

Laborator 2

Pick and Place

Operația de Pick and Place presupune deplasarea unei piese din poziția A ($pick$) în poziția B ($place$) cu ajutorul robotului.

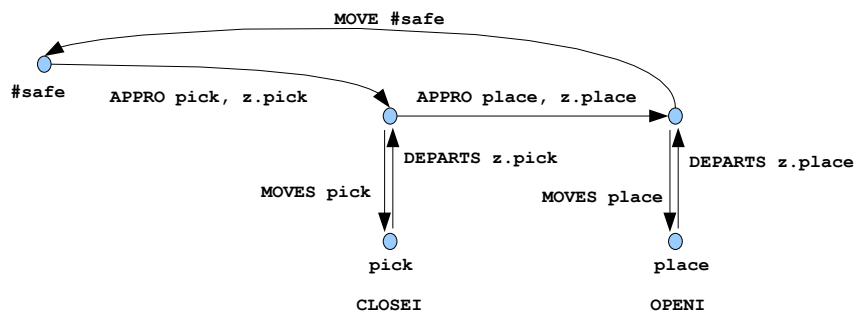


Figura 2.1: *Secvența Pick and Place*

Mișcarea robotului va începe dintr-o locație numită $safe$, în care robotul este retras astfel încât mișcarea până în punctele $pick$ și $place$ să poată fi efectuată în siguranță, fără a lovi obiectele din jur.

Pașii necesari pentru învățarea operației de Pick and Place:

- Inițializarea sistemului
- Învățarea locației $safe$
- Învățarea locațiilor $pick$ și $place$
- Scrierea programului $pick.place$
- Testarea programului

Învățarea unei locații

Locațiile pot fi *transformări* sau *puncte de precizie*.

Un punct de precizie descrie complet poziția brațului robot prin definirea poziției individuale a fiecărei articulații. Pentru roboții Cobra, un punct de precizie are 4 componente (RRPR - rotație, rotație, translație, rotație). Pentru roboții Viper, un punct de precizie are 6 componente, toate de rotație. Punctele de precizie sunt marcate prin prefixul "#".

O transformare specifică locația (poziția și orientarea în spațiu) a efectorului terminal, și are 6 componente: $x, y, z, yaw, pitch, roll$.

Robotul Cobra poate atinge aceeași locație în două moduri: fie în configurația LEFTY, fie în RIGHTY. Există locații care pot fi atinse cu ambele configurații, precum și locații care pot fi atinse doar în LEFTY sau doar în RIGHTY.

Cu alte cuvinte, unui punct de precizie îi corespunde întotdeauna o singură transformare (o singură locație a efectorului terminal), însă unei transformări (locații) îi pot corespunde mai multe puncte de precizie. Calculul transformării asociate unui punct de precizie poartă numele de *cinematică directă*, și are întotdeauna o soluție unică, iar calculul punctelor de precizie corespunzătoare unei transformări poartă numele de *cinematică inversă*, și de obicei are mai multe soluții (Fig. 2.2). În majoritatea cazurilor, pentru cinematica inversă se poate alege o soluție unică prin specificarea *configurației* robotului.

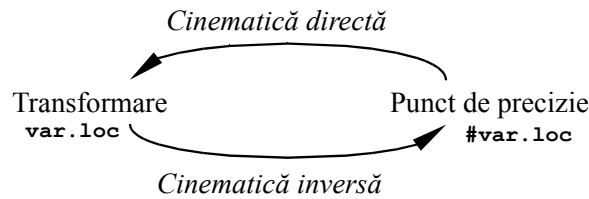


Figura 2.2: Modelul cinematic

Punctul de precizie este un vector în *spațiul articulațiilor*, iar transformarea este un vector în *spațiul operațional* (de obicei cartezian).

Transformările permit efectuarea de operații în spațiul cartezian (translații, rotații, compuneri de transformări etc). Punctele de precizie nu permit acest lucru.

Poziția *safe* se recomandă a fi învățată ca punct de precizie, deoarece este unic definită. Pozițiile *pick* și *place* se recomandă a fi învățate ca transformări,

unul din motive fiind faptul că instrucțiunea APPRO nu acceptă puncte de precizie ca argument.

Poziția se învață astfel:

- Se deplasează robotul în poziția dorită folosind MCP-ul
- Se introduce de la consola monitor V^+ comanda:
.HERE nume.locatie
- Pentru cele 3 locații, comenzile sunt:
.HERE #safe
.HERE pick
.HERE place
- Se confirmă învățarea locației cu ENTER de două ori.

Pozițiile învățate astfel sunt variabile globale și pot fi folosite în program.

Controlul mișcării

Instrucțiuni de mișcare

- MOVE dst și MOVES dst
- deplasare spre locația destinație dst
- APPRO dst, z și APPROS dst, z
- deplasare deasupra¹ locației dst, la o distanță egală cu z mm.
- DEPART d și DEPARTS d
- deplasare deasupra locației curente pe o distanță egală cu d milimetri

Instrucțiunile cu sufixul S (MOVES, APPROS, DEPARTS) execută o mișcare liniară în spațiul operațional (cartezian), efectorul terminal deplasându-se pe un segment de dreaptă. Instrucțiunile fără sufixul S (MOVE, APPRO, DEPART) execută o mișcare liniară în spațiul articulațiilor, care corespunde unei traiectorii curbilini în spațiul operațional.

În timpul unei mișcări liniare în spațiul cartezian, configurația robotului (LEFTY / RIGHTY, ABOVE / BELOW, FLIP / NOFLIP) nu poate fi schimbată. Este posibilă schimbarea configurației doar prin mișcări în spațiul articulațiilor.

¹în sensul negativ al axei Z_{TOOL}

Sistemul V^+ execută mișcările robotului în paralel cu execuția programului. Astfel, o instrucțiune de mișcare (`MOVE`, `MOVES`, `APPROS`, `DEPARTS` etc) va returna controlul imediat ce mișcarea robotului a început. Instrucțiunile următoare se vor executa în paralel cu mișcarea robotului. Pentru a aștepta terminarea mișcării curente se poate folosi instrucțiunea `BREAK`.

Setarea vitezei

Viteza de lucru a robotului se poate seta în procente față de viteza maximă, sau în milimetri pe secundă, cu ajutorul instrucțiunii `SPEED`:

- `SPEED 50 ALWAYS` - setează viteza generală de lucru la 50% din valoarea maximă (valabilă pentru întregul program)
- `SPEED 20` - setează viteza generală de lucru la 20% din valoarea maximă (valabilă doar pentru instrucțiunea de mișcare următoare)
- `SPEED 30 MMPS` - setează viteza liniară a efectorului terminal la 30 milimetri pe secundă (valabilă doar pentru instrucțiunea de mișcare următoare)

Operațiile de apropiere și depărtare se vor executa cu viteză redusă, deoarece este necesară poziționarea precisă a robotului. Deplasarea între punctele *pick* și *place* poate fi făcută cu viteză mare.

Se poate seta de asemenea viteza *monitor*, care limitează viteza globală de lucru. Astfel, o viteză de 20% monitor specificată la consola V^+ , combinată cu o viteză de `50 ALWAYS` specificată în program, va avea ca rezultat o viteză efectivă de lucru de 10% din valoarea maximă.

Viteza monitor se specifică cu comanda `monitor .speed`, care este diferită de instrucțiunea program `SPEED`. Astfel, comanda `monitor .speed 10` setează viteza monitor la 10%, și este echivalentă cu `.do speed 10 monitor`. În program, instrucțiunea `SPEED 10` va seta viteza de 10% pentru următoarea mișcare.

`.speed 10` este echivalent cu: `.do speed 10 monitor`

Viteza monitor este utilă pentru testarea programelor. În cazul în care programul conține o eroare care duce la coliziune, rularea cu viteză monitor redusă (de exemplu 5%) permite operatorului să acționeze butonul E-Stop pentru a evita accidentele. După ce utilizatorul este sigur că programul robot funcționează fără erori, poate crește viteza monitor *gradual*, asigurându-se că mărirea vitezei nu influențează negativ funcționarea programului.

Controlul gripper-ului

Gripper-ul este efectorul terminal care permite manipularea pieselor. Gripper-ele din laborator sunt acționate pneumatic; pot exista gripper-e acționate de un motor electric (mai lente), gripper-e cu vacuum (foarte rapide), electromagnetice, electrostatice etc.

Gripper-ul nu acționează instantaneu (are nevoie de un timp pentru a se închide sau se deschide). Sistemul V^+ permite setarea acestui timp folosind parametrul `HAND.TIME`. Pentru gripper-ul de pe Cobra 600TT, timpul de închidere/deschidere a gripper-ului este de 0.5 secunde. În cazul celorlalți roboți, acest parametru este 0.2 secunde.

Un gripper se poate comanda prin comenzile `OPEN / OPENI`, `CLOSE / CLOSEI`, precum și prin `MOVET / MOVEST`. Semnificația acestora este:

- `OPENI / CLOSEI` deschide sau închide gripper-ul imediat după terminarea mișcării curente, și efectuează o temporizare egală cu valoarea parametrului `HAND.TIME`.
- `OPEN / CLOSE` deschide sau închide gripper-ul în timpul mișcării următoare, fără temporizare.
- `MOVET dst, TRUE / MOVET dst, FALSE`, precum și `MOVEST` cu parametri identici: deschide/închide gripper-ul în timpul mișcării spre `dst`. Echivalent cu `OPEN / CLOSE` urmat de `MOVE / MOVES`.

Testarea programului

Testarea unui program V^+ cuprinde următoarele etape:

- Rularea în modul *Dry Run*, fără a mișca efectiv robotul;
- Rularea cu viteză redusă, de 5%;
- Creșterea graduală a vitezei de lucru: 20, 50, 70 și 100%.

Înainte de rularea efectivă a programului pe robot, acesta trebuie testat în modul *Dry Run*, fără a mișca robotul. Acest pas permite depistarea erorilor de logică din program, precum și existența variabilelor neinițializate sau tastate greșit.

Pașii necesari pentru testarea unui program robot:

- Se selectează din MCP un mod de lucru diferit de *Comp*. Acest lucru previne orice posibilitate de a mișca robotul accidental din program sau din linia de comandă. Pentru mai multă siguranță, viteza monitor se setează la 5%:

```
.speed 5
```

- Se activează switch-ul DRY.RUN:

```
.enable dry.run
```

- Se execută programul:

```
.exec numeprogram
```

- Se verifică mesajele afișate la consolă; în cazul în care există erori, acestea se corectează. Este utilă afișarea variabilelor folosite în program cu instrucțiunea TYPE, sau activarea switch-ului TRACE pentru afișarea liniilor de program executate.
- După ce programul rulează corect în modul *Dry Run*, acesta se rulează cu viteză redusă:

```
.disable dry.run
```

```
.speed 5
```

```
.exec numeprogram
```

- Dacă programul a rulat corect cu viteză redusă, aceasta poate fi crescută gradual, de exemplu la 20, 50, 70 și 100%. În timpul rulării programului, utilizatorul este pregătit să acționeze butonul Emergency Stop în cazul în care traseul robotului nu este cel dorit.

Programul Pick and Place

```
1 .PROGRAM pick.place()
2     ; Laboratorul 2 - Pick and Place
3
4     GLOBAL #safe, pick, place ; locatii robot
5     AUTO z.pick
6     AUTO z.place             ; variabile locale
7     z.pick = 80
8     z.place = 100
9
10    PARAMETER HAND.TIME = 0.5
11    SPEED 100 ALWAYS
12
13    OPEN
14    MOVE #safe      ; incepe miscarea din #safe
15    BREAK          ; cu gripper-ul deschis
16
17    APPRO pick, z.pick ; deplasare deasupra
18    BREAK          ; punctului 'pick'
19
20    SPEED 50
21    MOVES pick
22    CLOSEI          ; prindere piesa
23
24    SPEED 30
25    DEPARTS z.pick
26    BREAK
27
28    APPRO place, z.place ; deplasare deasupra
29    BREAK          ; punctului 'place'
30
31    SPEED 20
32    MOVES place
33    OPENI          ; asezare piesa
34
35    SPEED 50
36    DEPARTS z.place
37    BREAK
38
39    MOVE #safe      ; intoarcere in #safe
40 .END
```

Teme

Tema 1

Identificați instrucțiunile de mișcare ale limbajului V^+ și grupați-le pe categorii:

- Instrucțiuni care operează cu transformări
- Instrucțiuni care operează cu puncte de precizie
- Instrucțiuni care operează atât cu transformări, cât și cu puncte de precizie

Tema 2

Se presupune că punctul `pick` poate fi atins doar în configurația `LEFTY`, iar punctul `place` poate fi atins doar în `RIGHTY`. În cazul în care un punct nu poate fi atins cu configurația curentă, V^+ va afișa un mesaj de eroare.

Modificați programul `pick.place` astfel încât să poată funcționa corect în situația descrisă mai sus.

Tema 3

- Ce se întâmplă dacă programatorul uită să seteze parametrul `HAND.TIME`? Ce urmări poate avea setarea incorectă a acestui parametru?
- Ce se întâmplă dacă în loc de `OPENI/CLOSEI` se folosește `OPEN/CLOSE` pe liniile 22 și 33 în programul `pick.place`? Modificați programul astfel încât să funcționeze corect cu `OPEN/CLOSE`.
- Ce se întâmplă dacă se folosește `OPENI` în loc de `OPEN` pe linia 13 din programul `pick.place`?