

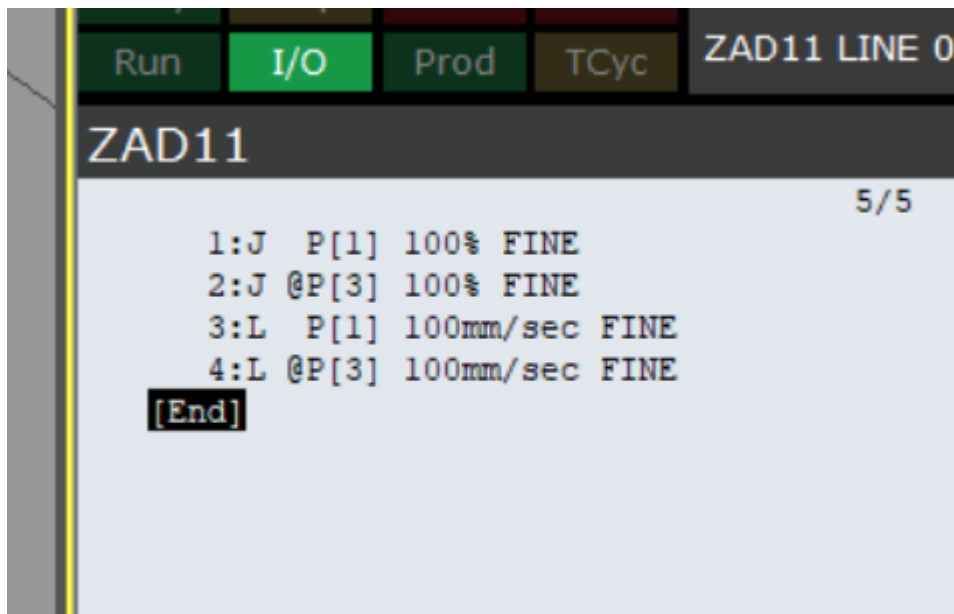
Zadanie 2

Programowanie robota FANUC

Jan Bronicki, Kacper Gąsieniec

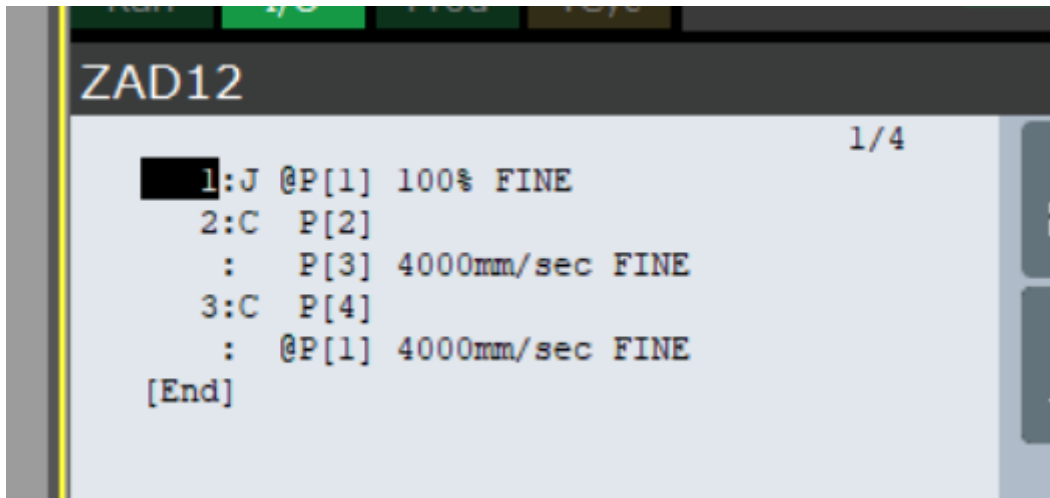
Zadanie 1

1. W pierwszym podpunkcie porównujemy ruch manipulatora w dwóch różnych przestrzeniach: konfiguracyjnej oraz zadaniowej. W tym celu tworzymy program, który porusza manipulatorem pomiędzy tymi samymi punktami w przestrzeni zadaniowej J i przestrzeni konfiguracyjnej L

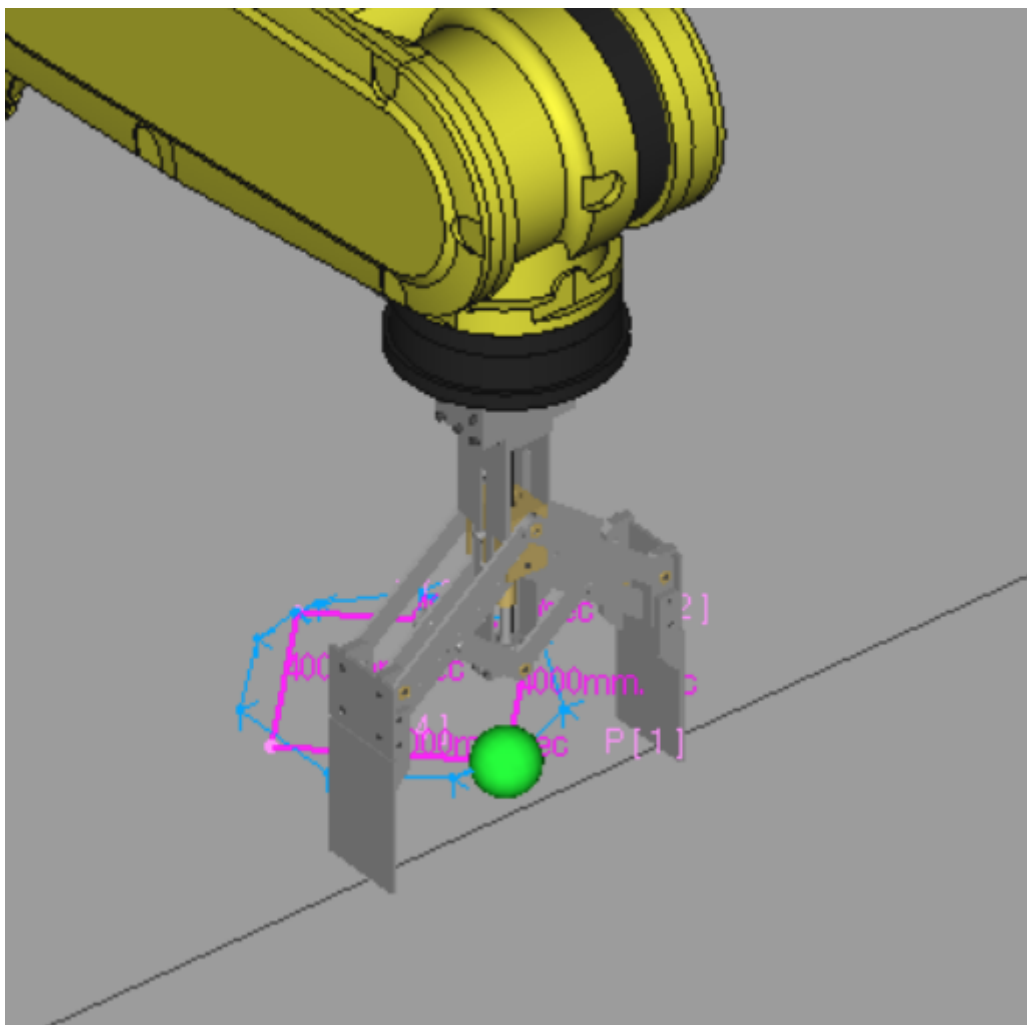


Widać różnice w poruszaniu się w obu przestrzeniach. W zadaniowej J manipulator porusza się dobierając najoptymalniejszą ścieżkę ruchu, dla innych punktów, a w przestrzeni konfiguracyjnej manipulator porusza się w liniach prostych

2. W podpunkcie drugim zakreślamy manipulatorem okrąg. Wykorzystujemy tryb Circle. Zaznaczamy punkt początkowy a potem drugi okręgu oraz punkt 3 do którego manipulator poruszy się po łuku, następnie powtórzyliśmy to dla punktów 4 i 1

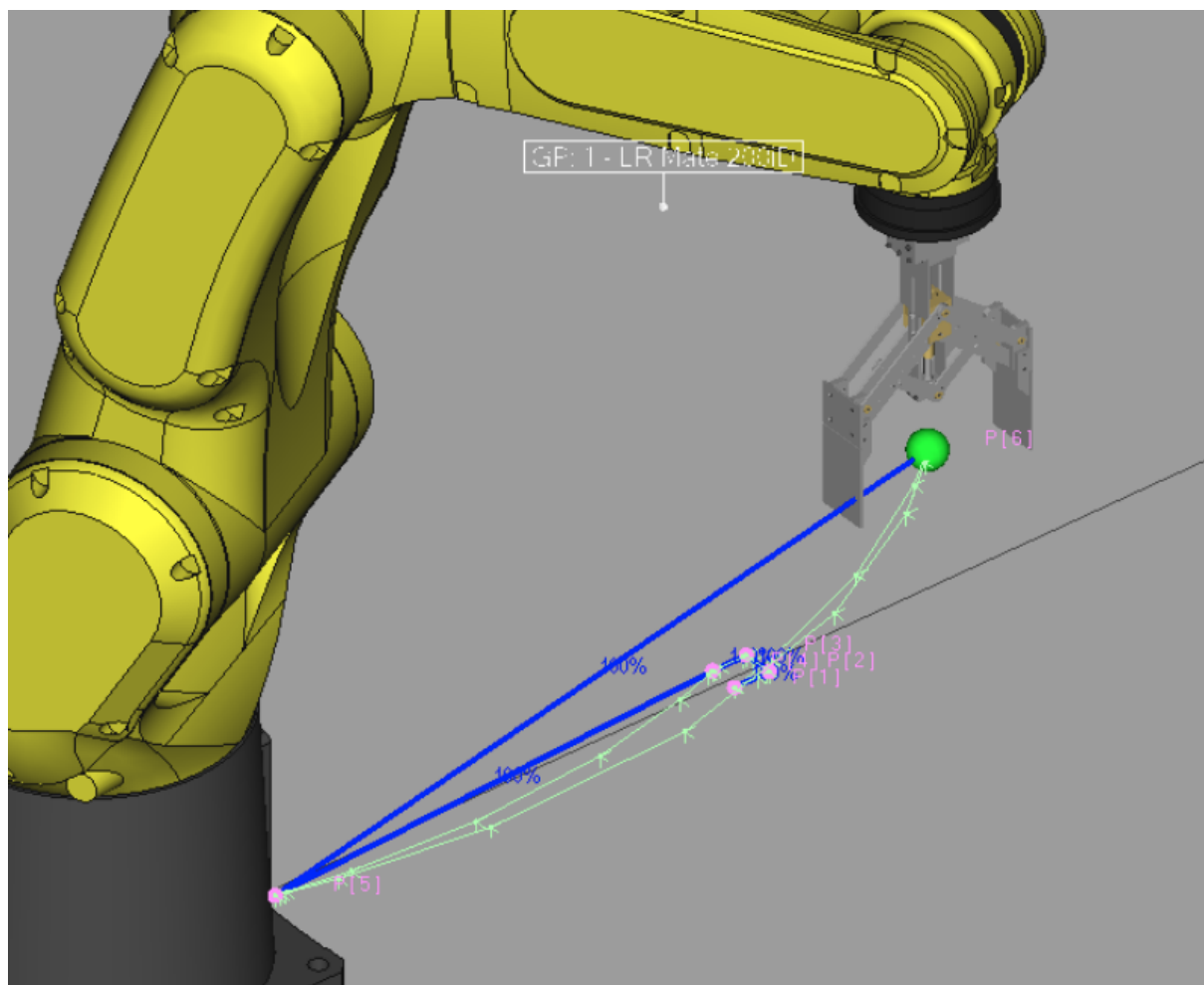
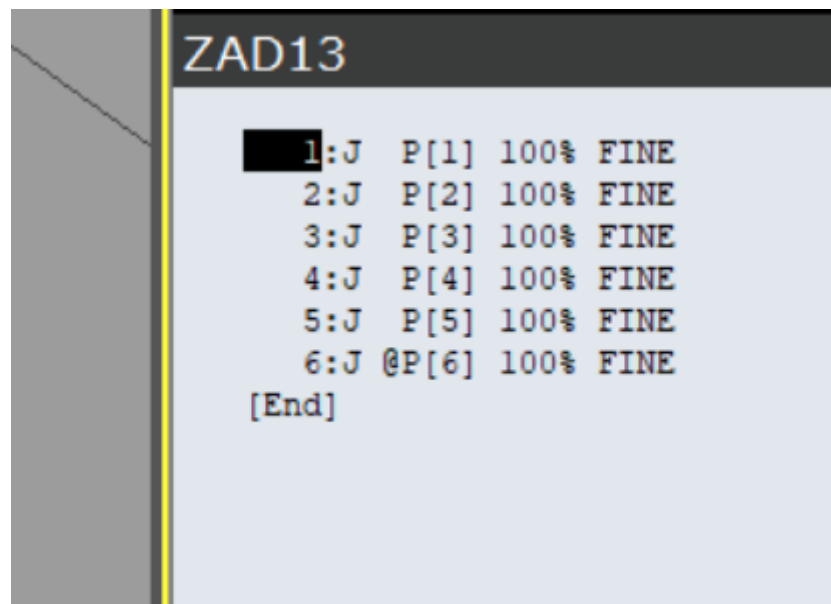


Manipulator jest w stanie nakreślić okrąg dzięki zastosowanym metodą

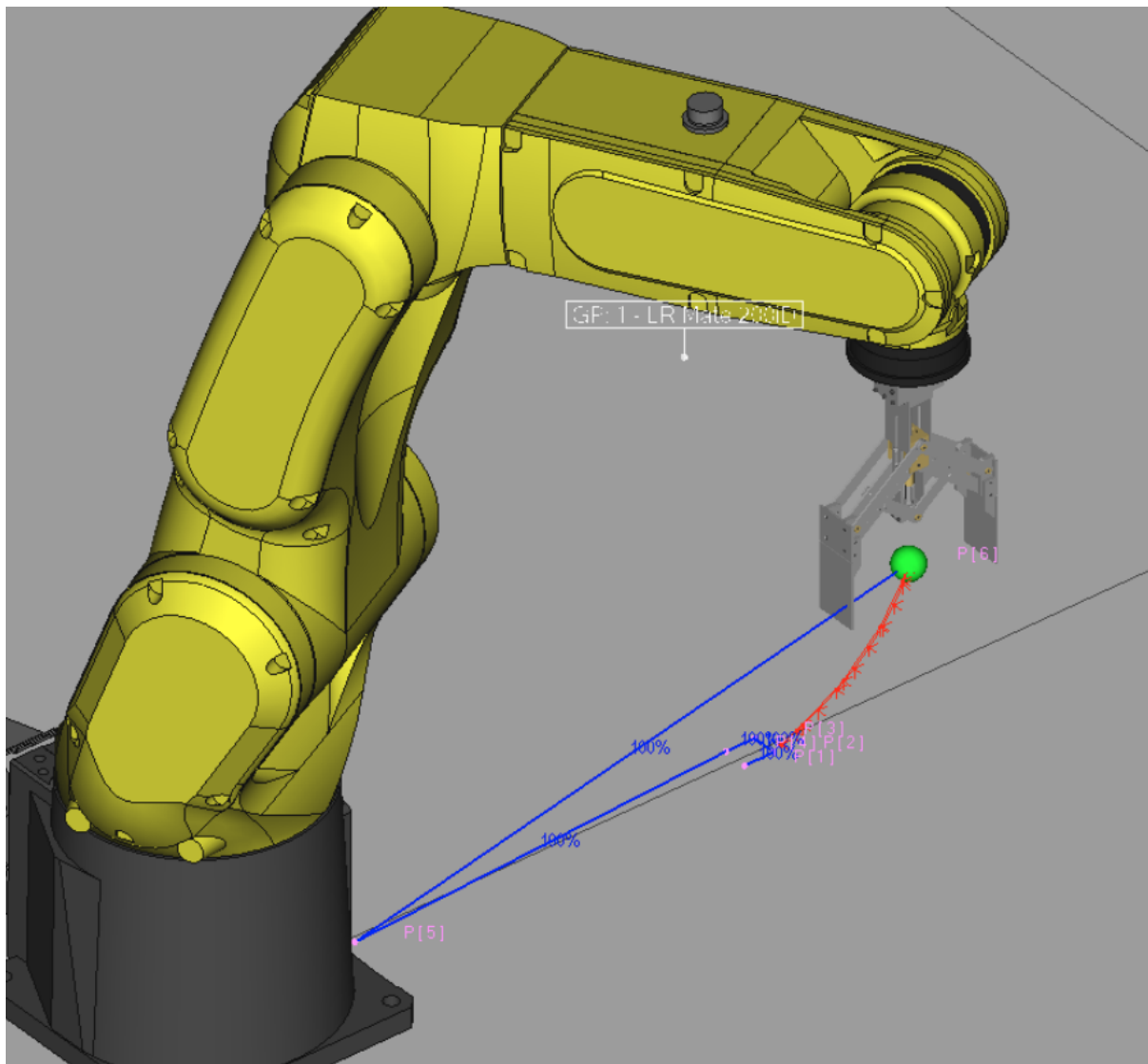
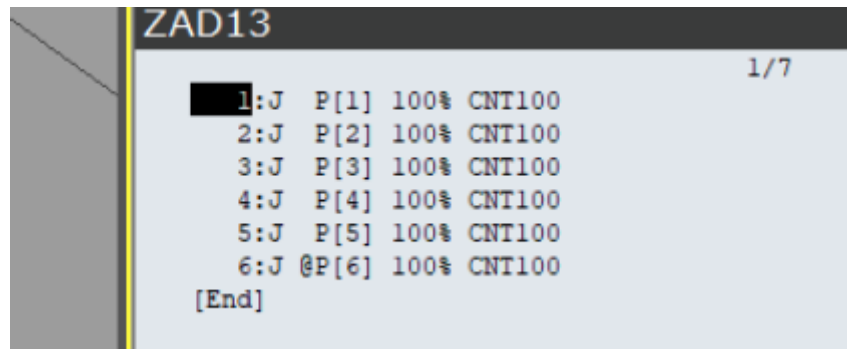


3. W trzecim podpunkcie porównujemy dwa typy ścieżek: FINE i CNT100. Zaznaczamy 6 punktów, 4 blisko siebie, 2 daleko i porównujemy ruchy manipulatora dla obu ścieżek

FINE



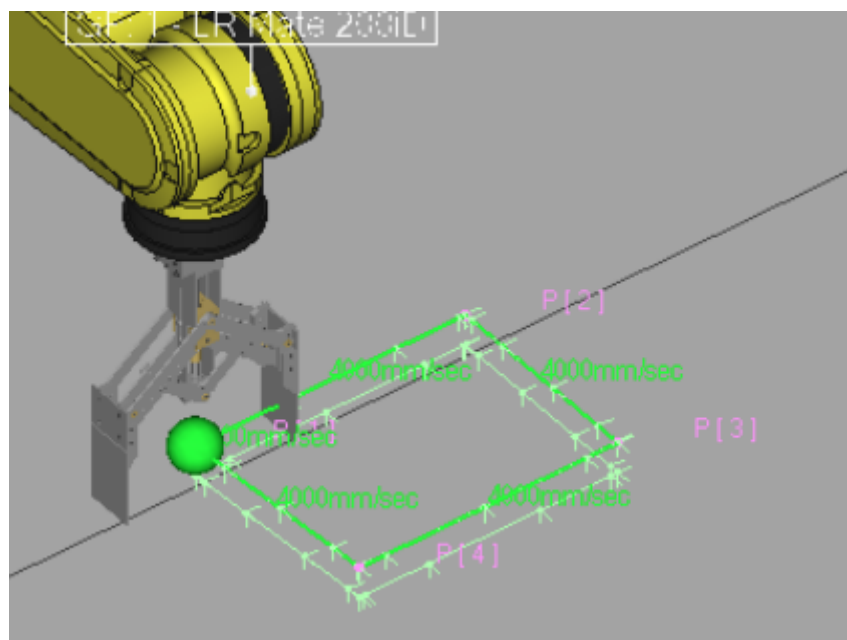
CNT100

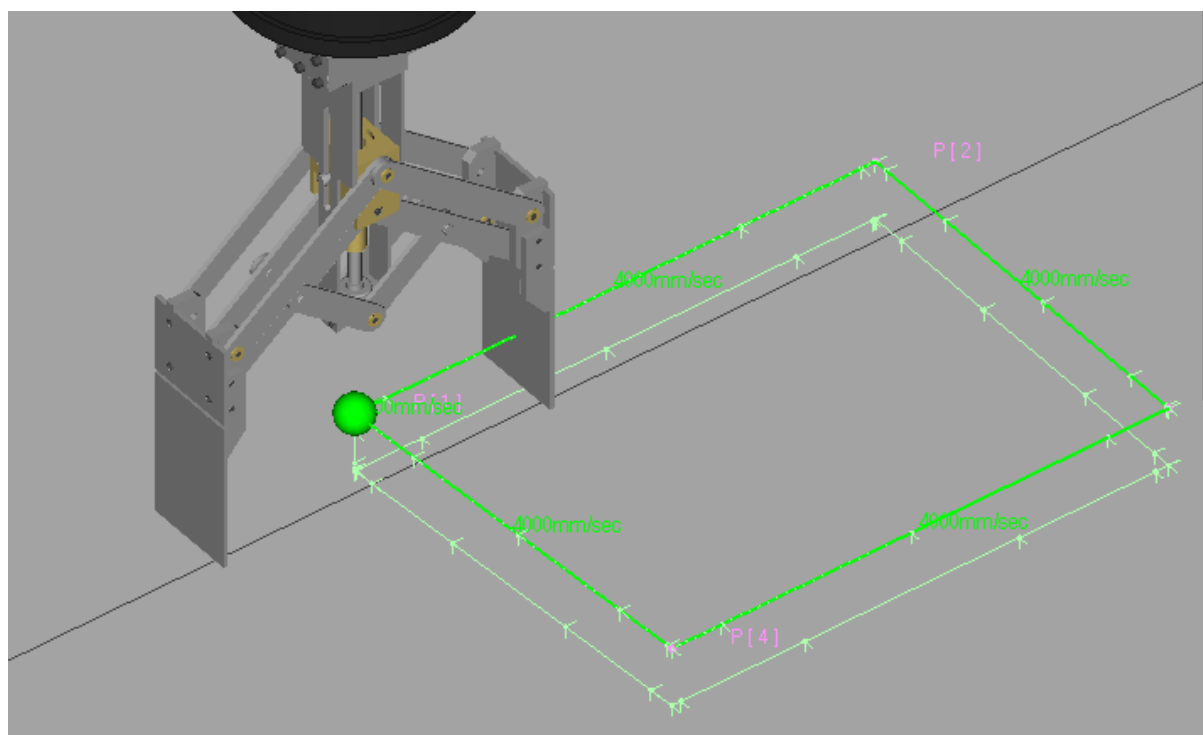
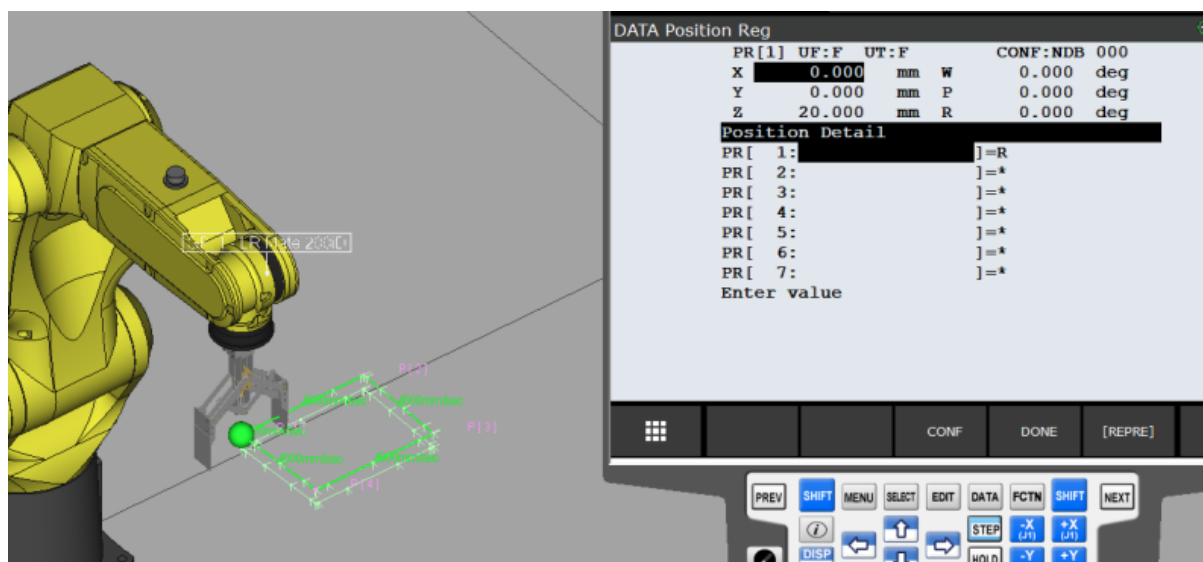


Ruch w trybie FINE jest bardziej dokładny, ale manipulator zatrzymuje się na krótki czas w danym punkcie. Ruch w CNT100 jest mniej dokładny. Widać to bardzo w punktach blisko siebie, ale za to ruch jest płynniejszy

4. W punkcie czwartym mamy za zadanie zakreślić kwadrat. A potem korzystając z OFFSET zmodyfikować parametry tego kwadratu. W pętli głównej wyznaczamy 4 punkty po których porusza się manipulator, a potem dla punktów 5-8 o tych samych pozycjach do punktu 1-4 wyznaczamy OFFSET i zmieniamy położenie zmieniając parametr Z

```
1:L @P[1] 4000mm/sec FINE
: Offset,PR[1]
2:L P[2] 4000mm/sec FINE
: Offset,PR[1]
3:L P[3] 4000mm/sec FINE
: Offset,PR[1]
4:L P[4] 4000mm/sec FINE
: Offset,PR[1]
5:L @P[1] 4000mm/sec FINE
: Offset,PR[1]
6: PR[1,3]=20
7:L @P[1] 4000mm/sec FINE
8:L P[2] 4000mm/sec FINE
9:L P[3] 4000mm/sec FINE
10:L P[4] 4000mm/sec FINE
11:L @P[1] 4000mm/sec FINE
[End]
```

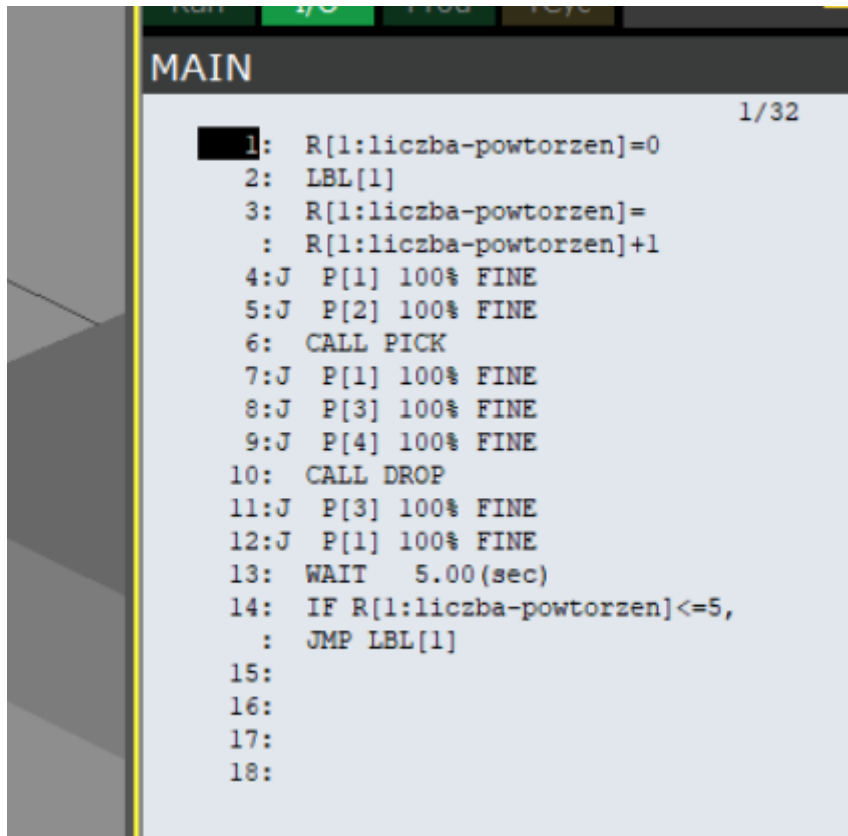




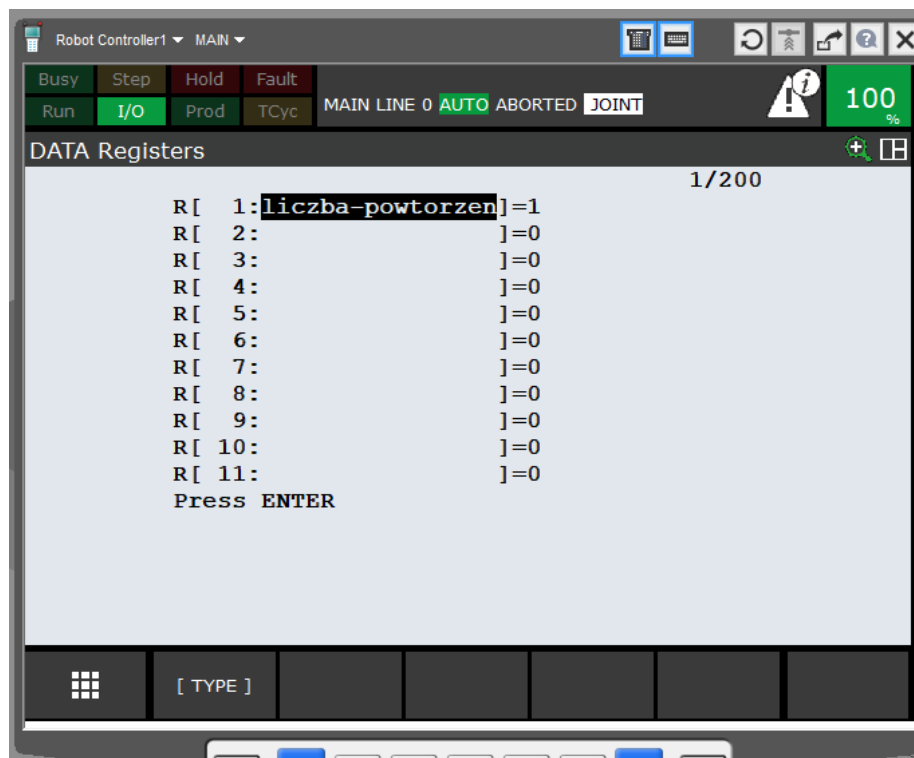
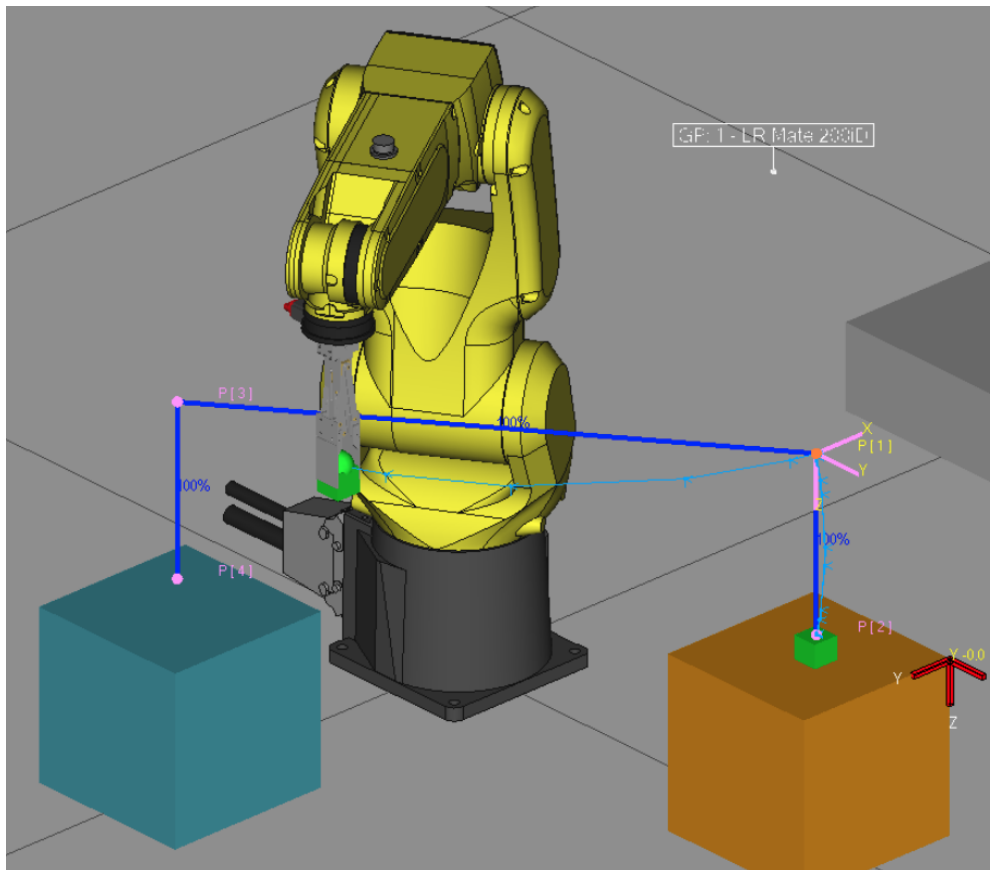
Za pomocą funkcji OFFSET można w łatwy sposób modyfikować parametry wielu punktów naraz.

Zadanie 2

Piszemy program w którym robot przenosi klocki z jednego stołu na drugi. W pierwszej kolejności piszemy programy PickUp i Drop w SimulationProgramEditor, które służą do obsługi podnoszenia i opuszczania klocków w odpowiednich miejscach. Następnie za pomocą TechPendants piszemy program, który wykonuje w pętli 5 powtórzeń podnoszenia przemieszczania i upuszczania klocka. Po każdym cyklu program oczekuje 6s (łącznie z ruchem ok 8s) na pojawienia się nowego klocka. Po 5 cyklach program kończy zadanie:

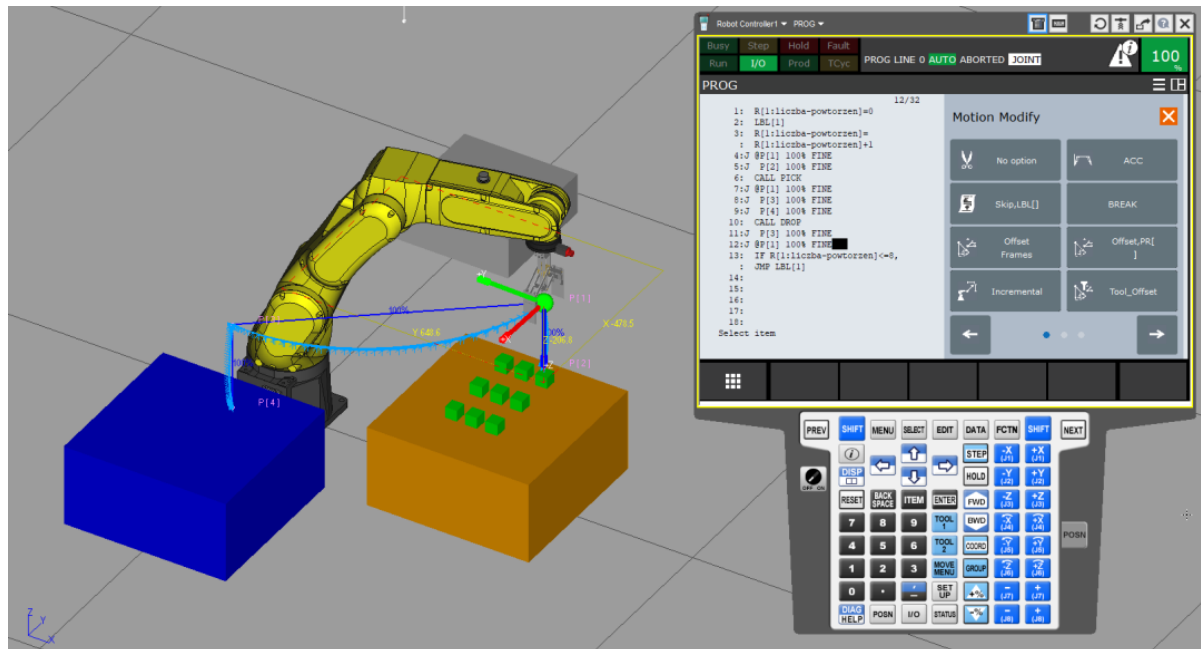


```
MAIN
1/32
1: R[1:liczba-powtorzen]=0
2: LBL[1]
3: R[1:liczba-powtorzen]=
   : R[1:liczba-powtorzen]+1
4:J  P[1] 100% FINE
5:J  P[2] 100% FINE
6:  CALL PICK
7:J  P[1] 100% FINE
8:J  P[3] 100% FINE
9:J  P[4] 100% FINE
10: CALL DROP
11:J  P[3] 100% FINE
12:J  P[1] 100% FINE
13: WAIT 5.00(sec)
14: IF R[1:liczba-powtorzen]<=5,
   : JMP LBL[1]
15:
16:
17:
18:
```



Zadanie 3

Zadanie 3 polega na przeniesieniu palety bloczków z jednego stołu na drugi. W ćwiczeniu wykorzystujemy 6 punktów:



W teorii polecenie OFFSET powinna przesuwac manipulator pomiedzy klockami (po X o 100 po Y o 50), ale nie udalo sie nam tego zaimplementowac (bład programu). OFFSET powinien zmienic pozycje punktu pierwszego w taki sposob zeby znajdowal sie nad nastepnym klockiem. Ewentualnie mozna bylo zaprogramowac osobne punkty dla wszystkich kolejnych obiektow.