

# **UNITÀ DI GOVERNO**

**Architettura funzionale**

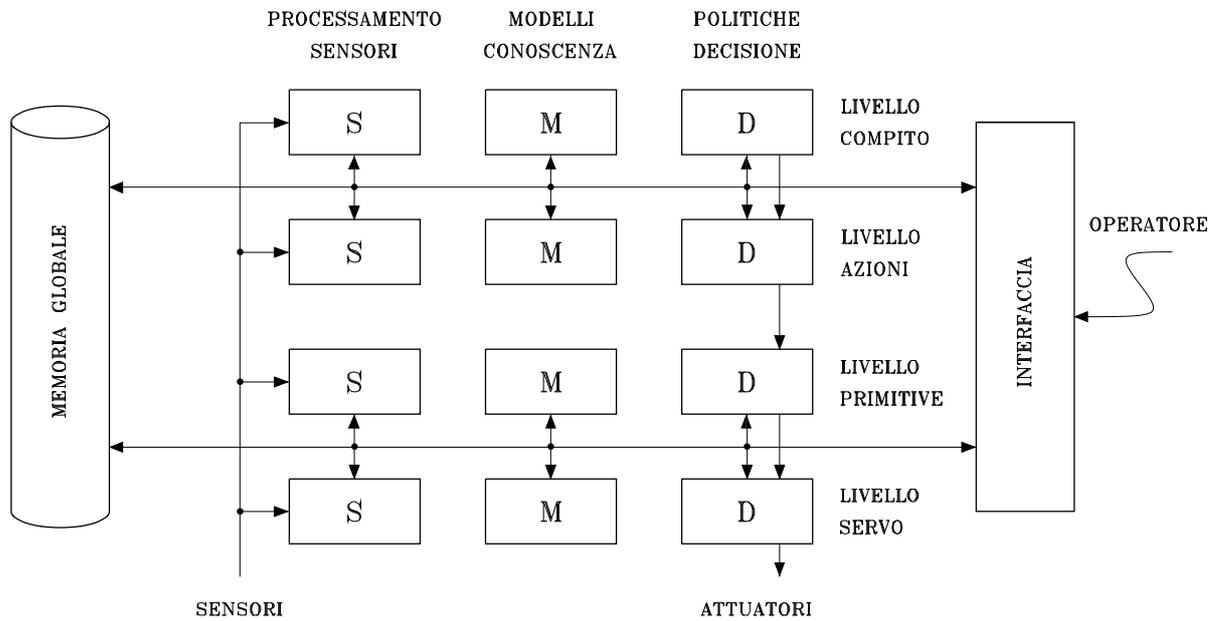
**Ambiente di programmazione**

**Architettura hardware**

## ARCHITETTURA FUNZIONALE

- Unità di governo
  - ★ abilità di muovere oggetti fisici nell'ambiente di lavoro, (capacità di *manipolazione*)
  - ★ abilità di ottenere informazioni sullo stato del sistema e su quello dell'ambiente di lavoro (capacità *sensoriali*)
  - ★ abilità di utilizzare informazioni per modificare il comportamento del sistema in maniera preprogrammata (capacità di comportamento *intelligente*)
  - ★ abilità di immagazzinare, elaborare e rendere disponibili dati relativi all'attività del sistema (capacità di *processamento dati*)

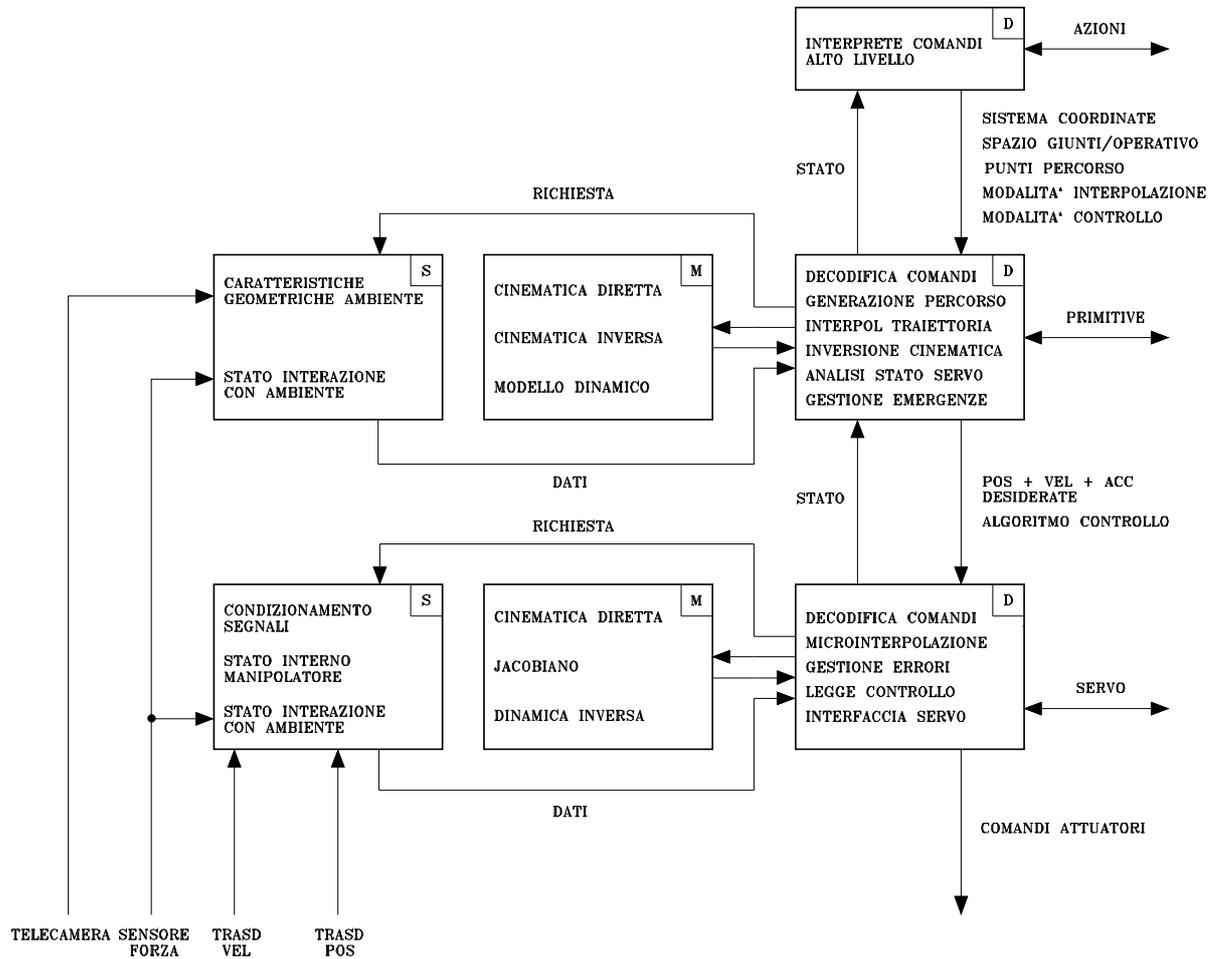
## Modello di riferimento



- Evoluzione verso livelli superiori
  - ★ compiti complessi
  - ★ missioni

- Livello del *compito*
  - ★ specifica del compito a un livello di astrazione elevato
  - ★ decomposizione in una serie di azioni (coordinamento spazio-temporale)
  - ★ scelta sulla base dei modelli della conoscenza e delle informazioni sulla scena
  
- Livello delle *azioni*
  - ★ comandi simbolici dal livello del compito tradotti in percorsi di moto (azioni elementari)
  - ★ scelta sulla base dei modelli del manipolatore e delle informazioni sull'ambiente
  
- Livello delle *primitive*
  - ★ sequenza di configurazioni dal livello delle azioni
  - ★ calcolo traiettorie di moto (riferimenti per i servo)
  - ★ definizione della strategia di controllo
  - ★ scelta sulla base del modello dinamico e delle informazioni dei sensori
  
- Livello dei *servo*
  - ★ direttive dal livello delle primitive
  - ★ calcolo algoritmi di controllo (segnali di attuazione)
  - ★ microinterpolazione
  - ★ aggiornamento sulla base delle misure dei sensori

# Architettura funzionale per robot industriali



## **AMBIENTE DI PROGRAMMAZIONE**

- Caratteristiche (comuni con quelli dei calcolatori + ...)
  - ★ sistema operativo in tempo reale
  - ★ modellazione del mondo
  - ★ controllo del moto
  - ★ lettura dei dati sensoriali
  - ★ interattività con il sistema fisico
  - ★ capacità di rivelazione di errori
  - ★ capacità di ripristino di situazioni operative corrette
  - ★ struttura specifica dei linguaggi

## Requisiti imposti dall'articolazione in moduli

- **Trattamento dei dati sensoriali**
  - ★ servo (condizionamento segnali in tempo reale)
  - ★ primitive (dati sensoriali in terne di riferimento)
  - ★ azioni (elaborazione dati ad alto livello)
  - ★ compito (riconoscimento oggetti nella scena)
  
- **Consultazione dei modelli di conoscenza**
  - ★ servo (calcolo numerico dei modelli in tempi brevi)
  - ★ primitive (trasformazioni di coordinate)
  - ★ azioni (simulazione + rappresentazione CAD)
  - ★ compito (sistema esperto)
  
- **Funzioni decisionali**
  - ★ servo (calcolo in tempo reale)
  - ★ primitive (condizionamento logico)
  - ★ azioni (sincronizzazione + sistema di interruzioni)
  - ★ compito (gestione di processi concorrenti + debugger)

## **Generazioni di ambienti di programmazione**

Programmazione a basso livello (Assembly, C)

- Programmazione per insegnamento
  - Programmazione orientata al robot
  - Programmazione orientata agli oggetti
- ★ la generazione successiva incorpora le caratteristiche della generazione precedente

## Programmazione per insegnamento

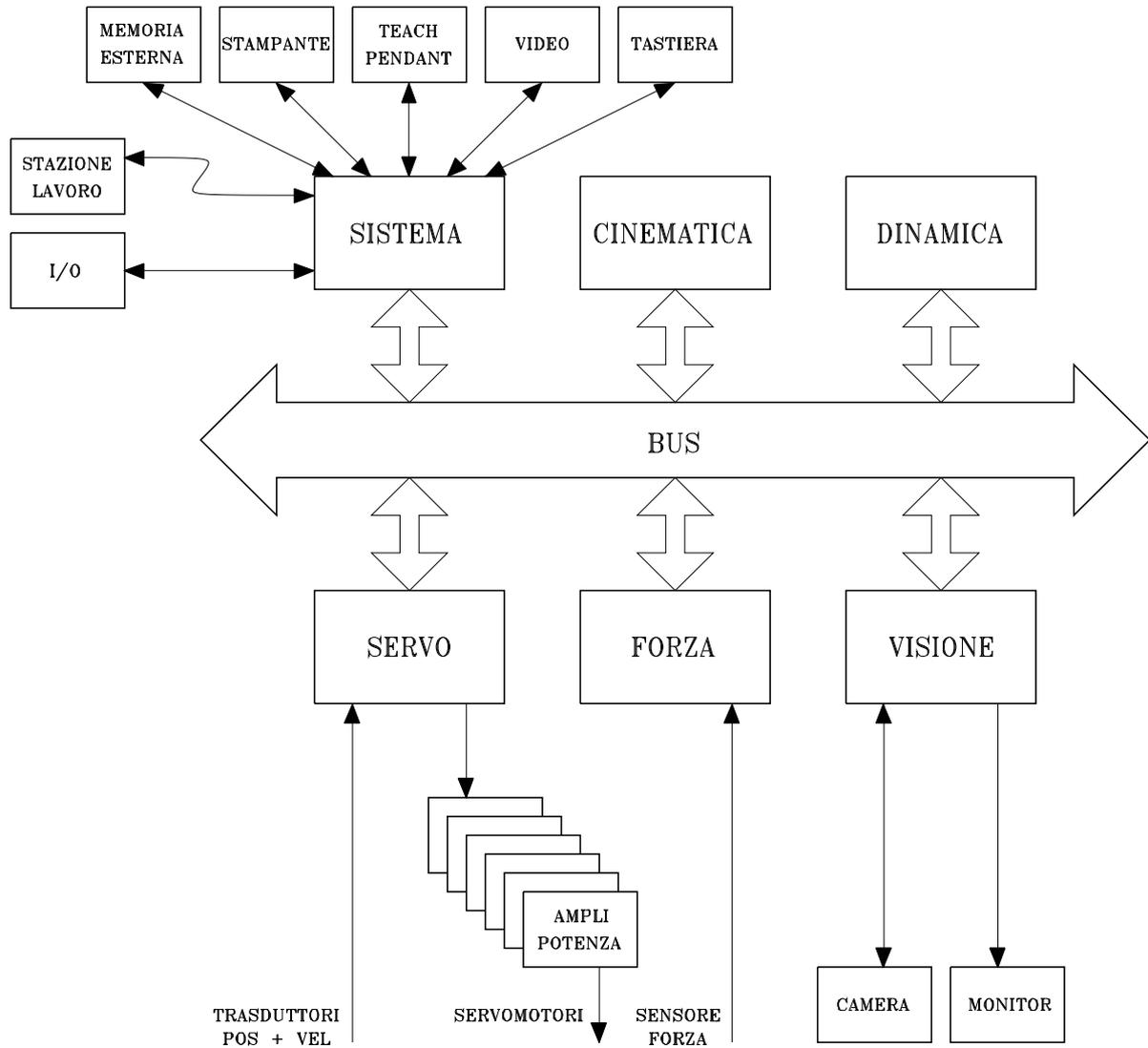
- Guida dell'operatore lungo il percorso desiderato attraverso teach pendant
- Registrazione delle posizioni dei giunti
- Ripetizione del moto
  - ★ capacità di programmazione limitata (tecnico di officina)
  - ★ disponibilità del robot
  - ★ saldatura a punti, verniciatura, palettizzazione
- Adozione di linguaggi
  - ★ acquisizione di posture significative mediante apprendimento
  - ★ calcolo della locazione dell'organo terminale rispetto a una terna di riferimento (cinematica diretta)
  - ★ assegnazione di primitive di moto con parametri di traiettoria
  - ★ calcolo dei riferimenti dei servo (cinematica inversa)
  - ★ condizionamento delle sequenze di apprendimento a semplici sensori (presenza oggetto nella pinza)
  - ★ correzione delle sequenze di moto a mezzo di semplici editori di testo
  - ★ semplici operazioni di collegamento tra sottoinsiemi di sequenze elementari

## Programmazione orientata al robot

- ★ integrazione di numerose funzioni di linguaggi di programmazione ad alto livello (BASIC, PASCAL) con quelle specifiche richieste nelle applicazioni robotiche
- Alternative
  - ★ sviluppo di *linguaggi ad hoc* per applicazioni robotiche
  - ★ sviluppo di *librerie di programmi* per robotica di supporto a linguaggi standard di programmazione
  - ★ programmatore esperto
  - ★ programmazione fuori linea
  - ★ interazione con altre macchine

- Caratteristiche di linguaggio di *programmazione strutturata*
  - ★ editore di testo
  - ★ strutture complesse di rappresentazione dei dati
  - ★ uso estensivo di variabili di stato predefinite
  - ★ realizzazione di operazioni di algebra matriciale
  - ★ uso estensivo di rappresentazioni simboliche per le terne di coordinate
  - ★ possibilità di specificare il moto coordinato di più terne relative a oggetti connessi rigidamente per il tramite di una sola terna
  - ★ inclusione di sottoprogrammi con scambio di dati e parametri
  - ★ uso di funzioni di condizionamento logico e cicli di attesa mediante semafori
  - ★ capacità di parallelizzazione del calcolo
  - ★ funzioni di controllore logico programmabile (PLC)

# ARCHITETTURA HARDWARE



- *Scheda sistema*
  - ★ un microprocessore con coprocessore matematico
  - ★ una memoria EPROM di inizializzazione
  - ★ una memoria RAM locale
  - ★ una memoria RAM condivisa con le altre schede attraverso il bus
  - ★ un insieme di porte seriali e parallele di interfaccia con il bus e con il mondo esterno
  - ★ contatori, registri, temporizzatori
  - ★ gestore delle interruzioni
  
- *Funzioni*
  - ★ interfaccia con l'operatore attraverso teach pendant, tastiera, video, stampante
  - ★ interfaccia con una memoria esterna (disco rigido) usata per immagazzinare dati e programmi applicativi
  - ★ interfaccia con rete locale di connessione (per esempio, Ethernet) con stazioni di lavoro e con altre unità di governo
  - ★ interfaccia I/O con apparecchiature periferiche presenti nell'area di lavoro (alimentatori, nastri trasportatori, sensori ON/OFF)
  - ★ inizializzazione del sistema
  - ★ interprete del linguaggio di programmazione
  - ★ arbitro del bus
  
- *Processori suppletivi o alternativi (DSP, Transputer)*

- Funzioni della scheda *cinematica*
  - ★ calcolo delle primitive di moto
  - ★ calcolo della cinematica diretta, della cinematica inversa e dello Jacobiano
  - ★ verifica di congruenza della traiettoria
  - ★ gestione della ridondanza cinematica
  
- Funzioni della scheda *dinamica*
  - ★ calcolo della dinamica inversa
  
- Funzioni della scheda *servo*
  - ★ microinterpolazione dei riferimenti
  - ★ calcolo dell'algoritmo di controllo
  - ★ conversione D/A ed interfaccia con amplificatori di potenza
  - ★ trattamento dati dei trasduttori di posizione e di velocità
  - ★ interruzione del moto in situazioni di malfunzionamento
  
- Funzioni della scheda *forza*
  - ★ condizionamento dei segnali forniti dal sensore di forza
  - ★ rappresentazione delle forze in terne di coordinate assegnate
  
- Funzioni della scheda *visione*
  - ★ elaborazione dei segnali forniti dalla telecamera
  - ★ estrazione di caratteristiche geometriche della scena
  - ★ localizzazione degli oggetti in terne di coordinate assegnate