Università degli studi di Parma
Dipartimento di Ingegneria e Architettura
Sistemi operativi e in tempo reale – a.a. 2023/24

Introduzione ai sistemi in tempo reale

prof. Stefano Caselli
stefano.caselli@unipr.it
http://rimlab.ce.unipr.it

Indice

- *Introduzione ai sistemi in tempo reale*
  - Sistemi di elaborazione operanti con vincoli temporali e sistemi embedded
  - Tipologie dei sistemi in tempo reale e parametri caratteristici
  - Modello di riferimento per i sistemi di elaborazione in tempo reale

Indice

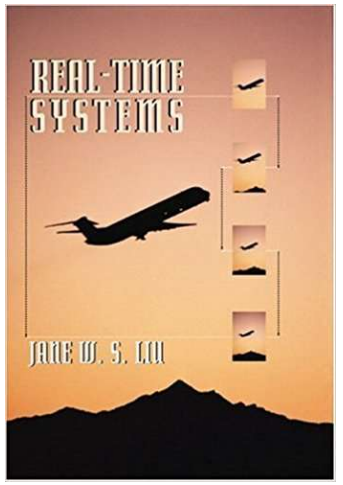
- *Scheduling*
  - Task aperiodici
  - Task periodici
  - Sistemi di task misti (periodici e aperiodici)
  - Algoritmi di scheduling: Rate Monotonic, Erliest Deadline First, ed altri
  - Protocolli di accesso a risorse condivise

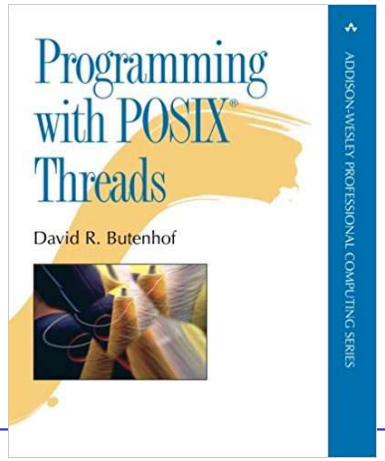
Indice

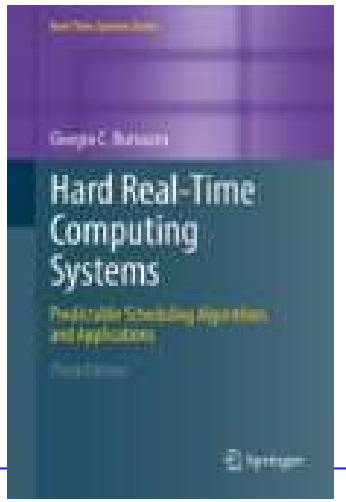
- *Sistemi operativi e programmazione multithread*
  - Funzionalità dei sistemi operativi per l'elaborazione in tempo reale
  - Lo standard POSIX
  - Funzionalità del sistema operativo Linux
  - Programmazione con thread POSIX e C++

Risorse

- J. W.S. Liu, «Real-Time Systems», Prentice-Hall, 2000. (Chapt. 1-8).
- G. Buttazzo, «Hard Real-Time Computing Systems», Springer, 2011.
- D.R. Butenhof, «Programming with POSIX Threads», Addison-Wesley, 1997.





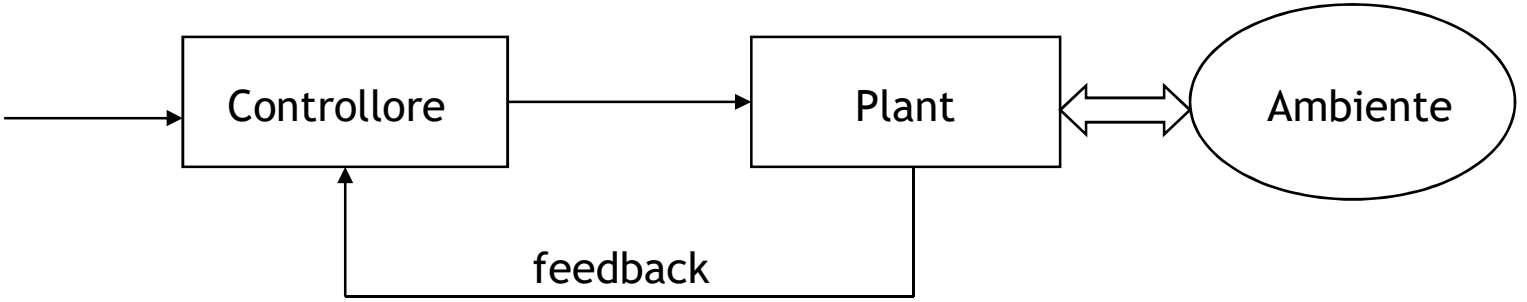


Dove si trovano i sistemi in tempo reale

- In ogni applicazione di controllo di un sistema fisico si possono distinguere tre componenti principali:
  - il *sistema da controllare*
    - talvolta chiamato *plant*
    - eventualmente comprensivo di sensori ed attuatori
  - il *controllore*
    - invia segnali al sistema in base ad obiettivi di controllo predefiniti
  - l'*ambiente* in cui opera il sistema
- I sistemi in tempo reale che interagiscono con sistemi fisici sono talvolta denominati *sistemi ciberfisici* (*cyber-physical systems*, *CPS*), o anche *sistemi embedded (embedded sys., ES)*

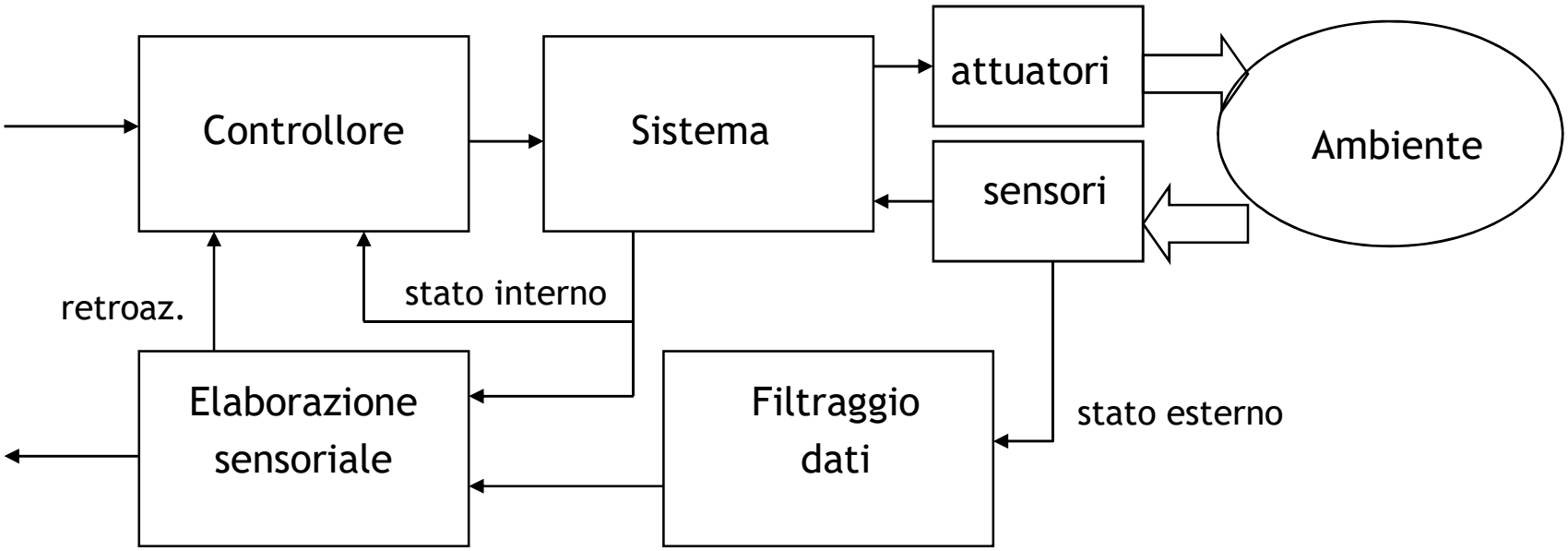
Un tipico sistema di controllo

- Schema a blocchi:



- Quale è il ruolo del sistema in tempo reale?

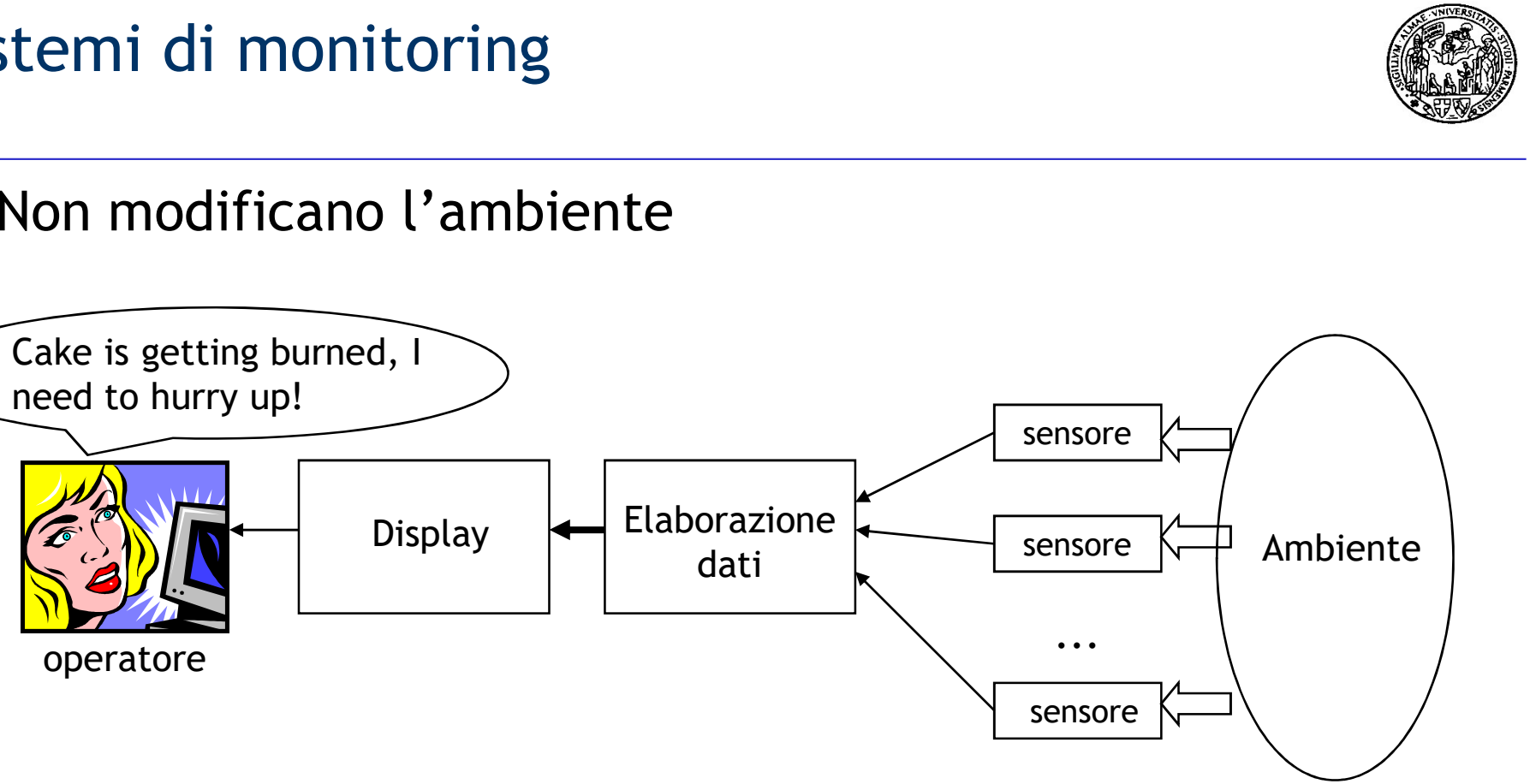
Più in dettaglio



Tipi di sistemi di controllo

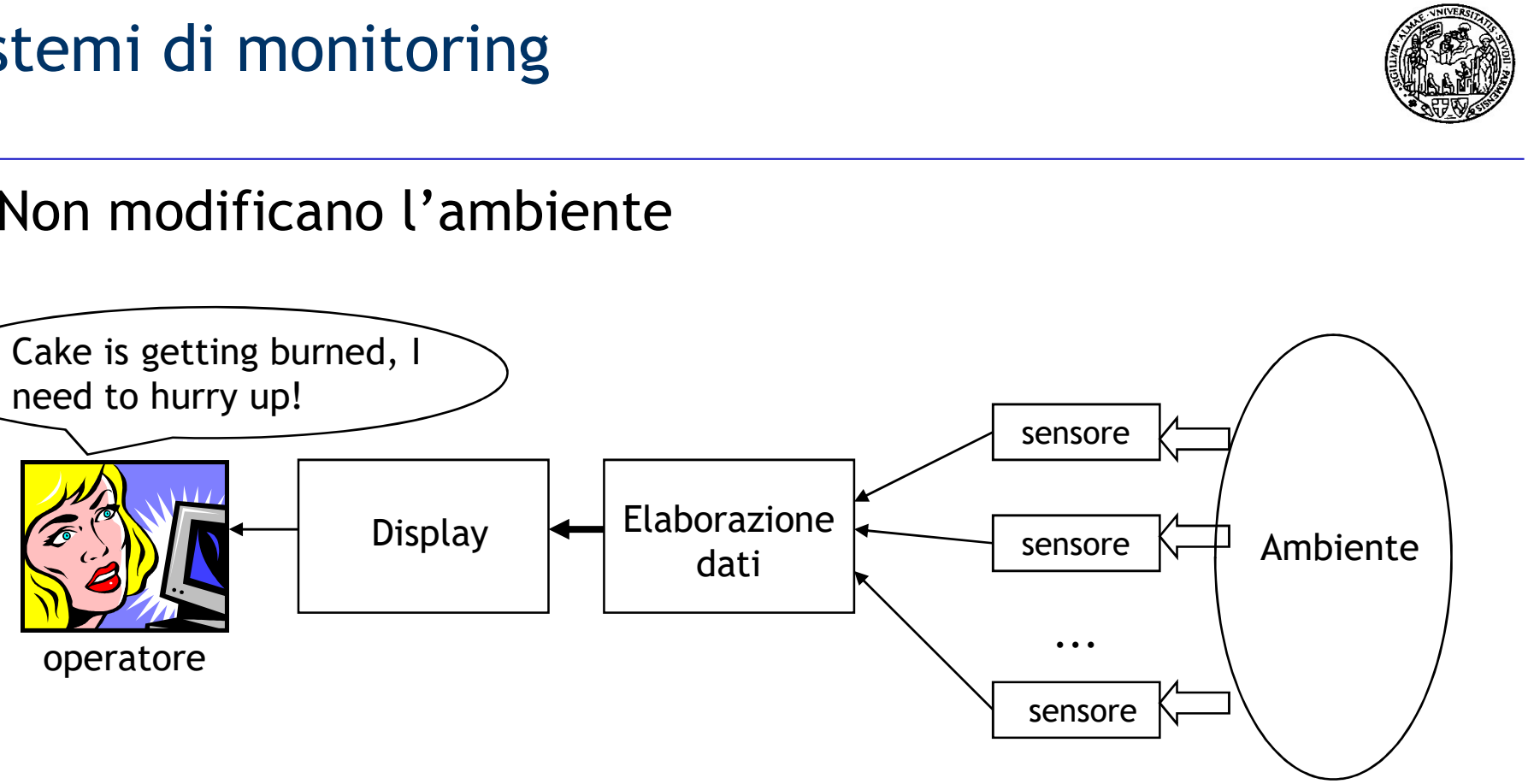
- In base all'interazione sistema-ambiente, si possono distinguere tre tipi di sistemi di controllo:
  - sistemi di monitoring
    - non modificano l'ambiente
  - sistemi di controllo ad anello aperto
    - modificano in modo lasco l'ambiente
  - sistemi di controllo ad anello chiuso
    - interazione stretta tra percezione e azione

Sistemi di monitoring



- Non modificano l'ambiente
- Esempi: sistemi di sorveglianza, traffico aereo, monitoring di impianti industriali, farmaceutica
- Attività supervisionate e spesso normate, con vincoli sui tempi di restituzione delle informazioni all'operatore

Sistemi di monitoring

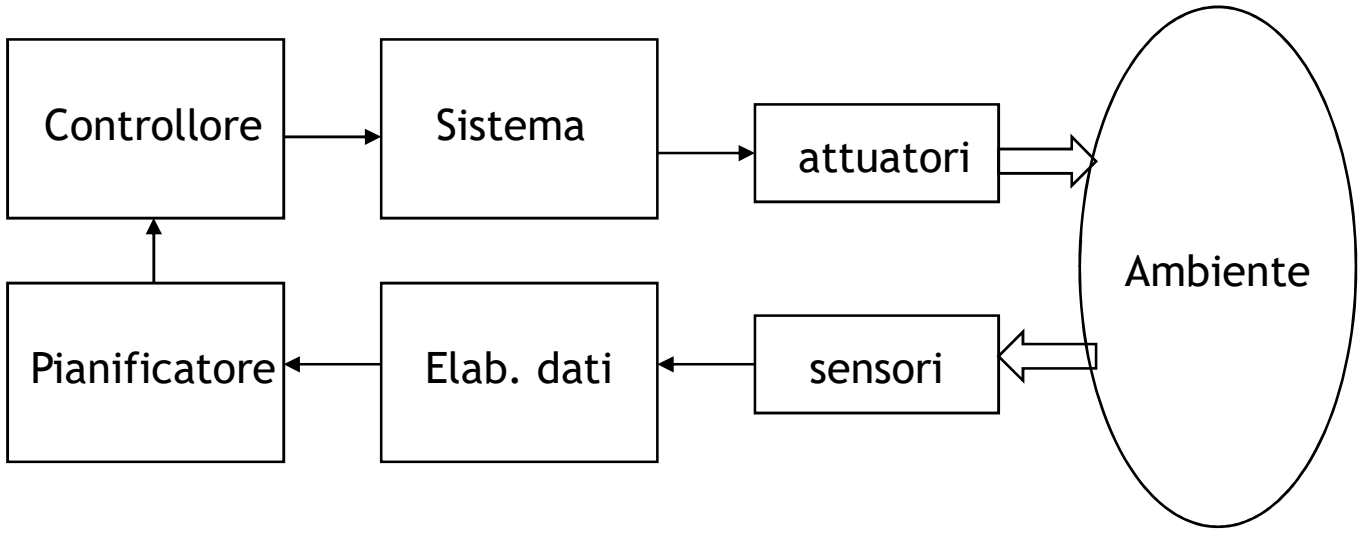


- Non modificano l'ambiente
- Esempi: sistemi di sorveglianza, traffico aereo, monitoring di impianti industriali, farmaceutica
- Attività supervisionate e spesso normate, con vincoli sui tempi di restituzione delle informazioni all'operatore

BAT «Best Available Technologies»

Sistemi di controllo ad anello aperto

- Percezione e controllo accoppiati in modo lasco

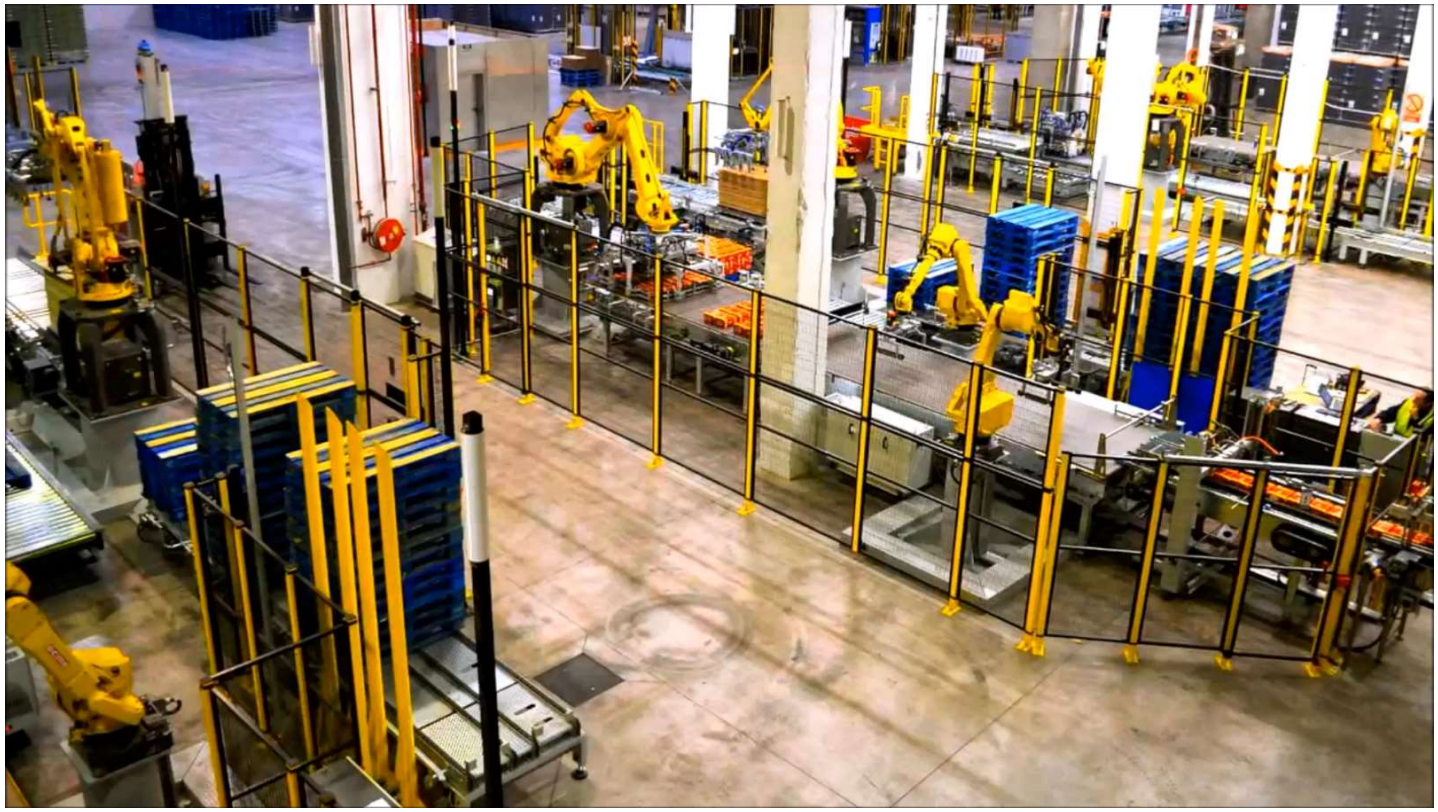


- Esempio: robot in compiti pre-pianificati (montaggio, linee industriali)

(Da una tesi di LMII)

Linea di pallettizzazione con robot – vantaggi:

- ✓RIDUZIONE COSTI
- ✓AUMENTO DEL THROUGHPUT
- ✓ELEVATI STANDARD QUALITATIVI
- ✓MIGLIORAMENTO DEL PROCESSO PRODUTTIVO
- ✓CONTROLLO DELLA LINEA PRODUTTIVA
- ✓RIDUZIONE DEL CARICO DI LAVORO PER OPERATORI
- ✓ACCURATEZZA E PRECISIONE OPERAZIONI
- ✓UNIFORMITÀ OUTPUT
- ✓AGEVOLE CAMBIO FORMATO PRODOTTI

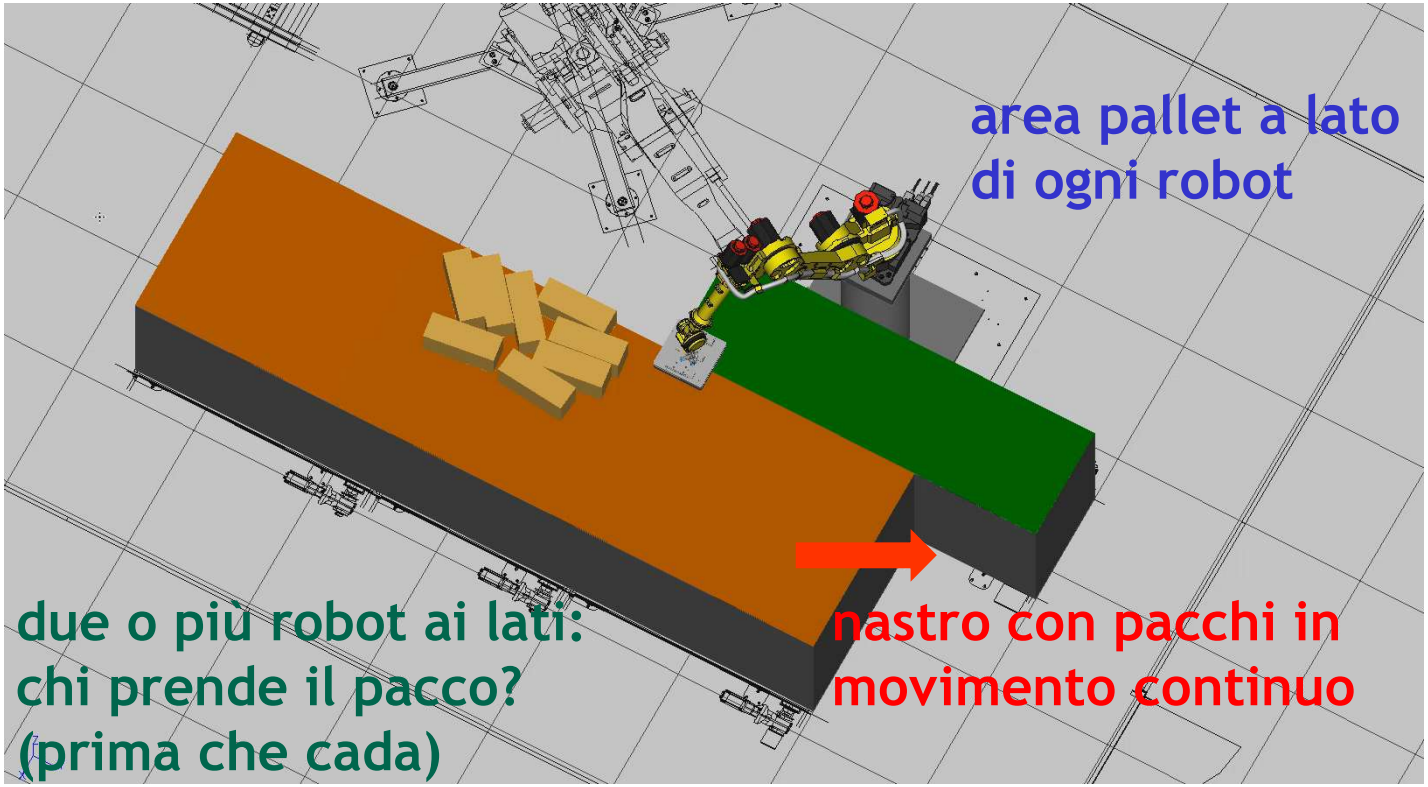


Linea OCME srl

(Da una tesi di LMII)

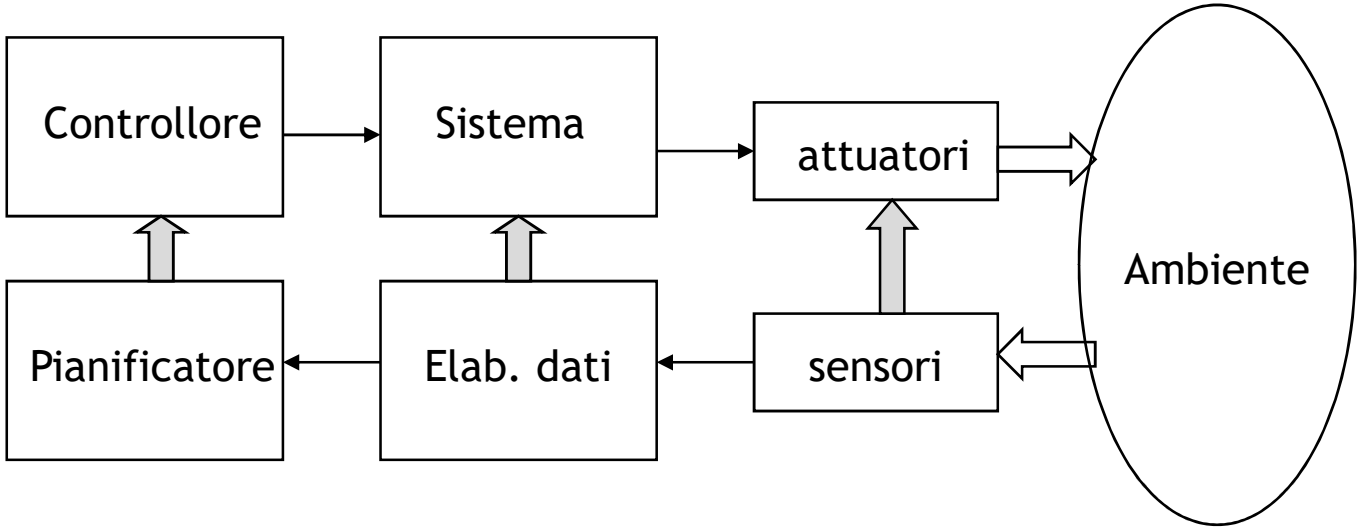
Linea multirobot per smistamento pacchi:

- ✓SIMULAZIONE PROGRAMMA DI CONTROLLO CON ROBOGUIDE FANUC
- ✓CONFIGURAZIONE PACCHI RILEVATA MEDIANTE SISTEMA DI VISIONE
- ✓PIANIF. MOVIMENTI PER EVITARE URTI
- ✓QUALE PACCO? TUTTI I PACCHI DEVONO ESSERE PRELEVATI
- ✓PIU' MANIPOLATORI LUNGO LA LINEA
- ✓ELABORAZIONE INIZIALE ASSEGNA PACCHI A CIASCUN MANIPOLATORE



Sistemi di controllo ad anello chiuso

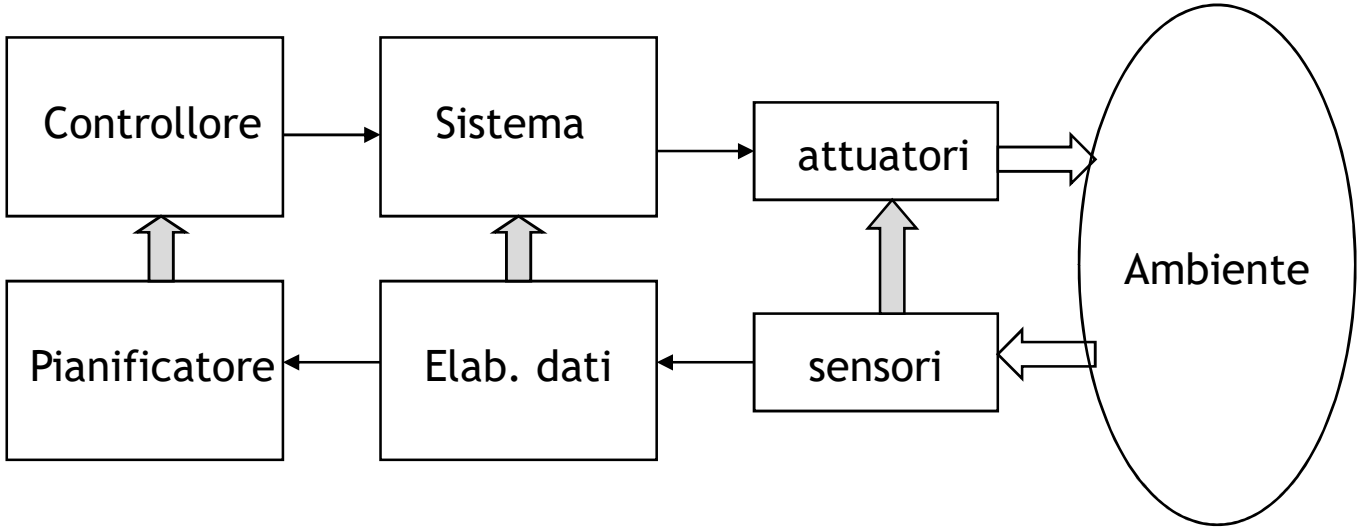
- Percezione e controllo strettamente accoppiati



- Esempi: sistemi militari ed industriali, controllori aerei, robot, sistemi biologici, vita nel mondo reale fisico

Sistemi di controllo ad anello chiuso

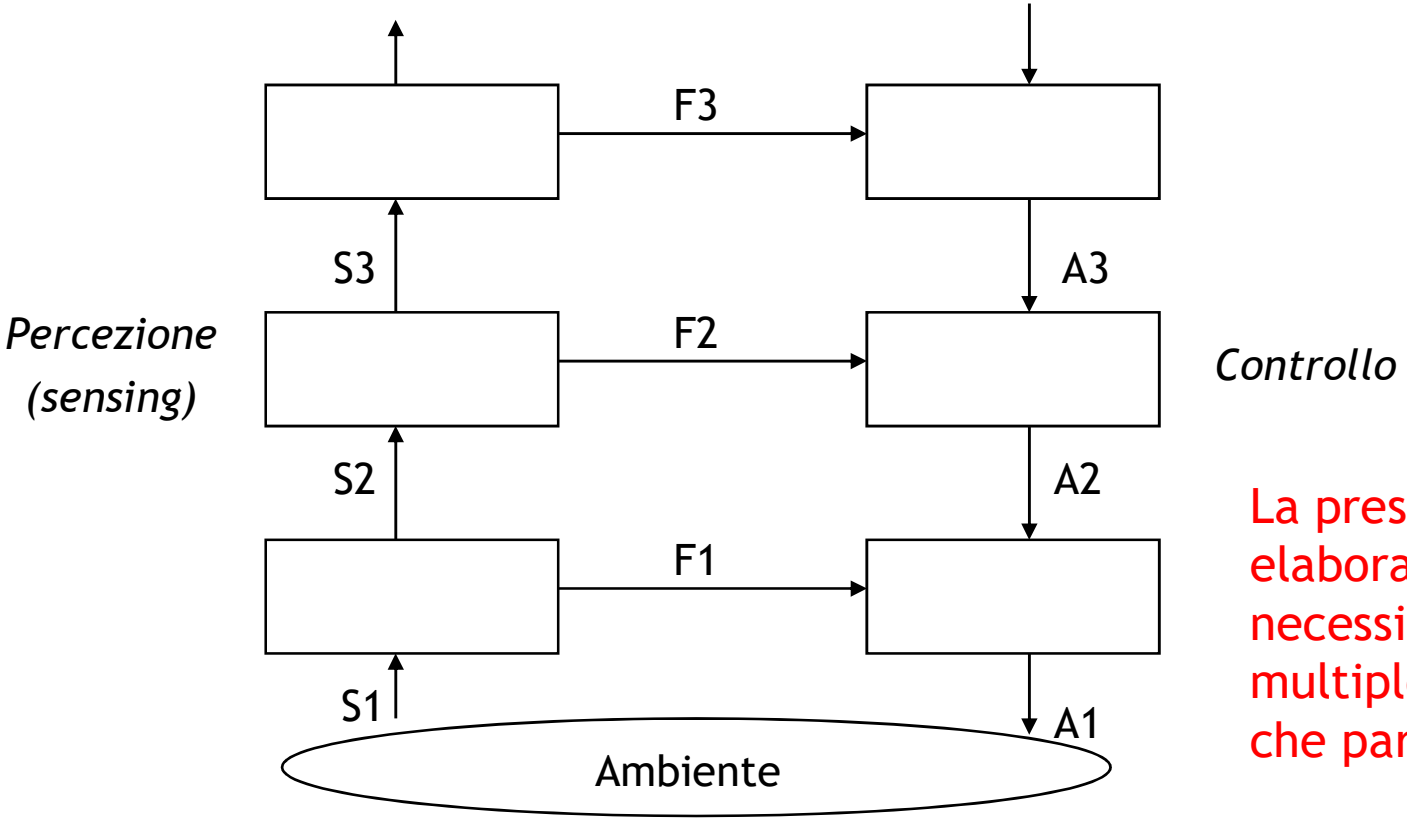
- Percezione e controllo strettamente accoppiati



La robotica di servizio deve essere *collaborativa* ed è soggetta a normative stringenti

- Esempi: sistemi militari ed industriali, controllori aerei, robot, sistemi biologici, vita nel mondo reale fisico

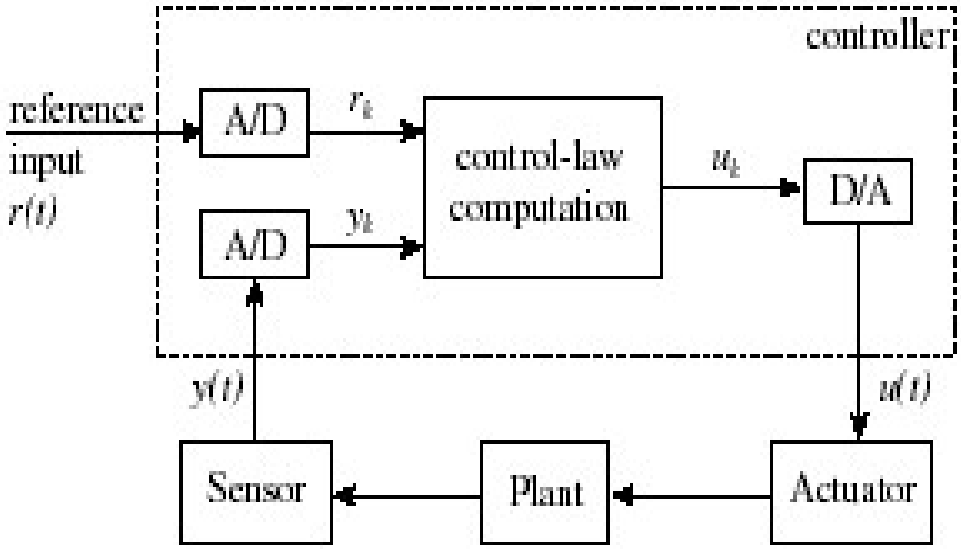
Sistemi di controllo con retroazione multilivello



La presenza di più livelli di elaborazione determina la necessità di scale temporali multiple per i diversi task che partecipano al controllo

Controllo digitale

- Sistemi a tempo campionato



- Legge di controllo (ad es. per controllo PID) del tipo:

$$u(k)=u(k-1)+ae(k)+be(k-1)+ce(k-2)$$

Controllo digitale

- Anello di controllo in retroazione:
  - inizializza timer per interruzione periodica con periodo T;
  - ad ogni interruzione da timer do:
    - conversione A/D di *y*;
    - lettura o conversione A/D di *r*;
    - calcolo del segnale di controllo *u*;
    - attuazione di *u* ed esecuzione di conversione D/A;

    end do;
- Ipotizziamo che il sistema renda disponibile un *timer*, che una volta configurato generi un'interruzione ogni T unità di tempo

Periodo di campionamento

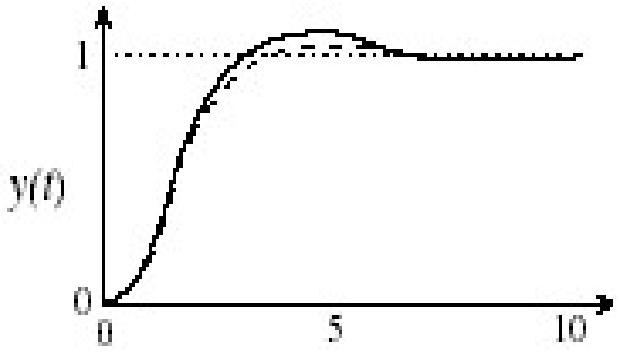
- Il periodo di campionamento T è un parametro di progetto importante
- I coefficienti della legge di controllo dipendono dal periodo di campionamento
- Ad es., per l'anello di controllo PID la derivata di e(t) si calcola da differenze finite (e(k)-e(k-1))/T, l'integrale con la regola trapezoidale, etc.
- Un periodo "piccolo" approssima meglio la legge di controllo analogica, ma produce un maggior carico computazionale → *tradeoff*

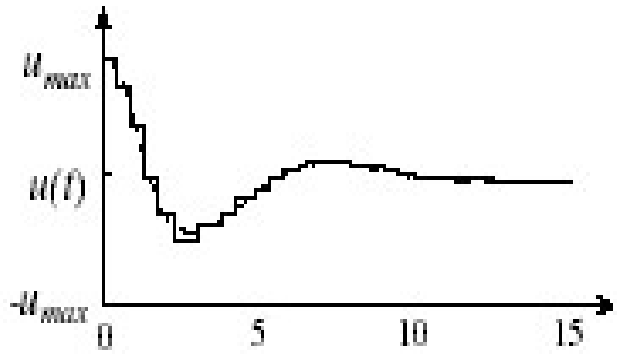
Scelta del periodo di campionamento

- Prontezza percepita del sistema complessivo (plant+controllore):
  - Se il sistema è gestito da un operatore, un nuovo comando può essere gestito dal controllore con un ritardo max di T
  - $T \leq 100ms$ per ogni ingresso manuale

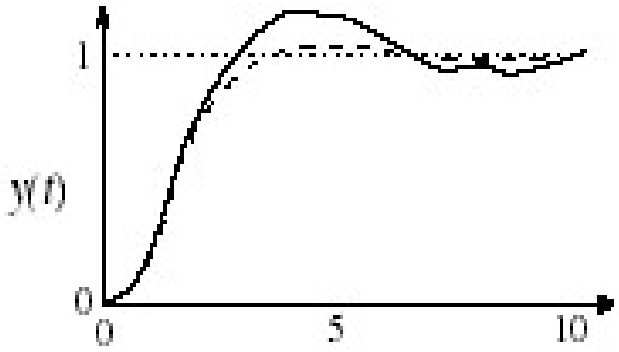
- Dinamica del sistema:
  - occorre garantire sia una risposta corretta da parte del sistema sia la sua stabilità
  - il parametro di riferimento è R/T, ove R è il tempo di salita della risposta a gradino

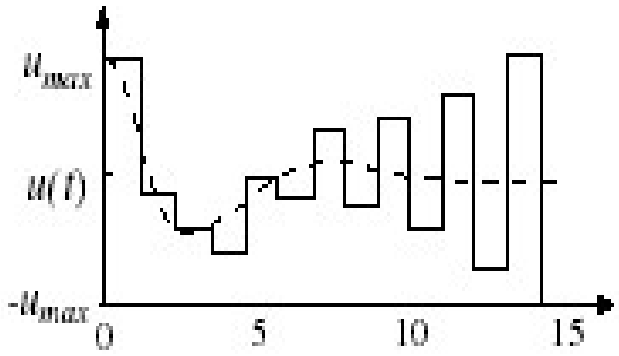
Effetti del periodo di campionamento





R/T=8





R/T=4

Periodo di campionamento

- *Rules of thumb* verificate in casi specifici ...

- Valori consigliati per R/T nel range 10-20
- R/T ≈ 20 assicura una risposta molto vicina a quella del corrispondente sistema con controllo analogico
- R/T ≈ 10 dà luogo ad un limitato degrado della risposta
- R/T = 4 è spesso il minimo valore che dà luogo ad una risposta accettabile, stabilità borderline

Periodo di campionamento

- R/T molto elevati, ad es. >> 20, determinano una eccessiva influenza dell'errore di quantizzazione nell'azione di controllo (lungh. parola finita)

- In base al teor. di Shannon, $1/T \geq 2B$, ove B è la banda lorda, $B=1/2R$
  - poco restrittivo ma potrebbe richiedere valori del segnale di controllo u(k) eccessivi o non realizzabili
  - in pratica, si avrebbe instabilità

Sistemi ciberfisici con più variabili controllate

- Un impianto complesso ha tipicamente *più variabili controllate*, con *caratteristiche dinamiche molto diverse* (ad es.: velocità motore e temperatura)
- Controllare tutte le variabili di un impianto complesso alla frequenza imposta dalla dinamica più veloce non è fattibile per diversi motivi e implicherebbe uno spreco di risorse → sistemi *multirate* (anelli di controllo a frequenze diverse)
- Per variabili correlate si possono spesso usare insiemi di frequenze armoniche (semplicità, efficienza)

Esempio: Sistema di controllo dell'assetto di volo per un elicottero

- Ciclo principale a 180 Hz
- Cicli minori a 90 Hz e 30 Hz
- I comandi del pilota sono letti in un ciclo a 30 Hz

- (dettagli a pag. 6-7 del libro di Jane Liu)

- E' *una parte* del sistema di controllo complessivo di un elicottero
- Le altre parti, non riferite all'assetto di volo, non sono vincolate ad eseguire con frequenze armoniche

Sistemi di controllo più complessi

- Ulteriori problemi di progetto dei sistemi in tempo reale:

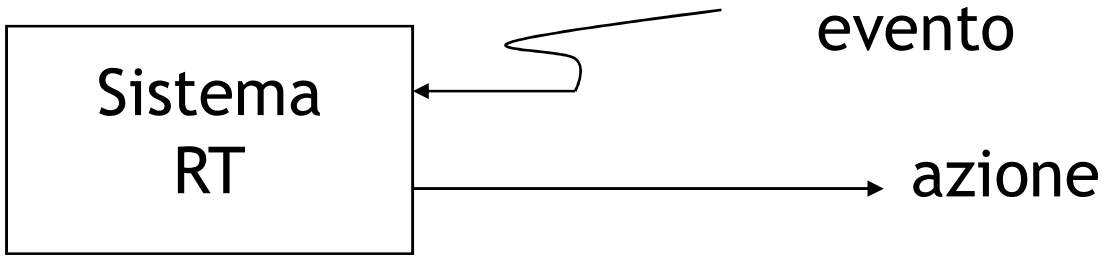
- Algortimi ad elevata complessità intrinseca da eseguire con vincoli temporali (filtraggio stocastico, filtri particellari e di Kalman, stima online in segnali affetti da rumore, etc.) → *complessità intrinseca*
- Anelli di controllo che incorporano ad ogni passo elaborazioni ad alta varianza (pianificatore, ricerca di target in immagine, etc.) → *tempo di esecuzione non deterministico*
- Gruppi di variabili tra loro non correlate → *difficoltà ad impostare le frequenze in modo armonico*

Vincoli e scadenze temporali

- L'interazione stretta con l'ambiente richiede reazioni agli eventi, da parte del sistema, entro precise *scadenze temporali*
- Le scadenze temporali sono imposte dalla dinamica dell'ambiente

➔Il sistema operativo deve essere in grado di eseguire task rispettando scadenze e vincoli temporali

Sistemi in tempo reale

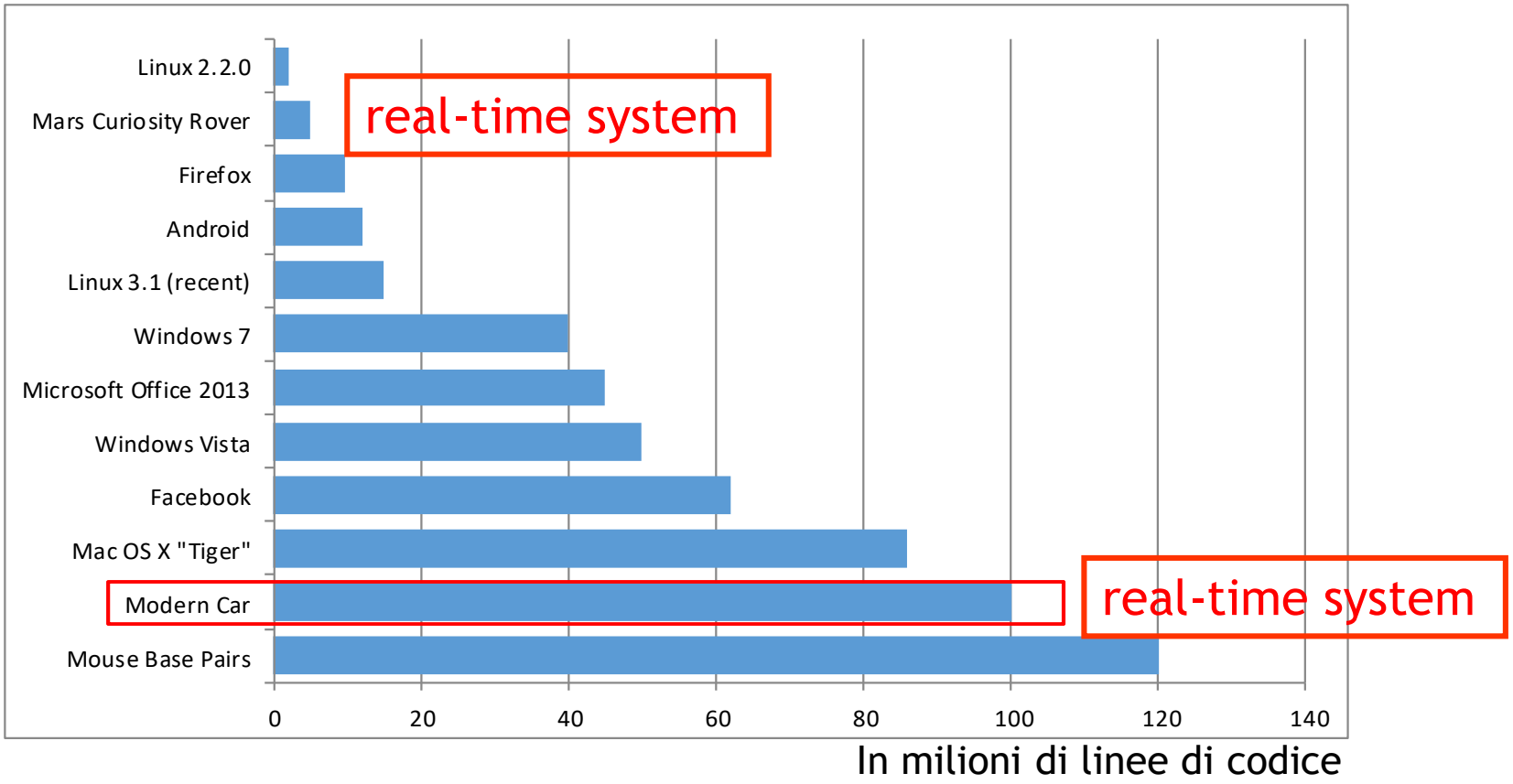


- Un sistema di elaborazione in grado di rispondere ad eventi rispettando precisi vincoli temporali è un *Sistema in Tempo Reale*

Applicazioni real-time

- controllo di impianti nucleari e chimici
- robotica
- automotive, *x*-by-wire
- controllo assetto aereo in volo, atterraggio, ...
- sistemi medicali
- gestione impianti ferroviari
- monitoraggio e controllo di traffico aereo
- sistemi di telecomunicazione
- multimedia
- alas, militari ...

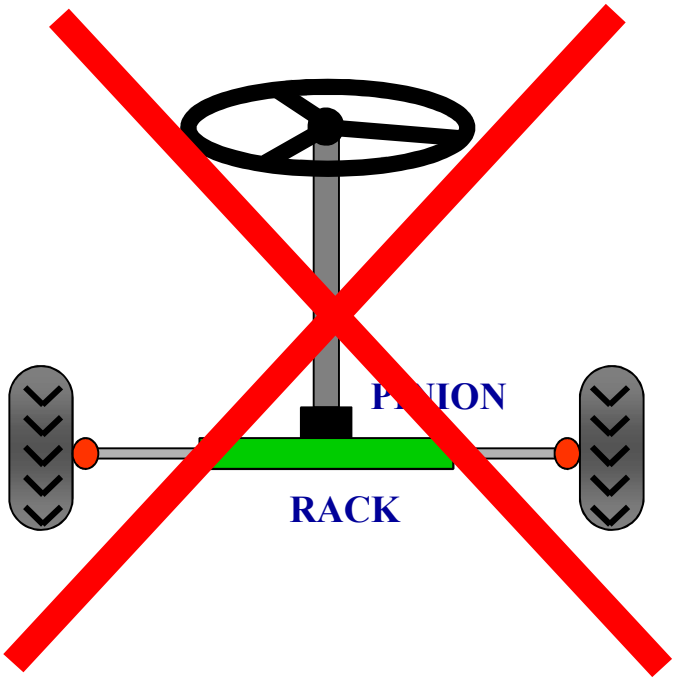
La crescente complessità del software



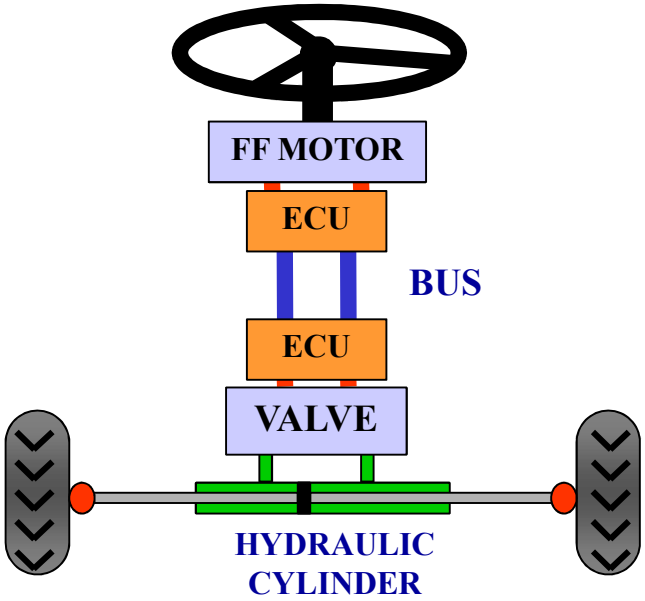
- https://informationisbeautiful.net/visualizations/million-lines-of-code/
- In un'auto moderna: circa 100M linee di codice e 20+ CPU

Steer by wire

Meccanico



Meccatronico



Meglio rispettare le deadline …

Sistemi embedded

- I sistemi di elaborazione RT sono spesso nascosti, integrati in altri apparati
- Il corretto funzionamento del sistema complessivo può dipendere strettamente dalla tempestività della elaborazione (*embedded real-time systems*)

Sistemi RT critici

- I malfunzionamenti, in alcuni dei sistemi RT elencati, possono avere conseguenze importanti:
  - sistemi *mission critical* (integrità applicazione: *money at stake*)
  - sistemi *safety critical* (integrità persone: *people at stake*)

- I sistemi RT critici:
  - devono funzionare correttamente e reagire in modo pronto
  - necessitano di *garanzie formali* di correttezza e di tempo di risposta
- *Validazione* --> dimostrazione rigorosa del comportamento temporale del sistema

Esempio

- Il controllo di un braccio robotico:
  - livello “servo”
  - livello interpolatore delle traiettorie
  - livello “move”
  - livello pianificatore del compito

- Scale temporali diverse per la gestione degli eventi ai diversi livelli

Esempio

- Il controllo di un robot mobile in un ambiente domestico:
  - ancora articolato su diversi livelli (servo, traiettorie, move, task)
  - qui però l'ambiente è *molto* dinamico

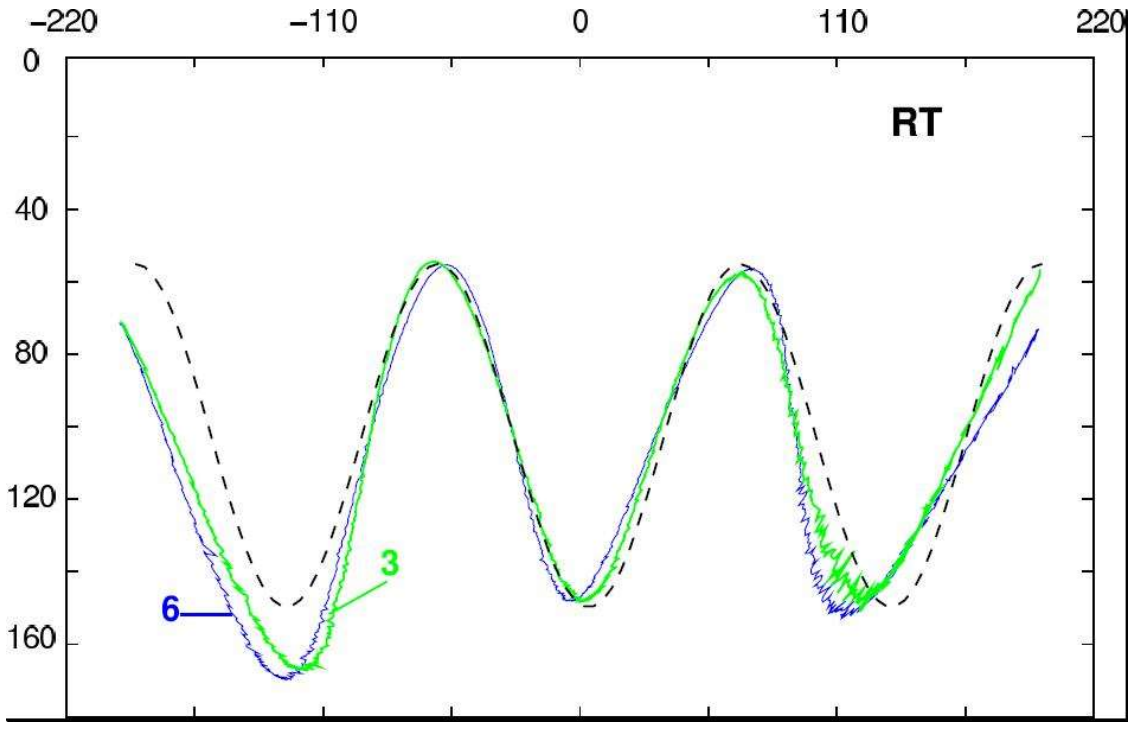
Robot Nomad200, RIMLab,
Circa 1996-2002

- Q: Cosa succede in questi sistemi se non si reagisce *in tempo utile* agli eventi?

Figure 1: The Robot RHINO in the 'Deutsches Museum Bonn'.

Q: Cosa succede …

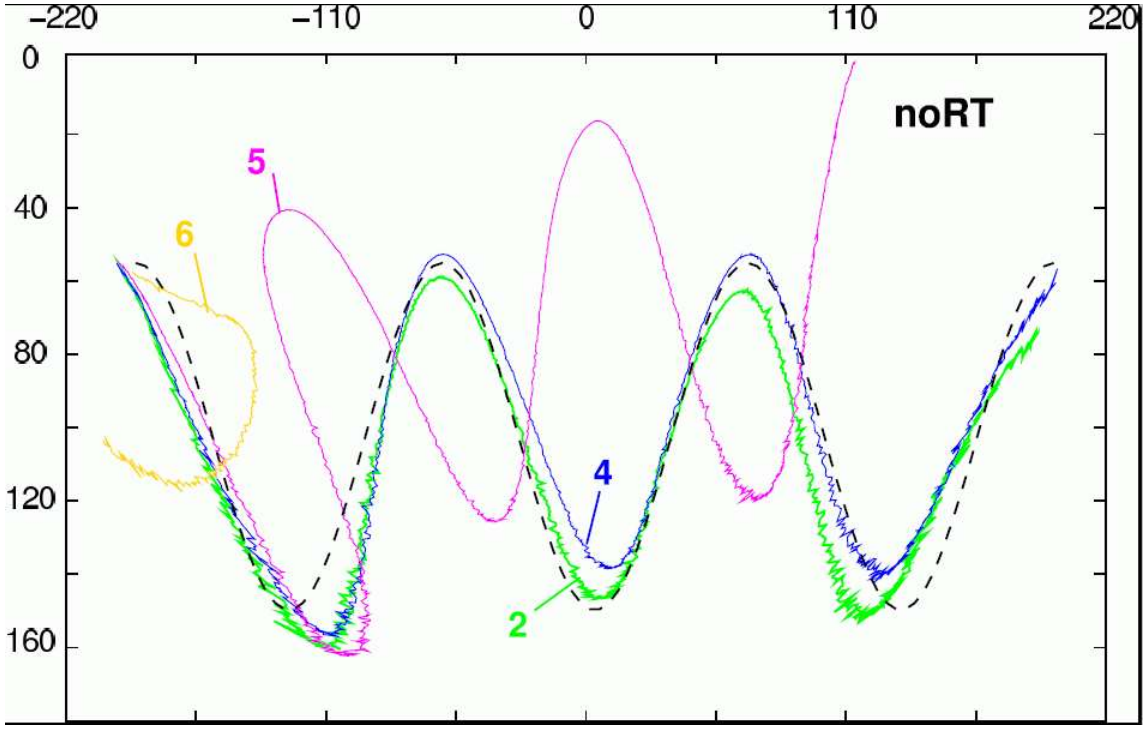
- Esecuzione di una traiettoria curvilinea, al variare del carico computazionale



Con supporto di esecuzione RT

Q: Cosa succede …

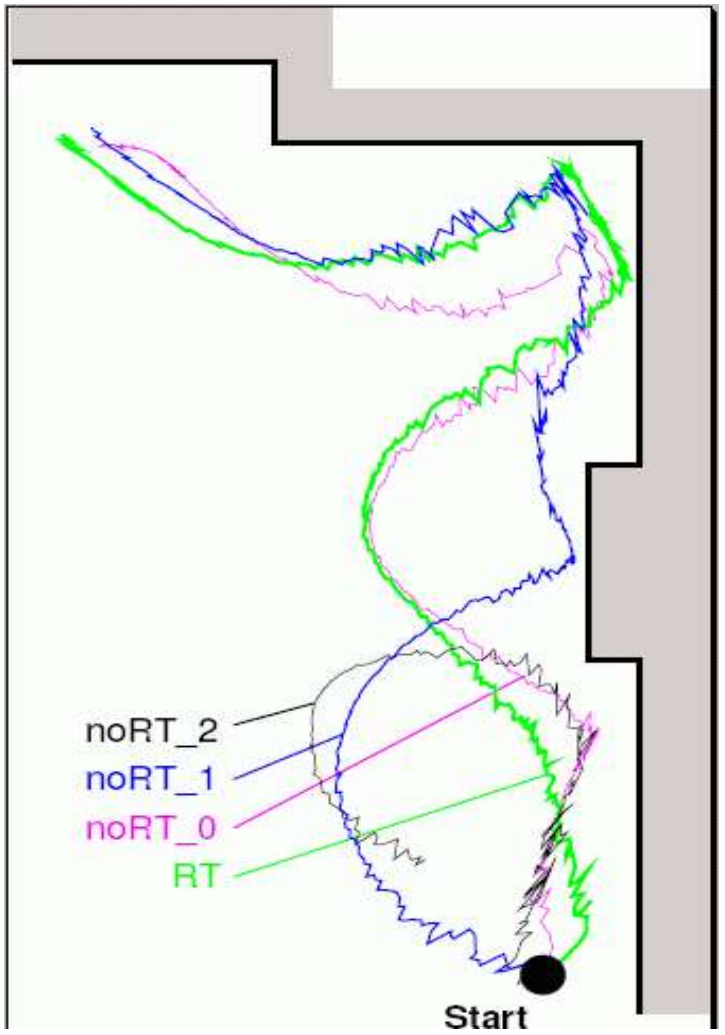
- Senza supporto di esecuzione RT:



➔ Comportamenti impredicibili e potenzialmente pericolosi

Assenza di supporto per elaborazione RT

- Effetti in un compito che prevede interazioni sensomotorie:
- evidente *degrado delle prestazioni* in assenza di supporto RT
- *comportamento non prevedibile* in presenza di un carico computazionale elevato o variabile



Approccio empirico

- Molte applicazioni RT, ampiamente diffuse, sono progettate secondo tecniche o con soluzioni empiriche:
  - programmazione assembly
  - temporizzazione con timer hardware dedicati
  - programmazione di driver di basso livello
  - modifica delle priorità in modo ad hoc
- Tutto utile, ma risolutivo solo nei casi più semplici!

Problemi con approcci empirici

- Programmazione difficoltosa, efficacia basata fortemente sulle capacità del programmatore
- Codice poco comprensibile
- Scarsa manutenibilità del codice
- Difficile verifica del rispetto dei vincoli temporali

⇨ scarsa affidabilità

Alcune indicazioni

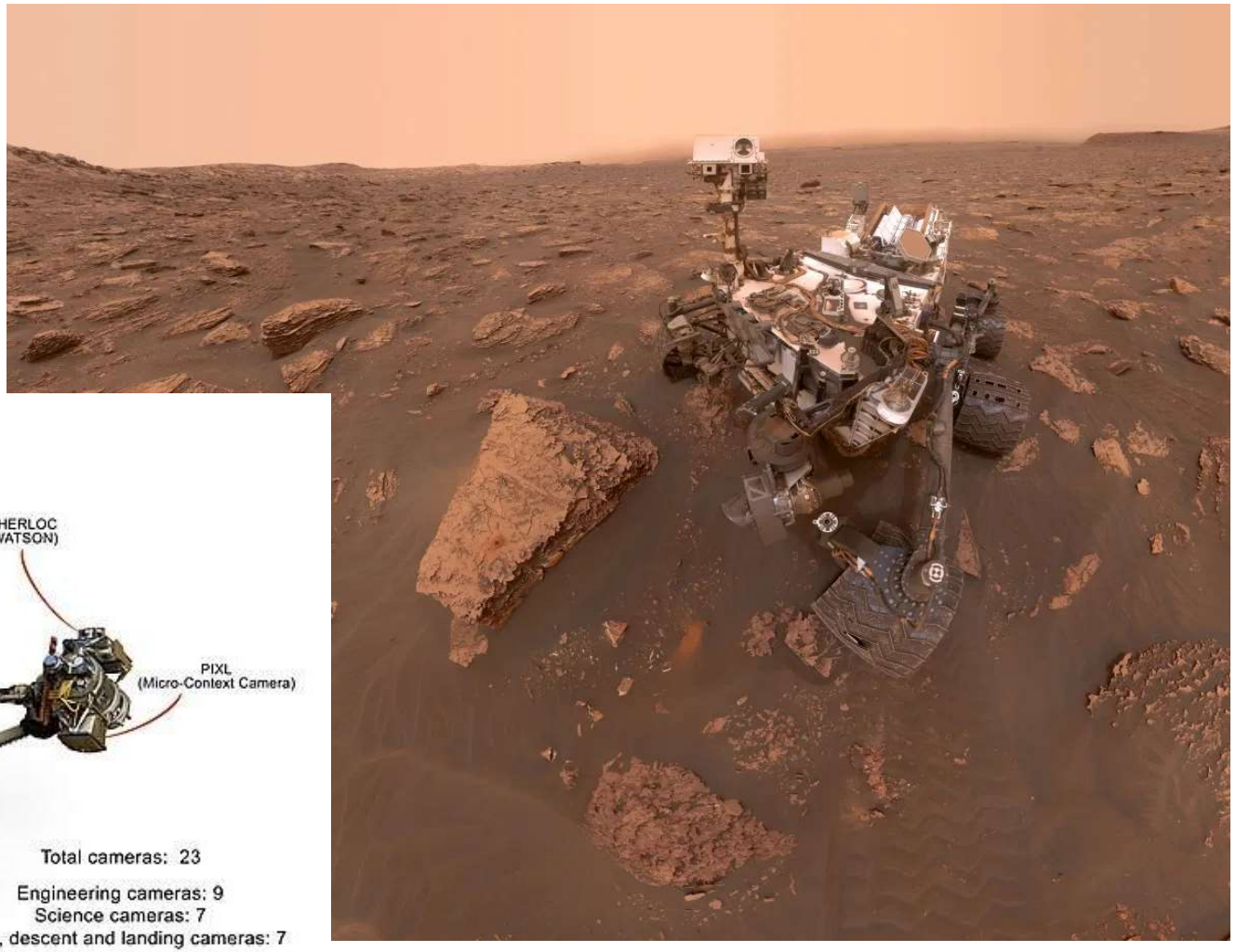
- I *collaudi* («testing»), pur necessari, permettono solo una verifica *parziale* del comportamento di un sistema; non ne *garantiscono* la correttezza! (Butler e Finelli, 1993)
- La *predicibilità* al livello del supporto del sistema operativo (kernel) ha un ruolo importante nei sistemi in tempo reale
- E' necessario gestire le situazioni di *sovraccarico* ed integrare meccanismi di *tolleranza ai guasti*
- I sistemi critici devono essere progettati adottando *ipotesi pessimistiche* (scenari «worst case»)

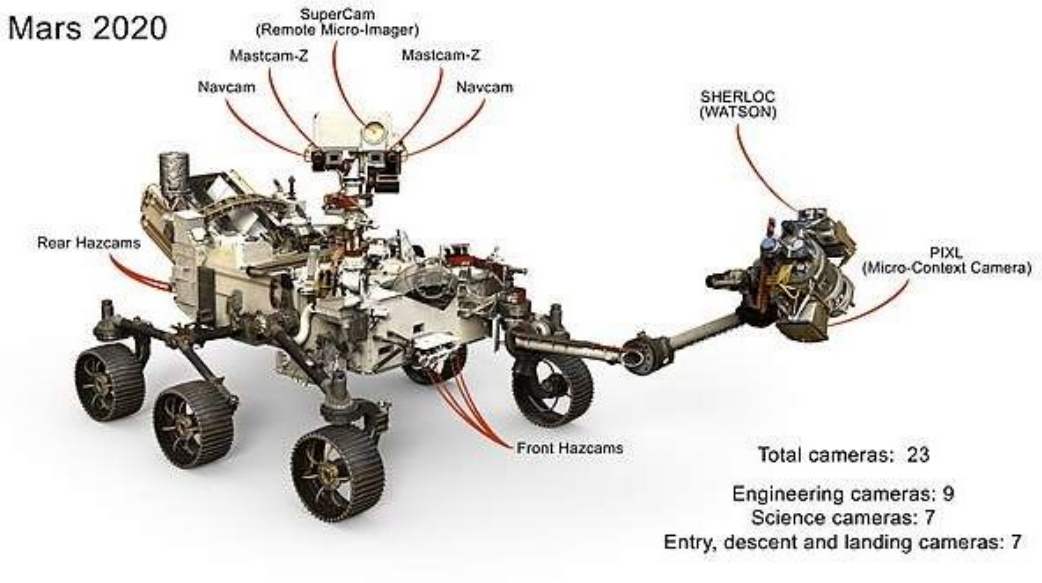
Programmazione concorrente *e* in tempo reale

- Le applicazioni real-time significative richiedono la presenza di task (real-time) interagenti
- → architettura multitask, tipicamente con interazione in ambiente globale (perché?)
- → la programmazione concorrente è alla base della programmazione in tempo reale …

Perseverance, 2021

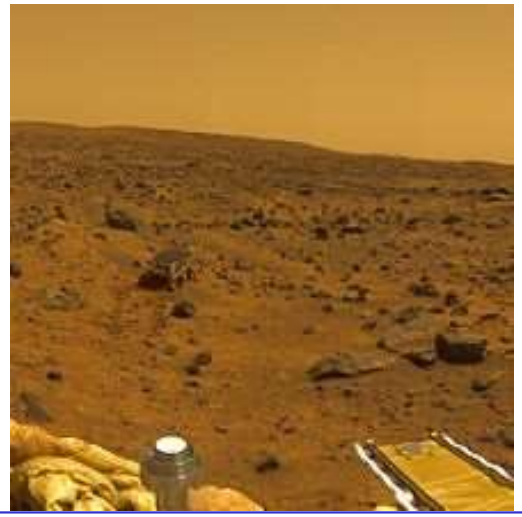
- Su Marte dal 18 Febbraio
- Un concentrato di tecnologia
- Come funziona?





Sojourner on Mars – 1997: una storia robotica *e* real-time

- Sojourner è il rover scaricato su Marte nel 1997 dalla navetta Pathfinder:
- 20MHz CPU, 128MB DRAM, VxWorks OS
- telecamere, strumenti scientifici, batterie, solare, attuazione, comunicazione
- sistema multithread



Reset? Reboot?

Wrap-up

- I sistemi che devono operare in tempo reale sono pervasivi
- L'elaborazione che li guida può richiedere molteplici attività periodiche, non periodiche, attività di durata impredicibile, interazione con operatori
- Complessità crescente
- Il rispetto di vincoli temporali è cruciale per la sicurezza e l'integrità dei sistemi
- Come *certificare* un veicolo autonomo? Un impianto industriale?
- Spesso è necessario dare *garanzie formali*!
  - Robot collaborativi, impianti farmaceutici, norme BAT, ...