Introduzione alla Teoria del Controllo Automatico

Controllo Retroazionato Proporzionale per Sistemi Dinamici Lineari Stazionari (LTI)

Naoki Sean Pross

Scuola Arti e Mestieri di Bellinzona

11 Maggio 2023

Indice

Indice

Applicazioni: Atterraggio di Space X

Convex Optimization for Trajectory Generation: A Tutorial on Generating Dynamically Feasible Trajectories Reliably and Efficiently

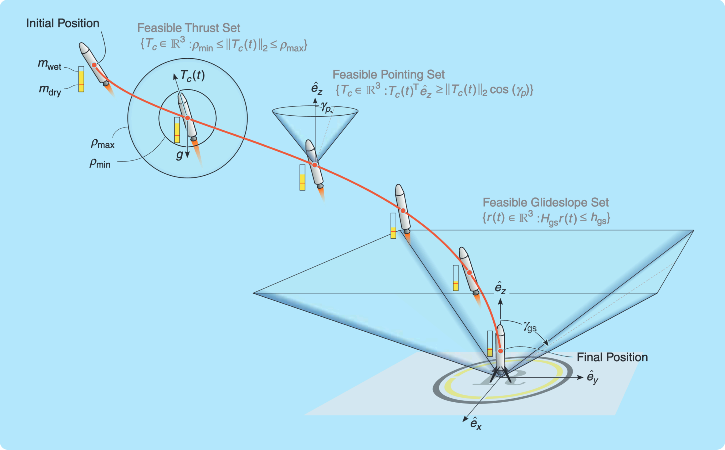


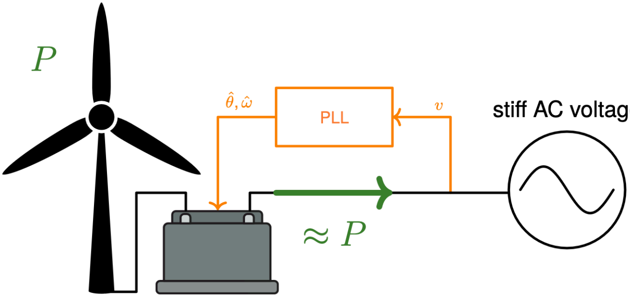
FIGURE 20 The three-degrees-of-freedom powered descent guidance problem, showing some of the relevant constraints on the rocket-powered lander's trajectory. The thrust direction $T_c(t)/\|T_c(t)\|_2$ servers as a proxy for the vehicle attitude.

https://ieeexplore.ieee.org/document/9905530

Applicazioni: Stabilità della Rete Elettrica

Control of Power Converters in Low-Inertia Power Systems

Limitations of grid-following control



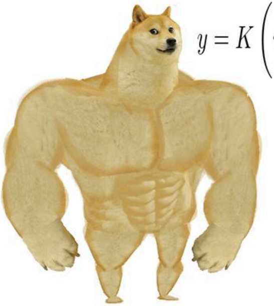
- *is good for* transferring power to a strong grid (what if everyone follows?)
- *is not good for* providing a voltage reference, stabilization, or black start
- tomorrow's grid needs ***grid-forming control*** $\equiv$ *emergence of synchronization*

https://people.ee.ethz.ch/~floriand/docs/Slides/Dorfler_MED_2019_Extended.pdf

Obiettivo di Oggi

Proportional Integral Feedback Control

$$y = K\left(u + \int_0^t \frac{u\, d\tau}{T}\right)$$



Open Loop Control (you)

```
setMotorSpeed(motorA, 100);
delay(500);
setMotorSpeed(motorB, 100);
delay(500);
...
```

$$U^\star(x(k)) := \underset{U_k}{\text{argmin}}\ l_f(x_N) + \sum_{i=0}^{N-1} l$$

subj. to $x_k = x(k)$

$x_{k+i+1} = Ax_{k}$

$x_{k+i} \in \mathcal{X}$

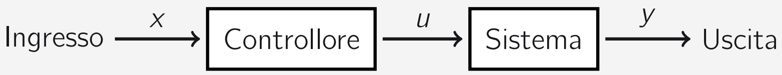
$u_{k+i} \in \mathcal{U}$

$U_k = \{u_k, u_{k+}$

Indice

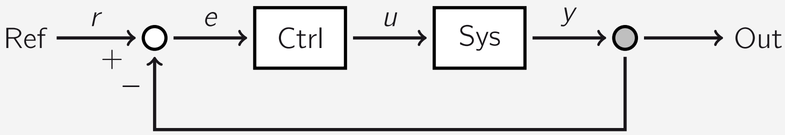
Strategie di Controllo

Open Loop



- Il controllore non sa se il valore d'ingresso x è risultato in un valore corretto all'uscita y

Closed Loop (Retroazione)

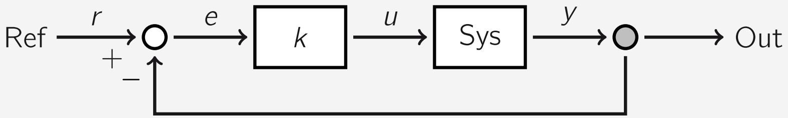


- Sostituiamo l'ingresso x con una referenza r e controlliamo l'errore e

Indice

Controllo a Retroazione Proporzionale

Closed Loop (Retroazione)

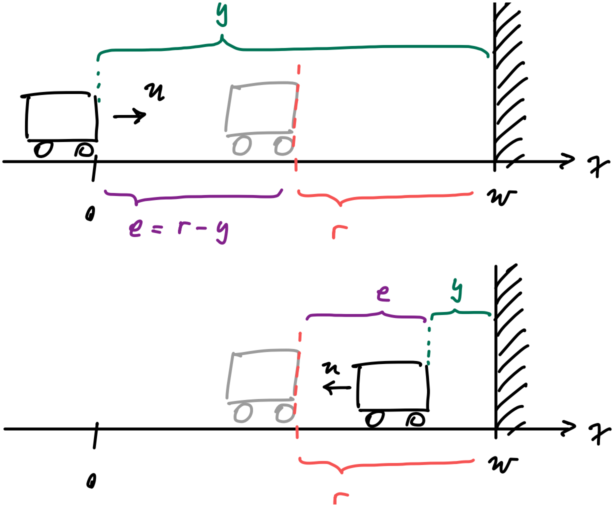


Controllo Proporzionale

Il controllo u è proporzionale all'errore

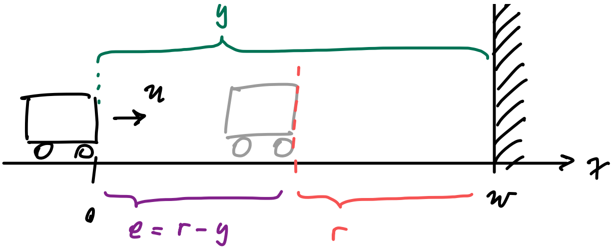
$$e = r - y \quad \rightsquigarrow \quad u = ke = k(r - y) \qquad k \in \mathbb{R}$$

Esempio: Posizionamento di un Robot (1/7)



$y \equiv$ Distanza dalla parete, $\quad u \equiv$ Velocità

Esempio: Posizionamento di un Robot (2/7)

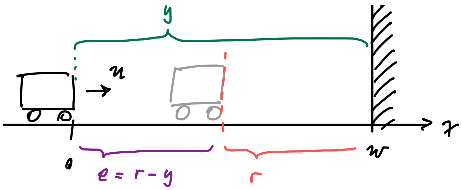


Modello del Robot

Il nostro modello assume che usiamo

$$u_i = k(r - y_i) \quad \text{per controllare} \quad x_{i+1} = x_i + u_i \Delta T.$$

Esempio: Posizionamento di un Robot (3/7)



Modello e dinamica

- Misurazioni di y ogni ΔT. Controllo della velocità:

$$u = \frac{y - r}{\Delta T} = \frac{-e}{\Delta T} \qquad \Longrightarrow \qquad k = \frac{-1}{\Delta T}$$

- Il motore ha una velocità massima $u_{\max}$ e minima $u_{\min}$:

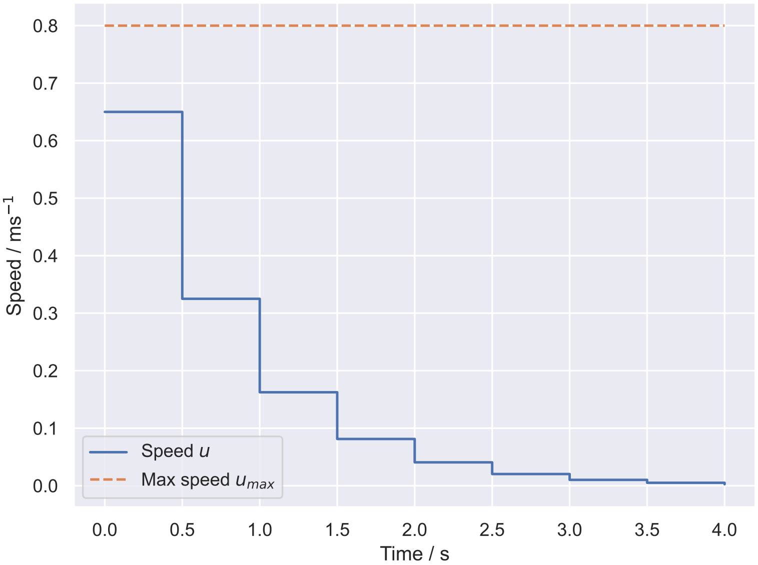
$$u = \max\left\{ u_{\min},\ \min\left\{ \frac{y - r}{N \Delta T},\ u_{\max} \right\} \right\}, \qquad N \in \mathbb{N}$$

Esempio: Posizionamento di un Robot (4/7)

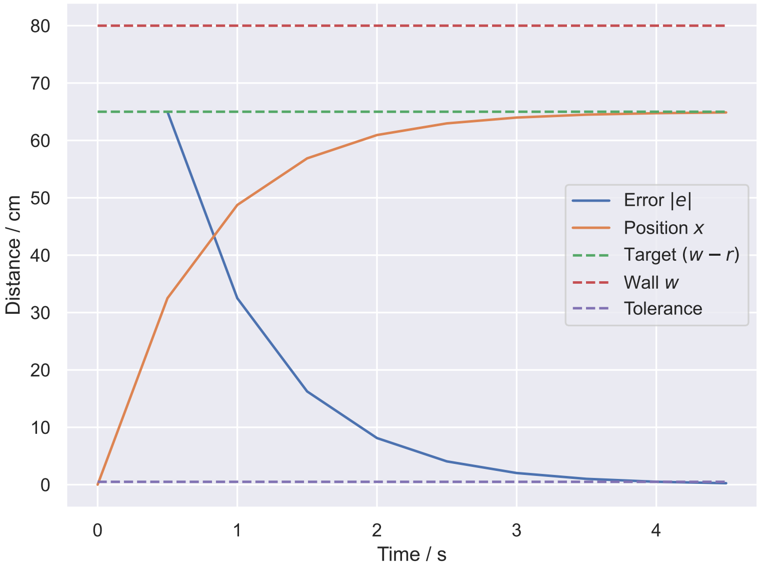
Implementazione in RobotC

```
const double dt = 500; // timestep in ms
const double n = 4; // then 500 ms * 4 = 2s
const double r = 150; // desired distance in mm
const double tol = 5; // allowed error tolerance in mm
// measurement, error in mm and velocity in mm/ms = m/s
double y, e, u;
e = INFINITY;
while (fabs(e) > tol) {
  y = getUSDistance(distanceCM); // measurement
  e = r - y; // error
  u = fmax(umin, fmin(-e / (n * dt), umax)); // control
  setMotorSpeed(scaleToMotorSpeed(u));
  sleep(dt);
}
setMotorSpeed(0);
```

Esempio: Posizionamento di un Robot (5/7)

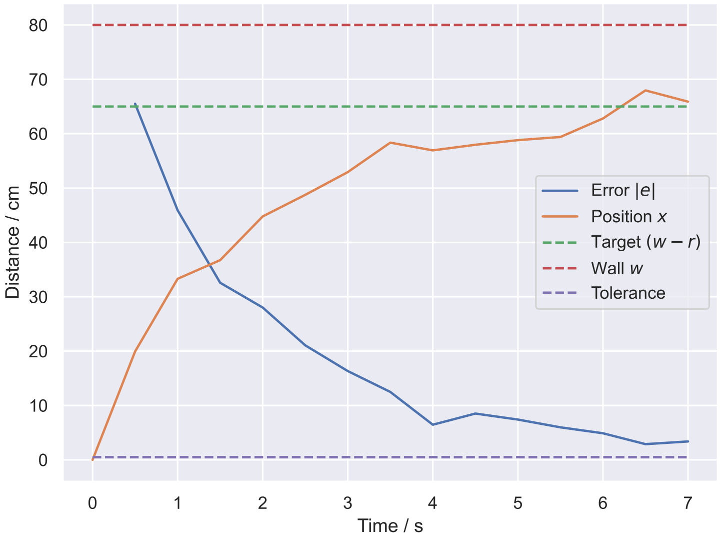


Esempio: Posizionamento di un Robot (6/7)



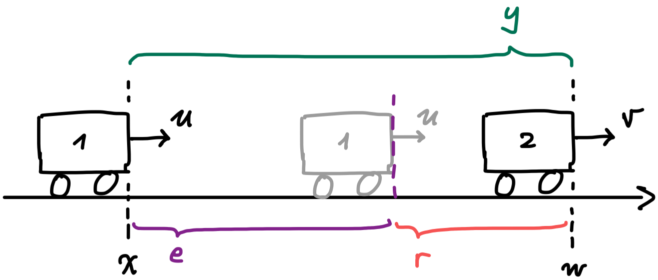
Esempio: Posizionamento di un Robot (7/7)

Più realisticamente con del rumore



Indice

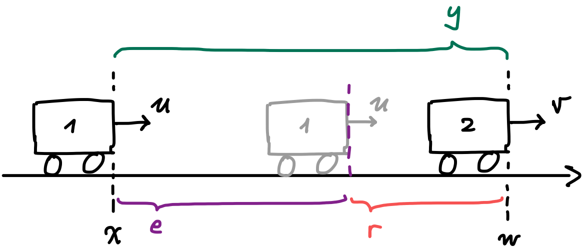
Problema della Retroazione Proporzionale (1/5)



Nuovo Problema: Insegumento

Controlla la velocità u per seguire un secondo carrello che si muove a velocità v. Il carrello (1) parte fermo.

Problema della Retroazione Proporzionale (2/5)

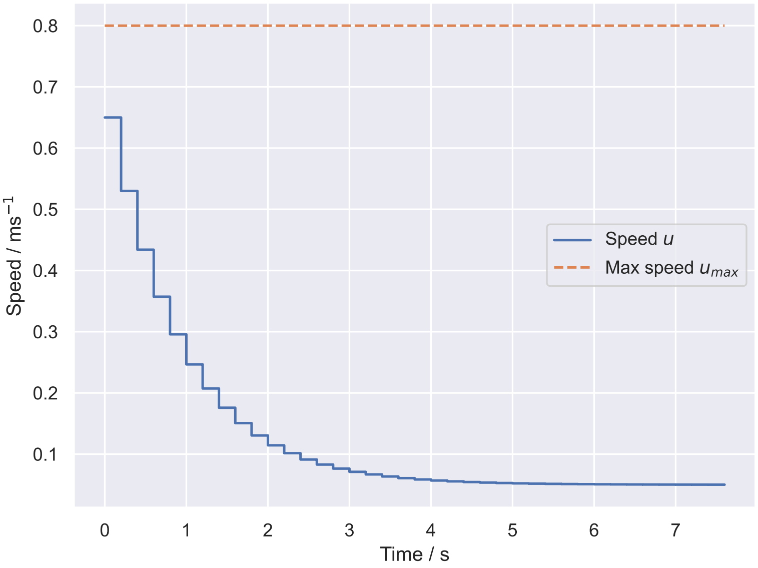


Retroazione Proporzionale

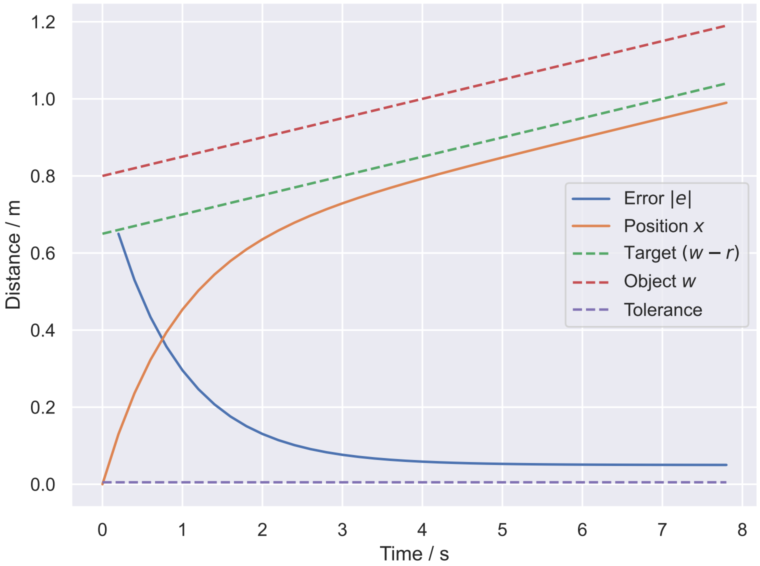
Formulazione in tempo discreto:

$$
\begin{aligned}
\text{Imposta} \qquad & u_i = k(r - y_i) \\
\text{per controllare} \qquad & x_{i+1} = x_i + u_i \Delta T \\
\text{mentre} \qquad & w_{i+1} = w_i + v \Delta T
\end{aligned}
$$

Problema della Retroazione Proporzionale (3/5)



Problema della Retroazione Proporzionale (4/5)



Problema della Retroazione Proporzionale (5/5)

Perché non funziona?

2° paradosso di Zenone (Achille e la tartaruga). Formalmente: Teorema del valore finale $\rightsquigarrow$ Trasformazione di Laplace $\rightsquigarrow$ Università.

$$e_\infty = \lim_{s \to 0} s \int_{\mathbb{R}^+} \left[y(t) - r(t) \right] e^{-st} \, dt \neq 0$$

Errore Stazionario

Un controllore di tipo P ha un errore stazionario se

- La referenza cambia in modo lineare (o più veloce)
- La dinamica del sistema cambia troppo velocemente (pendolo inverso, drone, ...)

Controllo Proporzionale-Integrale (1/2)

Controllo Integrale

- In pratica abbiamo una sequenza di misurazioni $y_0, y_1, \ldots, y_i$ separate da ΔT. Dunque anche una sequenza di errori $e_0, e_1, e_2, \ldots$, e una sequenza di controllo $u_0, u_1, u_2, \ldots$
- Aggiungendo l'errore precedente a quello attuale si forma un accumulatore (errore cumulativo):

$$\mathcal{E}_i = e_i + \mathcal{E}_{i-1}, \qquad \mathcal{E}_0 = 0.$$

- Impostiamo il controllo proporzionale all'errore cumulativo $u_i = k\mathcal{E}_i$ (i.e. controllo integrale)

Errore Cumulativo (Integrale)

- Se $e_i > 0 \implies \mathcal{E}_i$ aumenta $\implies u_i = k\mathcal{E}_i$ aumenta
- Se $e_i < 0 \implies \mathcal{E}_i$ diminuisce $\implies u_i = k\mathcal{E}_i$ diminuisce

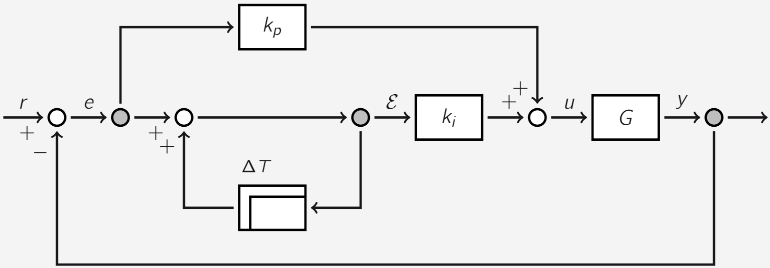
Controllo Proporzionale-Integrale (2/2)

Controllo PI

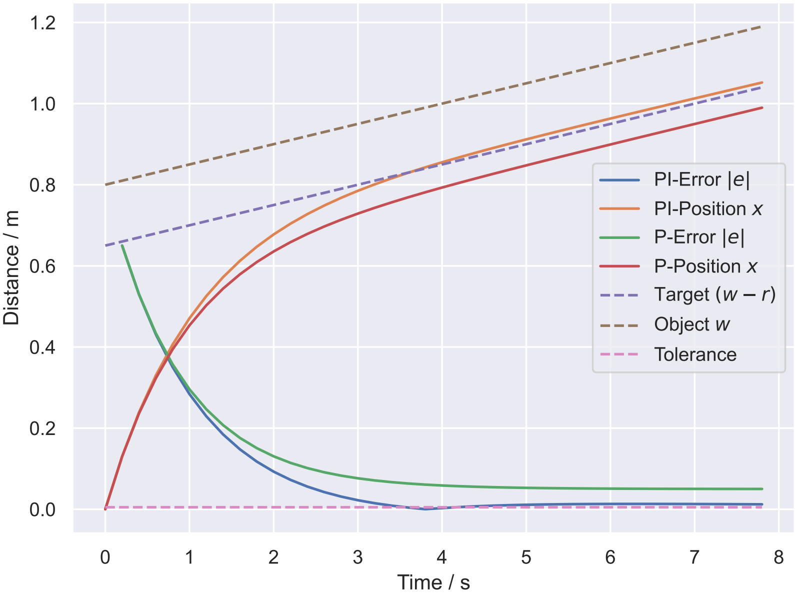
La legge di controllo proporzionale-integrale è data da

$$\mathcal{E}_i = r - y_i + \mathcal{E}_{i-1} \qquad \text{(accumulatore)}$$

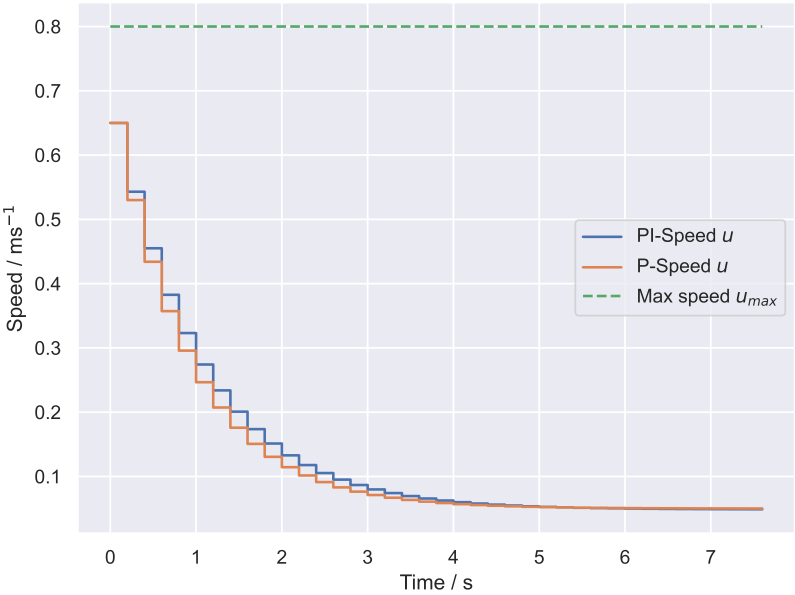
$$u_i = k_p(r - y_i) + k_i\mathcal{E}_i \qquad \text{(controllo)}$$



Esempio: Inseguimento di un Robot (1/2)



Esempio: Inseguimento di un Robot (2/2)



Esempio: Controllo PI

Implementazione in RobotC

```
// also define umin, umax, n, dt
const double kp = -1 / (n * dt);
const double ki = kp / 100;
double y, err, acc, u;

err = INFINITY;
acc = 0;
while (true) {
  y = getUSDistance(distanceCM);
  err = r - y;
  acc += err;
  u = fmax(umin, fmin(kp*err + ki*acc, umax));
  setMotorSpeed(scaleToMotorSpeed(u));
  sleep(dt);
}
```

Indice

Controllo PD, PID

Controllo Proporzionale-Derivativo

Cosa succede se si tiene in considerazione la variazione dell'errore?

$$\epsilon_i = r - y_i - e_{i-1} \qquad \text{(cambiamento dell'errore)}$$
$$u_i = k_p(r - y_i) + k_d \epsilon_i \qquad \text{(controllo)}$$

Controllo PID

Combina il controllo integrale e derivativo:

$$\mathcal{E}_i = r - y_i + \mathcal{E}_{i-1} \qquad \text{(accumulatore)}$$
$$\epsilon_i = r - y_i - e_{i-1} \qquad \text{(derivatore)}$$
$$u_i = k_p(r - y_i) + k_i \mathcal{E}_i + k_d \epsilon_i \qquad \text{(controllo)}$$

Controllo Automatico

Panoramica

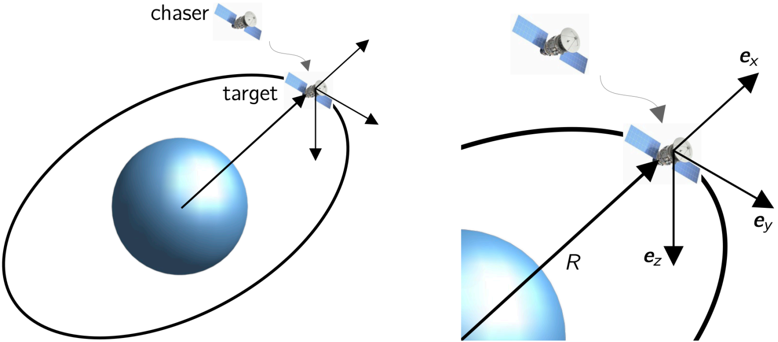
- **Controllo classico**: Sistemi SISO, PID, funzioni di trasferimento, diagrammi di Bode, Nyquist.
- **Controllo moderno**: Rappresentazione in spazio di stato
- **Controllo ottimale**: Indice di ottimalità (Hamiltoniano / Lagrangiano) da minimizzare: $\min_u J(u)$, teoria dei giochi
- **Controllo robusto**: Considerazioni stocastiche, introduzione di modelli probabilistici
- **Model Predictive Control**: Il controllore simula il sistema per trovare l'ingresso di controllo ottimale $u^\star = \arg\min_u J(u)$
- **Data Driven Predictive Control / Reinforcement Learning**: Creare un controllore senza avere un modello del sistema
- **Controllo non lineare**: Sistemi dinamici complessi, teoria del caos, teoria delle biforcazioni

Applicazioni del Controllo Automatico

Esempi di Applicazioni

- **Ingeneria Aerospaziale**: Navigazione orbitale; Controllo della propulsione; Ottimizzazione del consumo di carburante, ...
- **Industria Pertrolchimica / Farmaceutica**: Aggiungere reagenti in una soluzione mantenendo una concentrazione costante
- **Ingeneria Elettrica di Potenza**: Controllo della frequenza AC; Regolazione dei motori in base alla domanda; Alternatori per conversione AC/DC; Modello di controllo del flusso nei bacini idrografici (dighe)
- **Ingegneria Elettronica**: DC-DC buck converters, filtri adattivi, noise cancelling, circuiti per sistemi wireless (WiFi, BLE, RADAR, LiDAR)
- **Ingegneria Civile**: Controllo della temperatura di un edificio in base alla meteo; Sistemi antisismici; Decisione della logica per i semafori (controllo del traffico)
- **Medicina / Farmaceutica**: Modello a n-fasi per la somministrazione di un medicamento nel corpo umano; Chirurgia LASER per gli occhi
- **Ingegneria Meccanica**: Riduzione delle oscillazioni di un impianto; Controllo macchine CNC; Veicoli autonomi

Fine



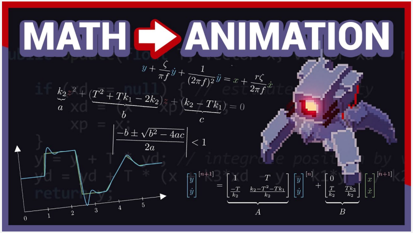
*Dr. Zellinger: Model Predictive Control Programming Exercise "Control for Spacecraft Rendezvous" (ETH Zürich **151-0660-00L**)*

Diapositive

- https://s.0hm.ch/hMokq
- https://files.thearcway.org/naopross/school/samb/SAMB_PI_Control.pdf

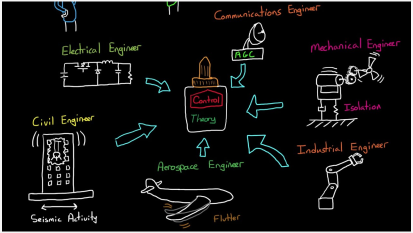
Vuoi saperne di più?

Giving Personality to Procedural Animations using Math



https://youtu.be/KPoeNZZ6H4s

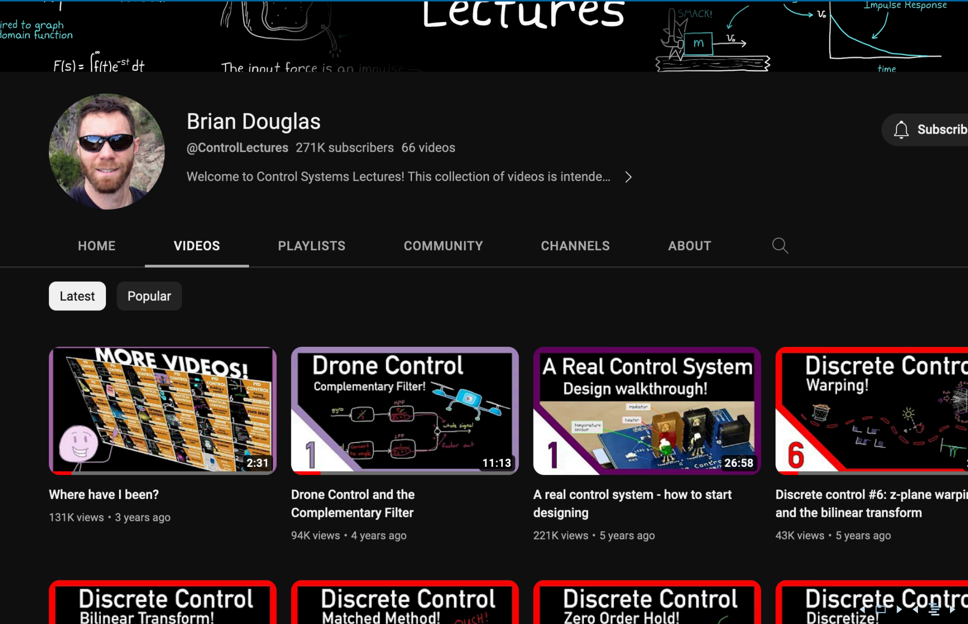
Why Learn Control Theory



https://youtu.be/oBc_BHxw78s

Vuoi saperne di più?

https://www.youtube.com/@ControlLectures/videos



Note di Pedanteria

Questa diapositiva non fa parte della presentazione, esiste unicamente per calmare le persone pignole

- Tecnicamente siccome stiamo lavorando in tempo discreto (i.e. campioni digitali ogni ΔT) a causa del ritardo causato dal campionamento il controllore proporzionale agisce come un tipo PT_1 (come la carica del condensatore) anziché come un "vero" controllore di tipo P.
- L'implementazione del controllore proporzionale in C introduce $\mathbf{max}(u_{\mathrm{min}}, \mathbf{min}(u, u_{\mathrm{max}}))$ ossia una saturazione che rende il controllore (e di conseguenza il sistema) non lineare.
- Il controllore PI presentato è un'approssimazione (approssimazione di Eulero) di un controllo "veramente" integrale, ossia a tempo continuo. Esistono approssimazioni migliori (per esempio la trasformazione bilineare di Tustin).