



# ***Bazele Roboticii***

## **Curs 08**

### **Programarea roboţilor**

**Gigel Măceşanu**



# Cuprins

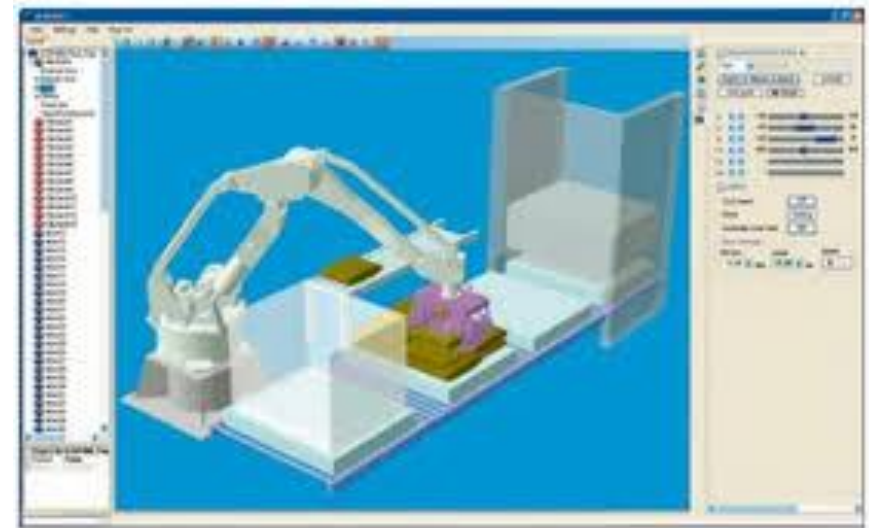
- Introducere
- Programarea online și offline
- Limbaje de programare





# Programarea robotului

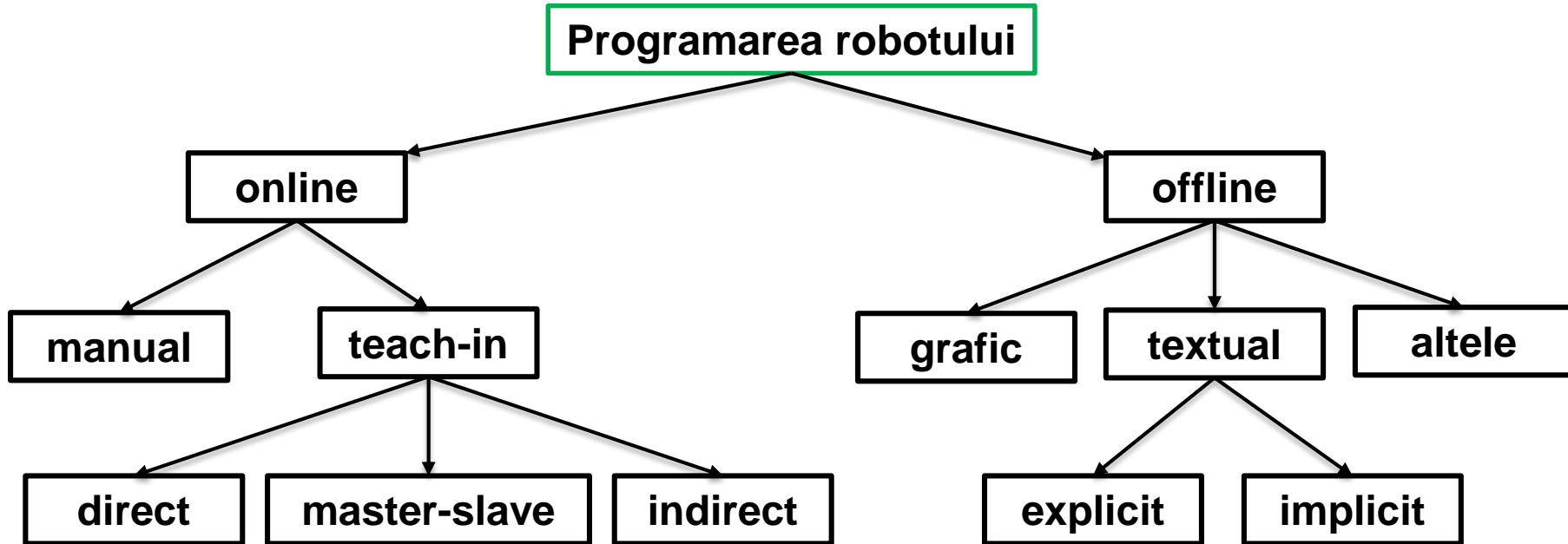
- **Programarea robotului** reprezintă transferarea unei succesiuni de mișcări complexe către controller-ul robotului, folosind instrucțiuni simple și diverse metode de programare, astfel încât să poată fi automatizat un proces de fabricație
- nu există încă un singur limbaj de programare pentru toate tipurile de roboți
- fiecare producător mai important poate avea implementat un limbaj de programare specific pentru roboții pe care îi produce.





# Metode de programare

- Metodele de programare disponibile sunt următoarele:



- Pentru programarea unei aplicații se pot folosi și combinații dintre mai multe metode de programare
- Este un lucru obișnuit să se folosească programarea *teach-in* pentru corecția pozițiilor planificate într-un program creat prin metoda *offline*



# Programarea online

- Metodele de programare *online* suportă în general numai comenzi pentru mișcări. De obicei nu suportă comenzi care să țină cont de senzori externi.
- Se împarte în:
  - **programarea manuală**
  - **programarea teach-in.**
- Programarea manuală se folosește numai pentru dispozitive de alimentare/evacuare



# Programarea manuală

- Programarea *manuală* înseamnă stabilirea de puncte de mișcare prin limitatori statici (opritori), la nivelul axelor
  - Se folosesc coordonatele robot
  - Sistemul de coordonate robot folosește valorile pozițiilor articulațiilor pentru a specifica un punct din spațiul de lucru
  - Articulațiile pot fi liniare sau rotaționale
  - De exemplu, în cazul roboților articulați cu 6 axe, pentru determinarea unică a unei poziții se dau toate cele 6 valori dorite pentru poziția articulațiilor.



# Programarea Teach-in

- Poate fi împărțită în trei categorii:
  - **teach-in direct,**
  - **programare master-slave**
  - **teach-in indirect**
  
- Caracteristici comune ale metodelor:
  - **Robotul este mișcat în timpul programării.**
  - **Punctele de poziționare sunt definite de senzorii interni (traductoarele de deplasare atașate axelor).**



# Programarea Teach-in

## ▪ Teach-in direct

- **Programatorul conduce cu mâna end-effector-ul robotului în pozițiile pe care dorește să le obțină (robotul are inhibat sau dezactivat sistemul de frânare al axelor).**
- **Pozițiile robotului sunt înregistrate:**
  - **direct (la o comandă explicită a programatorului)**
  - **automat (înregistrează exact toate mișcările pe care operatorul le aplică end-effector-ului).**
- **Programarea cu teach-in direct este utilă pentru roboții folosiți în aplicații de tratare a suprafețelor (vopsire, polizare, șlefuire etc.).**





# Programarea Teach-in

## ▪ Programarea *Master-Slave*

- este comparabilă cu programarea teach-in direct
- operatorul nu mișcă robotul în mod direct, ci folosește un model miniatură (sau simplificat) al robotului
- robotul principal (slave) va urmări mișcările realizate cu robotul model (master).

## ▪ *Teach-in indirect*

- este cea mai utilizată metodă de programare a roboților.
- robotul este mișcat prin acționarea unor taste funcționale de pe panoul de operare (Teach Panel) al robotului
- poziția dorită a robotului este memorată prin apăsarea unei taste funcționale de pe panoul de operare al robotului.



# Programarea *offline*

- Programarea *offline* a unui robot se face, de exemplu, prin generarea de text în programul robotului, respectând o sintaxă dată de un limbaj de programare specific roboților.
- Avantaje:
  - Nu este necesară prezența unui robot în timpul programării.
  - Se pot dezvolta programe complexe prin utilizarea unor structuri de programare (if.. then..., for .., while.. do..., etc.).
  - Sunt ușor de tratat semnale de la senzori și de utilizat comenzi pentru ieșiri digitale.
- Dezavantaje:
  - Deseori pozițiile nu se pot defini exact, din calcule matematice (este necesară și utilizarea metodei teach-in).
  - Este dificil de realizat un test complet al programului în modul *offline*.



# Programarea *offline*

- Programarea *offline* se poate realiza astfel:
  - **Programare offline în mod text**
  - **Programare offline în mod grafic**
- Programare offline în mod text
  - **Utilizează limbaje de programare de nivel înalt**
  - **Necesită un mediu de programare pentru:**
    - Implementarea (scrierea) programelor
    - Verificarea sintactică, semantică și compilarea programelor
    - Testarea programelor (simulare)
    - Stocarea și organizarea programelor (în directoare)
    - Interfețe către alte sisteme de dezvoltare programe
    - Comunicație cu utilizatorul
    - Încărcare (recuperare) a programelor în (din) controller-ul robotului (download, upload sau backup)



# Programarea *offline*

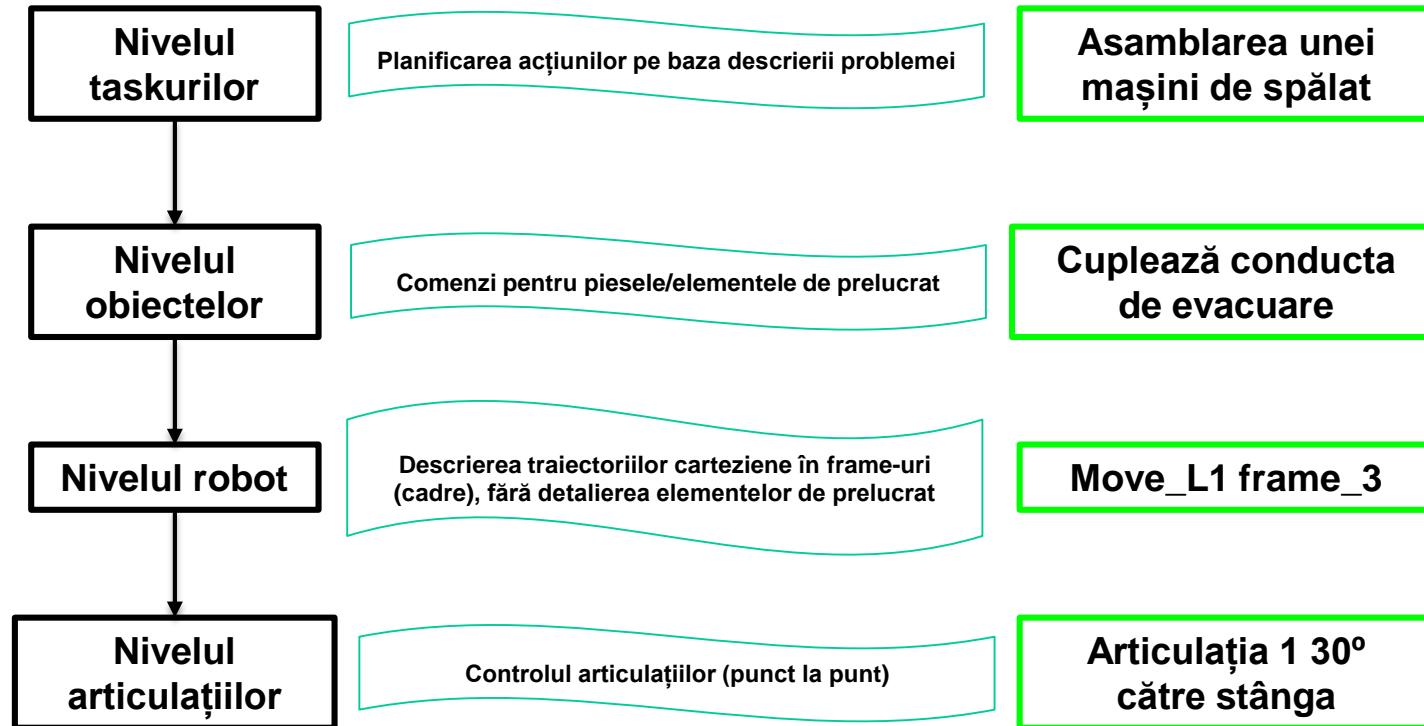
- Programarea *offline* în mod grafic

- Se folosesc sisteme CAD (*Computer-Aided Design*) în scopul de a modela geometric obiectele aplicației
- Utilizatorul introduce diverși parametri necesari în definirea caracteristicilor fiecărei traiectorii
- Sistemul calculează o traiectorie ca o secvență de puncte intermediare prin care robotul trebuie să treacă
- Funcția de simulare a sistemului poate reprezenta grafic orice mișcare, în vederea testării
- este folosit din ce în ce mai mult pentru aplicații complexe de tratare a suprafețelor: vopsire, polizare, șlefuire, periere, debavurare, sudare etc.



# Limbaje de programare pentru roboți

- Clasificarea se face după gradul de abstractizare:

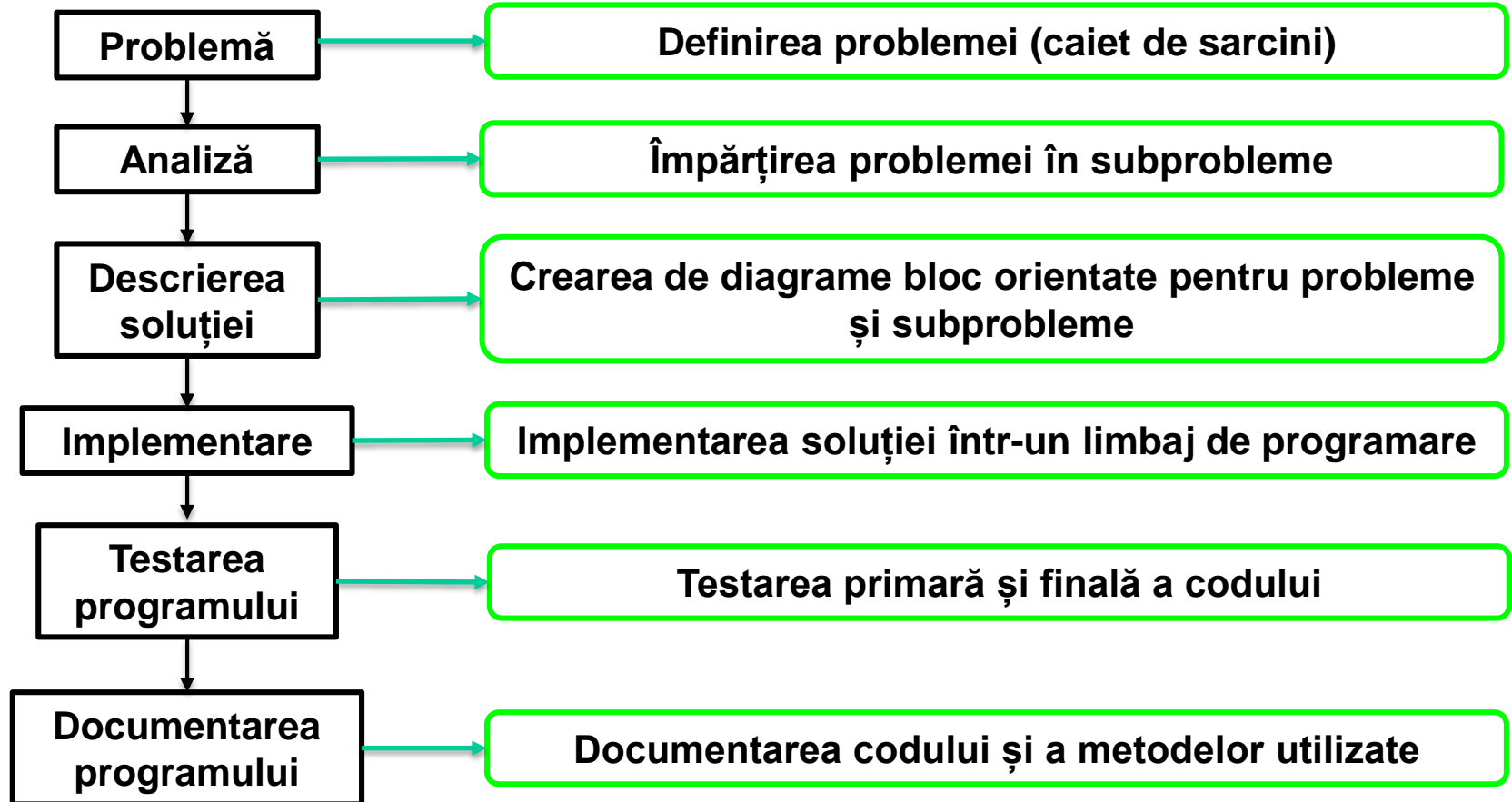


- **Limbajele de la nivelele task layer și object layer sunt denumite limbaje implicite**
- **Limbajele explicite sunt cele care operează la nivelele robot layer și joint layer**



# Limbaje de programare pentru roboți

- Dezvoltarea programelor pentru roboți industriali



- **La programele complexe se realizează împărțirea în sub-programe, ce conțin funcții clare și diferite**



# Limbaje de programare pentru roboți

- Dintre limbajele folosite pentru programarea roboților, amintim:
  - **MRL – Mitsubishi Robol Language**
    - programare a fost conceput și folosit de firma Mitsubishi pentru programarea roboților săi industriali din primele generații
    - încărcate din PC în controller sub formă de fișiere text
    - Controller-ul acționează ca un interpretor de comenzi, executând instrucțiunile linie după linie, la fel ca la mașinile CNC
  - **IRL – Industrial Robot Language**
    - descrie structura de bază și modul de realizare a programului de nivel înalt pentru roboți industriali IRL (*Industrial Robot Language*)
    - Limbajul permite utilizatorului să programeze mișcările unui robot și să execute organigrame logice conform unei structuri de program definite de utilizator



# Limbaje de programare pentru roboți

- **SLIM – Standard Language for Industrial Manipulators**
  - construit pe baza modelului limbajului BASIC
  - s-au adăugat instrucțiuni speciale pentru controlul mișcărilor și pentru tratarea intrărilor/ieșirilor digitale
  
- **MELFA-BASIC III și IV – Limbaje de programare, de nivel înalt, pentru roboți industriali**
  - este un limbaj de programare pentru roboți care are la bază limbajul universal BASIC
  - Conține două părți:
    - **Partea Declarativă:** Unde se declară datele utilizatorului (declarații de variabile, constante etc.)
    - **Partea de Instrucțiuni:** Unde se implementează programul conform cerințelor aplicației: se controlează mișcările robotului, se realizează comunicațiile cu echipamentele periferice, cu calculatorul sau cu panoul de control.





# Limbaje de programare pentru roboți

## ➤ Python

- limbaj declarativ care este axat pe programarea pe obiecte
- Aspectul declarativ se referă la faptul că programele în Python (Py) nu sunt compilate, ele având nevoie de un interpretor:
  - **Interpretor: este un program utilizat să transforme instrucțiunile Py în cod binar, cel care poate fi rulat pe procesorul calculatorului**

## ➤ Matlab

- Conține librării pentru programarea și controlul roboților
- Algoritmii disponibili: reprezentarea teatrului de operațiuni, planificarea mișcării, urmărirea traiectoriei
- Permite conectarea către ROS (*Robot Operating System*)
- Suportă generarea de cod C++, pentru aplicații unde este necesar acest limbaj



**Contact:**  
**Email: [gigel.macesanu@unitbv.ro](mailto:gigel.macesanu@unitbv.ro)**  
**Web: [rovis.unitbv.ro](http://rovis.unitbv.ro)**