

- Precisione finita della macchina

- Problema principale: non se ne considera l'influenza.
- Effetti: alterazione delle caratteristiche dinamiche del regolatore al variare del passo di campionamento, alterazione della struttura stessa del regolatore (un tipico esempio è la 'scomparsa' dell'azione integrale in un PI, cosa che vedremo),...
- Come si affronta: quando si passa dal continuo al discreto e si usa una macchina a precisione finita, si tiene conto delle due cose perché s'intrecciano e possono produrre risultati indesiderati.

- Ritardi (tipico dei sistemi distribuiti)

- Problema principale: ritardo dovuto all'errata sincronizzazione tra un master che calcola il controllo e gli slave che provvedono alla misura e all'attuazione,
- Effetti: riduzione del grado di stabilità dei loop coinvolti.
- Come si affronta: mettendo le sincronizzazioni che servono (il che richiede di capire quali servono) anche se sembra codice inutile.



Influenza dell'architettura sul progetto del controllo

- Alterazione della dinamica del regolatore (e quindi del loop) dovuta all'uso della trasmissione a 'exception report', ovvero a slave che 'trasmettono solo quando il valore è cambiato abbastanza (tipico dei sistemi distribuiti)
 - Problema principale: uso di tale metodo di trasmissione nei loop, nel tentativo di 'scaricare la rete'.
 - Effetti: comportamenti inspiegabili (di fatto introdurre la trasmissione a exception report significa alterare completamente la parte di sistema coinvolta).
 - Come si affronta: nei loop la trasmissione a exception report non si usa e basta, e fa niente se non usandola 'si carica la rete'.
- Altri problemi:
 - fluttuazioni nel passo di aggiornamento del controllo dovute alla comunicazione;
 - gestione dei segnali logici di una struttura di controllo a livelli del sistema dove la risposta in tempo reale non è garantibile;
 - ...



Influenza dell'architettura sul progetto del controllo

Soluzioni tipicamente adottate per i problemi evidenziati

- Sovradimensionare (nel senso delle prestazioni) il sistema di controllo o sue parti accuratamente scelte: questa soluzione ovviamente funziona, ma costa ed equivale a usare la forza bruta.
- Eliminare le possibili disuniformità temporali dedicando alla chiusura degli anelli critici dispositivi configurati in modo da non risentire per costruzione delle influenze del resto dell'architettura. In generale, questo approccio porta alla costruzione di un sistema complessivo fatto da tanti sottosistemi di tipo embedded.
- Accettare il fatto che il comportamento del regolatore 'è un po' diverso da quello voluto' e, dal momento che un'incertezza sul regolatore può essere vista come un'incertezza tout court sul loop di controllo, trattarla come fosse nel processo con uno dei molti metodi esistenti in letteratura per questo scopo.

Scrivere software per il controllo ha a che fare con l'Automatica almeno tanto quanto con l'Informatica. La seconda serve, ma se non si sa la prima non si va da nessuna parte (e se invece la si sa si possiede una professionalità preziosa).

Regolatori industriali e loro implementazione software



Regolatori industriali e loro implementazione software



Regolatori industriali e loro implementazione software

Premessa

- Lo scopo di questa sezione del corso è
 - illustrare le tipiche funzionalità dei regolatori industriali
 - e imparare i rudimenti di come li si realizza tramite software, impiegando allo scopo un qualsiasi linguaggio di programmazione procedurale.
- Quindi, sempre 'al livello di questo corso', sarete in grado d'implementare una legge di controllo - ovvero di 'realizzare $R(s)$ - usando soltanto 'un processore e un compilatore'. Questo vi renderà padroni della materia in un modo assai più profondo di quanto lo diventereste usando strumenti dedicati e vi permetterà anche d'iniziare a capire come tali strumenti funzionano (ottima cultura per un informatico).
- Per tenerci appunto al livello del corso, ci limiteremo qui a trattare un regolatore di struttura piuttosto semplice, ovvero un PID. Le osservazioni che faremo saranno però generali.
- Il 'seguito' si trova in corsi avanzati, quali ad esempio Laboratorio di Automazione per Ingegneria Informatica (LS INF) e i corsi dedicati ai metodi numerici nell'automatica.

Realizzazione di un regolatore PID digitale

Specifiche

- Legge di controllo PID ISA a 2 gradi di libertà:

$$CS(s) = K \left[bSP(s) - PV(s) + \frac{1}{sT_i} (SP(s) - PV(s)) + \frac{sT_d}{1 + sT_d/N} (cSP(s) - PV(s)) \right].$$

- Significato dei simboli:

<i>SP</i>	Set Point, segnale di riferimento,
<i>PV</i>	Process Variable, variabile controllata,
<i>CS</i>	Control Signal, segnale di controllo,
<i>K</i>	guadagno,
<i>T_i</i>	tempo integrale,
<i>T_d</i>	tempo derivativo,
<i>N</i>	rapporto tra <i>T_d</i> e la costante di tempo del 'derivative filter',
<i>b</i>	peso del SP nell'azione proporzionale,
<i>c</i>	peso del SP nell'azione derivativa.



Realizzazione di un regolatore PID digitale

Specifiche

- Implementazione digitale in forma incrementale con passo di campionamento T_s e discretizzazione tramite il metodo di Eulero implicito.
- Antiwindup.
- Modalità di funzionamento:
 - *Automatico*, in cui CS è calcolato dalla legge PID,
 - *Manuale*, in cui CS è manovrato dall'operatore,
 - *Tracking*, in cui CS è vincolato a seguire un segnale esterno (TR, Track Reference), il che serve di solito quando ci sono più regolatori commutabili a controllare una singola variabile di controllo.
- Commutazione 'bumpless' (senza salti) tra le diverse modalità di funzionamento.



PID ISA a due gradi di libertà

- La legge PID ISA può scriversi come

$$CS(t) = P(t) + I(t) + D(t)$$

dove $P(t)$, $I(t)$ e $D(t)$ sono rispettivamente le azioni proporzionale, integrale e derivativa, espresse in termini di trasformata di Laplace come

$$P(s) = K (bSP(s) - PV(s))$$

$$I(s) = \frac{K}{sT_i} (SP(s) - PV(s))$$

$$D(s) = \frac{sT_d}{1 + sT_d/N} (cSP(s) - PV(s))$$



Implementazione digitale in forma incrementale

- L'implementazione incrementale corrisponde a calcolare ad ogni passo del tempo discreto k il segnale di controllo a tempo discreto $CS^*(k)$ come

$$CS^*(k) = CS^*(k-1) + \Delta CS^*(k)$$

dove l'operatore Δ è definito come

$$\Delta := 1 - z^{-1}$$

come vedremo in seguito, questo tipo d'implementazione facilita significativamente la realizzazione dell'antiwindup, delle modalità di funzionamento manuale e tracking e della commutazione bumpless.



Implementazione digitale in forma incrementale

- Discretizzando con il metodo di Eulero implicito il calcolo delle tre azioni si ha

$$P^*(z) = K(bSP^*(z) - PV^*(z))$$

$$I^*(z) = K \frac{1}{\frac{z-1}{zT_s} T_i} (SP^*(z) - PV^*(z))$$

$$D^*(z) = K \frac{\frac{z-1}{zT_s} T_d}{1 + \frac{z-1}{zT_s} \frac{T_d}{N}} (cSP^*(z) - PV^*(z))$$

- Quindi, evidenziando l'operatore Δ ,

$$P^*(z) = K(bSP^*(z) - PV^*(z))$$

$$I^*(z) = K \frac{T_s}{\Delta T_i} (SP^*(z) - PV^*(z))$$

$$D^*(z) = K \frac{T_d/T_s}{1 + \Delta T_d/NT_s} (c\Delta SP^*(z) - \Delta PV^*(z))$$



Implementazione digitale in forma incrementale

- Il calcolo dell'incremento delle azioni P e I è quindi semplicemente

$$\Delta P^*(z) = K (b \Delta SP^*(z) - \Delta PV^*(z))$$

$$\Delta I^*(z) = K \frac{T_s}{T_i} (SP^*(z) - PV^*(z))$$

- Per l'azione D, invece, si ha

$$\begin{aligned} \left(1 + \Delta \frac{T_d}{NT_s}\right) D^*(z) &= K \frac{T_d}{T_s} (c \Delta SP^*(z) - \Delta PV^*(z)) \\ \left(1 + \frac{T_d}{NT_s} - z^{-1} \frac{T_d}{NT_s}\right) D^*(z) &= K \frac{T_d}{T_s} (c \Delta SP^*(z) - \Delta PV^*(z)) \end{aligned}$$

...

$$(T_d + NT_s + z^{-1} T_d) D^*(z) = KNT_d (c \Delta SP^*(z) - \Delta PV^*(z))$$

Implementazione digitale in forma incrementale

- In sintesi risulta pertanto

$$\Delta P^*(k) = K(b\Delta SP^*(k) - \Delta PV^*(k))$$

$$\Delta I^*(k) = K \frac{T_s}{T_i} (SP^*(k) - PV^*(k))$$

$$D^*(k) = \frac{T_d \dot{D}^*(k-1) + KNT_d (c\Delta SP^*(k) - \Delta PV^*(k))}{T_d + NT_s}$$

$$\Delta D^*(k) = D^*(k) - D^*(k-1)$$

- Osservazioni:
 - il PID ISA ha *due* variabili di stato: una è legata all'integratore, l'altra al cosiddetto 'polo del derivatore reale';
 - tale 'secondo polo' ha per costruzione modulo minore di 1.



Implementazione digitale in forma incrementale

- Calcolo della variazione del segnale di controllo (il suffisso 'old' denota nel codice i valori delle variabili al passo di controllo precedente, mentre 'D' sta per ' Δ ')

$$\begin{aligned} \text{DSP} &= \text{SP} - \text{SPold}; \\ \text{DPV} &= \text{PV} - \text{PVold}; \\ \text{DP} &= K * (b * \text{DSP} - \text{DPV}); \\ \text{DI} &= K * T_s / T_i * (\text{SP} - \text{PV}); \\ \text{D} &= (T_d * \text{Dold} + K * N * T_d * (c * \text{DSP} - \text{DPV})) / (T_d + N * T_s); \\ \text{DD} &= \text{D} - \text{Dold}; \\ \text{DCS} &= \text{Dp} + \text{DI} + \text{DD}; \end{aligned}$$

- Questo è di fatto il calcolo del segnale di controllo in automatico; ora esaminiamo le altre modalità di funzionamento del regolatore.



Implementazione digitale in forma incrementale

Tracking

- Il tracking è semplice da implementare. Usiamo la sintassi C per comodità e, già che ci siamo, includiamo l'antiwindup avendo CSmax e CSmin come limiti di saturazione:

```
DSP  = SP-SPold;
DPV  = PV-PVold;
switch(MODE)
{
    case AUTO:      DP  = K*(b*DSP-DPV);
                   DI  = K*Ts/Ti*(SP-PV);
                   D   = (Td*Dold+K*N*Td*(c*DSP-DPV))/(Td+N*Ts);
                   DD  = D-Dold;
                   DCS = Dp+DI+DD;
                   CS  = CSold+DCS;
                   break;
    case TRACKING:  CS  = TR;
                   break;
}
if (CS>CSmax) CS = CSmax;
if (CS<CSmin) CS = CSmin;
SPold = SP;
PVold = PV;
CSold = CS;
```



- Il manuale si può fare in tanti modi, a seconda anche di come si vuol costruire l'interfaccia utente del regolatore: supponendo ad esempio che al passaggio in manuale CS si fermi dov'è e lo si possa incrementare o decrementare di una quantità DCSman con due comandi CSinc e CSdec si può operare come segue:

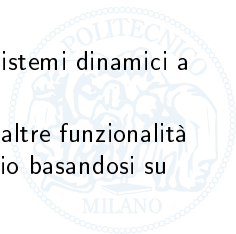
```
DSP = SP-SPold;
DPV = PV-PVold;
switch(MODE)
{
    case AUTO:    DP = K*(b*DSP-DPV);
                  DI = K*Ts/Ti*(SP-PV);
                  D = (Td*Dold+K*N*Td*(c*DSP-DPV))/(Td+N*Ts);
                  DD = D-Dold;
                  DCS = Dp+DI+DD;
                  CS = CSold+DCS;
                  break;
    case TRACKING: CS = TR;
                  break;
    case MANUAL:  if (CSinc) CS = CSold+DCSman;
                  if (CSdec) CS = CSold-DCSman;
                  break;
}
if (CS>CSmax) CS = CSmax;
if (CS<CSmin) CS = CSmin;
SPold = SP;
PVold = PV;
CSold = CS;
```



Implementazione digitale in forma incrementale

Conclusioni e ulteriori sviluppi

- Il codice appena descritto realizza la commutazione bumpless, dato che al passaggio da manuale o tracking in automatico la prima variazione di CS calcolata in automatico viene applicata all'ultimo valore ottenuto dal tracking o dal manuale. Verificate di aver ben capito la cosa e se no discutiamone.
- La commutazione da automatico a manuale è bumpless per costruzione, dato che CS si ferma all'ultimo valore calcolato in automatico.
- Non è invece garantita la commutazione bumpless da automatico a tracking, dato che questa in generale consegue a uno scollegamento dell'uscita del regolatore dall'effettivo ingresso dell'attuatore.
- Ora realizzeremo in Modelica alcuni regolatori, sia come sistemi dinamici a tempo continuo che come algoritmi digitali.
- Nel fare questo introdurremo (e implementeremo) alcune altre funzionalità tipiche dei regolatori industriali, che si comprendono meglio basandosi su qualche esempio.



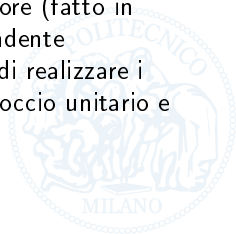
Esempi di regolatori in Modelica



Esempi di regolatori in Modelica

Premessa

- Implementeremo ora in Modelica due regolatori:
 - PI ISA a 1 g.d.l. con antiwindup (AW) e tracking (TRK) come sistema dinamico a tempo continuo;
 - PI ISA a 1 g.d.l. con AW e TRK come algoritmo digitale;
- Un terzo (un PID ISA a 2 g.d.l. con AW, TRK, manuale e blocchi in salita e discesa) sarà implementato, come algoritmo digitale, in laboratorio (in LabVIEW).
- In laboratorio vedremo anche come interfacciare il regolatore (fatto in LabVIEW) al 'mondo fisico', in modo che il ciclo comprendente modellizzazione e simulazione si chiuda con la possibilità di realizzare i regolatori progettati e messi a punto, il tutto con un approccio unitario e coerente.



Esempi di regolatori in Modelica

Premessa

- Oltre a vedere l'implementazione Modelica di alcuni regolatori, lo scopo della trattazione seguente è
 - mostrare come realizzare regolatori a tempo continuo e digitali con interfaccia uniforme, a tutto vantaggio della flessibilità nella simulazione;
 - mostrare anche che non sempre questo ha senso e che in particolare ne ha poco quando le funzionalità non lineari dell'*algoritmo* di controllo sono essenziali per la corretta simulazione, evidenziando in tal caso l'estrema utilità del poter 'mescolare' modelli a-causali e algoritmici.



Modelli Modelica dei regolatori

Premessa

- Anzitutto, però, risolviamo l'esercizio dato qualche lezione fa e consistente nella scrittura in Modelica del modello del sistema con funzione di trasferimento

$$\frac{Y(s)}{U(s)} = \frac{\mu}{1 + 2\frac{\xi}{\omega_n}s + \frac{s^2}{\omega_n^2}}$$

e nella sua simulazione per $0 \leq t \leq 10$ con $\mu = 1.5$, $\omega_n = 2$, $\xi = 0.4$, $u(t) = \text{sca}(t - 1)$ (ExerciseIntroModelica02.mo).

- Useremo poi il sistema del second'ordine appena descritto in Modelica per provare i regolatori che implementeremo.



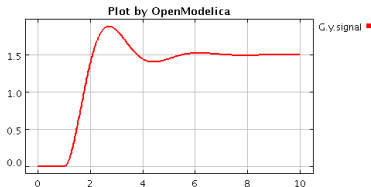
ExerciseIntroModelica02.mo

```
connector SignalIn
  input Real signal;
end SignalIn;

connector SignalOut
  output Real signal;
end SignalOut;

model TF2p0z
  SignalIn u;
  SignalOut y;
  parameter Real mu = 1 "gain";
  parameter Real wn = 0.2 "poles' natural frequency [r/s]";
  parameter Real csi = 0.5 "poles' damping factor [#]";
  Real dy;
equation
  dy = der(y.signal);
  mu*u.signal = der(dy)/wn^2+2*csi/wn*dy+y.signal;
end TF2p0z;

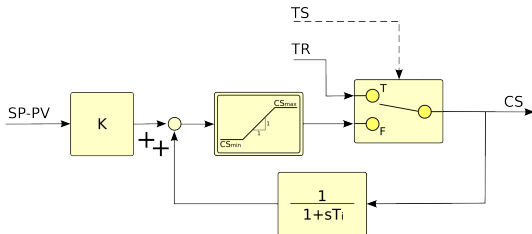
model ExerciseIntroModelica02
  TF2p0z G(mu=1.5,wn=2.0,csi=0.4);
equation
  G.u.signal = if time<1 then 0 else 1;
end ExerciseIntroModelica02;
```



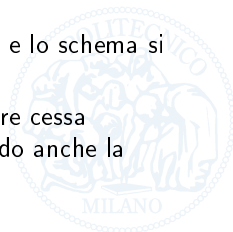
PI ISA a 1 g.d.l. con AW e TRK

Sistema dinamico a tempo continuo

Schema a blocchi:



- In assenza di saturazione e con $TS=false$ l'anello è chiuso e lo schema si comporta come un PI (verificalo).
- In saturazione o con $TS=true$ l'anello si apre e l'integratore cessa (correttamente) di esistere, evitando il windup e garantendo anche la commutazione bumpless da tracking ad automatico.



Modello Modelica (omettiamo le definizioni già date):

```
connector LogicalIn
  input Boolean flag;
end SignalIn;

connector LogicalOut
  output Boolean flag;
end SignalOut;

model PIIdofAwTrkCtime
  SignalIn SP;
  SignalIn PV;
  LogicalIn TS;
  SignalIn TR;
  SignalOut CS;
  parameter Real K      = 5 "Gain";
  parameter Real Ti     = 4 "Integral time [s]";
  parameter Real CSmax  = 1 "Maximum CS";
  parameter Real CSmin  = 0 "Minimum CS";
  parameter Real Ts     = 0.1 "Sampling time [s] (dummy, for compatibility with digital versions)";
protected
  Real satOut,fbOut;
equation
  CS.signal      = if TS.flag then TR.signal else satOut;
  satOut        = max(CSmin,min(CSmax,K*(SP.signal-PV.signal)+fbOut));
  Ti*der(fbOut)+fbOut = CS.signal;
end PIIdofAwTrkCtime;
```

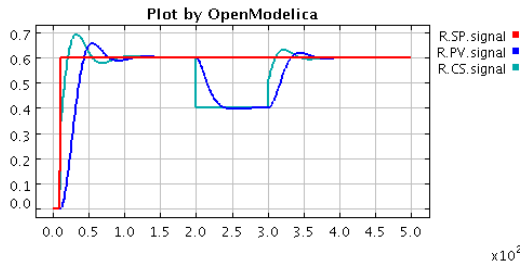


PI ISA a 1 g.d.l. con AW e TRK

Sistema dinamico a tempo continuo

Simulazione di prova:

```
model testPI1dofAwTrkCtime
  TF2p0z          P(mu=1,wn=0.1,csi=0.8);
  PI1dofAwTrkCtime R(K=0.5,Ti=8);
equation
  connect(P.y,R.PV);
  connect(R.CS,P.u);
  R.SP.signal = if time<10 then 0 else 0.6;
  R.TR.signal = 0.4;
  R.TS.flag   = if time>200 and time<300 then true else false;
end testPI1dofAwTrkCtime;
```

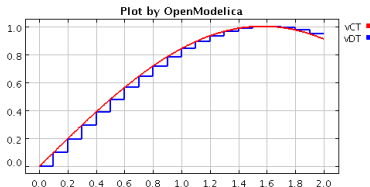


Sistemi digitali in Modelica

Premessa

- Il costrutto `when` permette d'introdurre in un modello delle *equazioni* da rendersi valide solo in certi istanti, tipicamente quelli in cui la funzione `sample` restituisce il valore true.
- Tipicamente tali equazioni coinvolgono variabili di tipo discreto, tali cioè da cambiare valore solo in certi istanti e mantenere negli altri l'ultimo acquisito
- L'esempio seguente chiarisce il significato e l'uso di `when` in unione a `sample`:

```
model demoWhenAndSample
  Real vCT;
  discrete Real vDT;
  parameter Real Ts = 0.1;
equation
  vCT = sin(time);
  when sample(0,Ts) then
    vDT = vCT;
  end when;
end demoWhenAndSample;
```



Sistemi digitali in Modelica

Premessa

- E' anche possibile scrivere in Modelica modelli 'algoritmici' dove la sezione `equation` è rimpiazzata da una sezione `algorithm` contenente un vero e proprio programma fatto da assegnamenti (per cui si usa il segno `:=`) invece che da equazioni e con tutte le istruzioni di controllo del flusso occorrenti.
- Si possono quindi implementare i regolatori ad esempio tramite `function` (come `sqrtReg`, ricordate?) e chiamarle in modelli algoritmici. Non lo facciamo in questo corso per ragioni di spazio, ma nella documentazione di (Open)Modelica si trovano tutte le informazioni che servono.
- Gli studenti interessati sono invitati a leggere prima il tutorial Peter Fitzson, Peter Bunus "Introduction to Object-Oriented Modeling and Simulation with OpenModelica" September 4, 2006 (disponibile in rete)



PI ISA a 1 g.d.l. con AW e TRK

Algoritmo digitale

```
model PIIdofAwTrkDigital
  SignalIn SP;
  SignalIn PV;
  LogicalIn TS;
  SignalIn TR;
  SignalOut CS;
  parameter Real K      = 5 "Gain";
  parameter Real Ti     = 4 "Integral time [s]";
  parameter Real CSmax  = 1 "Maximum CS";
  parameter Real CSmin  = 0 "Minimum CS";
  parameter Real Ts     = 0.4 "Sampling time [s]";

protected
  discrete Real sp;
  discrete Real dsp;
  discrete Real pv;
  discrete Real dpv;
  discrete Real cs;
  discrete Real dcs;

equation
  sp = SP.signal;
  pv = PV.signal;
  cs = CS.signal;
  when sample(0,Ts) then
    dsp = sp-pre(sp);
    dpv = pv-pre(pv);
    dcs = K*(dsp-dpv)+K*Ts/Ti*(sp-pv);
    cs = if TS.flag then TR.signal else max(CSmin,min(CSmax,pre(cs)+dcs));
  end when;
end PIIdofAwTrkDigital;
```

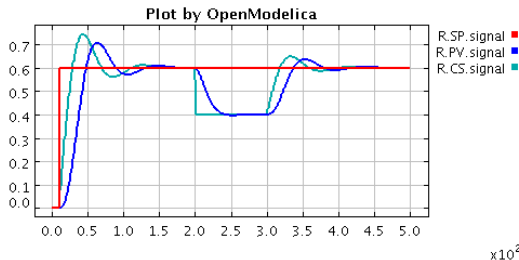


PI ISA a 1 g.d.l. con AW e TRK

Algoritmo digitale

Simulazione di prova:

```
model testPI1dofAwTrkDigital
  TF2p0z          P(mu=1,wn=0.1,csi=0.8);
  PI1dofAwTrkDigital R(K=0.5,Ti=8,Ts=2);
equation
  connect(P.y,R.PV);
  connect(R.CS,P.u);
  R.SP.signal = if time<10 then 0 else 0.6;
  R.TR.signal = 0.4;
  R.TS.flag   = if time>200 and time<300 then true else false;
end testPI1dofAwTrkDigital;
```



Esempio 'completo' di sistema di controllo in Modelica



Esempio di sistema di controllo in Modelica

Premessa

- Ora simuleremo un sistema 'completo' (processo+controllo), allo scopo di vedere come questo si fa in ambienti object-oriented come (Open)Modelica.
- Poi vedremo che il codice dei regolatori digitali si porta in un ambiente adatto all'effettiva implementazione del controllo su sistemi fisici in modo naturale.
- La notevolissima utilità di tutto questo per l'ingegneria del controllo dovrebbe esservi oramai chiara ed è una delle maggiori innovazioni dell'approccio object-oriented: soltanto pochi anni fa simulare un modello differenziale insieme a un algoritmo di controllo con la facilità con cui lo faremo adesso sarebbe stato semplicemente impensabile.



Esempio di sistema di controllo in Modelica

Premessa

- Il sistema che considereremo è fatto da tre serbatoi a pelo libero (Tank1, Tank2 e Tank3); Tank1 è alimentato da una pompa volumetrica che preleva fluido a pressione atmosferica; a valle dei tre serbatoi vi sono tre valvole lineari (nell'ordine V12, V23 e V3o, che scarica in atmosfera). Tre PI regolano il livello dei tre serbatoi agendo sulla valvola a valle.
- Disegnate lo schema qui sotto. Tra un minuto lo facciamo insieme.



Esempio di sistema di controllo in Modelica

Codice completo di ExampleControlled01.mo

```
connector Flange
  Real p "pressure at flange [Pa]";
  flow Real w "mass flowrate entering flange [kg/s]";
end Flange;

connector SignalIn
  input Real signal;
end SignalIn;
connector SignalOut
  output Real signal;
end SignalOut;

connector LogicalIn
  input Boolean flag;
end SignalIn;
connector LogicalOut
  output Boolean flag;
end SignalOut;

function sqrtReg
  input Real x;
  input Real delta=0.01 "Range of significant deviation from sqrt(x)";
  output Real y;
algorithm
  y := x/sqrt(sqrt(x*x+delta*delta));
end sqrtReg;
```



Esempio di sistema di controllo in Modelica

Codice completo di ExampleControlled01.mo

```
model FixedP
  Flange flange;
  parameter Real p=0 "flange fixed pressure [Pa]";
equation
  flange.p = p;
end FixedP;

model VolumetricPump
  Flange inlet;
  Flange outlet;
  SignalIn cmd;
  Real n "pump command [#, 0-1], 1 meaning rpm=nominal rpm";
  parameter Real n0 = 1000 "nominal rpm [#]";
  parameter Real w0 = 1 "mass flowrate at nominal rpm, i.e., n=1 [kg/s]";
equation
  n = min(1.0,max(cmd.signal,0.0));
  0 = inlet.w+outlet.w;
  inlet.w = w0*n;
end VolumetricPump;
```



Esempio di sistema di controllo in Modelica

Codice completo di ExampleControlled01.mo

```
model ValveWithLinearChar
  Flange inlet;
  Flange outlet;
  SignalIn cmd;
  Real x "valve opening command [#, 0-1]";
  parameter Real Cvmax = 0.01 "maximum Cv (x=1) [(kg/s)/sqrt(Pa)]";
equation
  x = min(1.0,max(cmd.signal,0.0));
  0 = inlet.w+outlet.w;
  0 = inlet.w-Cvmax*x*sqrtReg(inlet.p-outlet.p);
end ValveWithLinearChar;

model Tank2flanges
  Flange flange1;
  Flange flange2;
  Real l(start=l_start);
  Real p;
  Real M;
  parameter Real A      = 1 "tank base area [m^2]";
  parameter Real rho     = 1000 "fluid density [kg/m^3]";
  parameter Real l_start = 0 "initial fluid level [m]";
equation
  M = rho*A*l;
  der(M) = if M>=0 then flange1.w+flange2.w else 0;
  p = rho*9.81*l;
  p = flange1.p;
  p = flange2.p;
end Tank2flanges;
```



Esempio di sistema di controllo in Modelica

Codice completo di ExampleControlled01.mo

```
model PI1dofAW
  SignalIn SP;
  SignalIn PV;
  LogicalIn TS;
  SignalIn TR;
  SignalOut CS;
  Real satIn;
  Real linFBout(start=0);
  parameter Real K      = 1 "Gain";
  parameter Real Ti     = 10 "Integral time [s]";
  parameter Real CSmax  = 1 "Maximum CS";
  parameter Real CSmin  = 0 "Minimum CS";
  parameter Real Ts     = 0.1 "dummy Ts for digital replaceability";
equation
  satIn = K*(SP.signal-PV.signal)+linFBout;
  Ti*der(linFBout)+linFBout = CS.signal;
  CS.signal=max(CSmin,min(CSmax,satIn));
end PI1dofAW;
```



Esempio di sistema di controllo in Modelica

Codice completo di ExampleControlled01.mo

```
model PIIdofAwTrkDigital
  SignalIn SP;
  SignalIn PV;
  LogicalIn TS;
  SignalIn TR;
  SignalOut CS;
  parameter Real K      = 5 "Gain";
  parameter Real Ti     = 4 "Integral time [s]";
  parameter Real CSmax  = 1 "Maximum CS";
  parameter Real CSmin  = 0 "Minimum CS";
  parameter Real Ts     = 0.4 "Sampling time [s]";

protected
  discrete Real sp;
  discrete Real dsp;
  discrete Real pv;
  discrete Real dpv;
  discrete Real cs;
  discrete Real dcs;

equation
  sp = SP.signal;
  pv = PV.signal;
  cs = CS.signal;
  when sample(0,Ts) then
    dsp = sp-pre(sp);
    dpv = pv-pre(pv);
    dcs = K*(dsp-dpv)+K*Ts/Ti*(sp-pv);
    cs = if TS.flag then TR.signal else max(CSmin,min(CSmax,pre(cs)+dcs));
  end when;
end PIIdofAwTrkDigital;
```

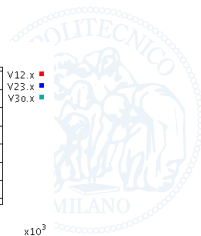
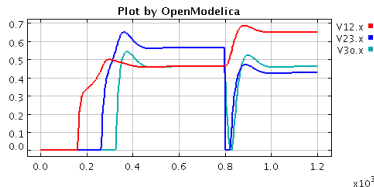
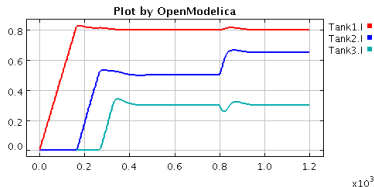


Esempio di sistema di controllo in Modelica

Codice completo di ExampleControlled01.mo

```
model ExampleControlled01
  FixedP source(p=0);
  FixedP sink(p=0);
  VolumetricPump Pump(w0=0.1);
  Tank2flanges Tank1(A=0.01);
  Tank2flanges Tank2(A=0.01);
  Tank2flanges Tank3(A=0.01);
  ValveWithLinearChar V12(Cvmax=2e-3);
  ValveWithLinearChar V23(Cvmax=2e-3);
  ValveWithLinearChar V3o(Cvmax=2e-3);
  PIIdofAW PI1(K=-8,Ti=40,Ts=10);
  PIIdofAW PI2(K=-8,Ti=40,Ts=10);
  PIIdofAW PI3(K=-8,Ti=40,Ts=10);
equation
  connect(source.flange,Pump.inlet);
  connect(Pump.outlet,Tank1.flange1);
  connect(Tank1.flange2,V12.inlet);
  connect(V12.outlet,Tank2.flange1);
  connect(Tank2.flange2,V23.inlet);
  connect(V23.outlet,Tank3.flange1);
  connect(Tank3.flange2,V3o.inlet);
  connect(V3o.outlet,sink.flange);
  connect(V12.cmd,PI11.CS);
  connect(V23.cmd,PI12.CS);
  connect(V3o.cmd,PI13.CS);
  PI11.TS.flag = false;
  PI12.TS.flag = false;
  PI13.TS.flag = false;
  PI11.TR.signal = 0;
  PI12.TR.signal = 0;
  PI13.TR.signal = 0;
  PI11.PV.signal = Tank1.l;
  PI12.PV.signal = Tank2.l;
  PI13.PV.signal = Tank3.l;
  PI11.SP.signal = 0.8;
  PI12.SP.signal = if time<800 then 0.5 else 0.65;
  PI13.SP.signal = 0.3;
  Pump.cmd.signal = 0.5;
end ExampleControlled01;
```

Simulazione per 1200 secondi (con regolatori a tempo continuo):



Esempio di sistema di controllo in Modelica

Codice completo di ExampleControlled01.mo

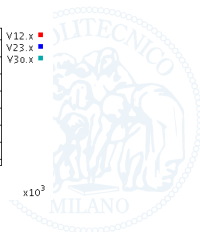
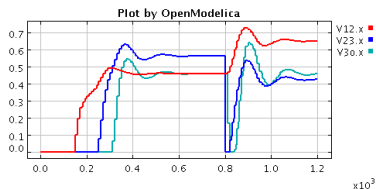
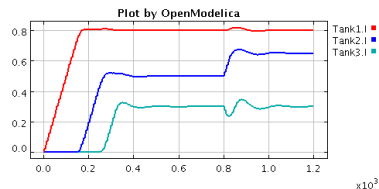
Per simulare lo stesso sistema ma con regolatori digitali basta sostituire le linee

```
PI1dofAW PI11(K=-8,Ti=40,Ts=10);  
PI1dofAW PI12(K=-8,Ti=40,Ts=10);  
PI1dofAW PI13(K=-8,Ti=40,Ts=10);
```

con

```
PI1dofAwTrkDigital PI11(K=-8,Ti=40,Ts=10);  
PI1dofAwTrkDigital PI12(K=-8,Ti=40,Ts=10);  
PI1dofAwTrkDigital PI13(K=-8,Ti=40,Ts=10);
```

Simulazione per 1200 secondi (con regolatori digitali):



Condizionamento dei segnali di processo e implementazione dei regolatori industriali

Questa esercitazione è svolta usando la lavagna e Scilab; non ci sono slide guida.



**Progetto, dimensionamento, simulazione e specifica d'implementazione
di un sistema di controllo di temperatura**



Laboratorio 1 (4LAB)

Introduzione rapida all'ambiente LabVIEW

**Realizzazione guidata di un regolatore PID industriale
in LabVIEW**



Esercizi di riepilogo

Questa esercitazione è svolta usando la lavagna; non ci sono slide guida.



Realizzazione e prova sperimentale di un sistema di controllo di temperatura

