

# ***Systemy programowania robotów przemysłowych - obsługa i podstawy programowania robotów KUKA***

## **Sterownik KR C4(8.x)**

1

### ***Klasy robotów KUKA***



#### **małe obciążenia**

KR 3  
KR 6  
KR 6 ARC  
KR 6 KS  
KR 16  
KR 16 L6  
KR 16 L6 ARC  
KR 16 KS  
KR 16 L6 KS



#### **średnie obciążenia**

KR 30 L16  
KR 30 K  
KR 60 K  
KR 30-3  
KR 60-3  
KR 30 HA  
KR 60 HA  
KR 30-3 KS  
KR 60-3 KS



#### **wysokie obciążenia**

KR 100 comp  
KR 140 comp  
KR 200 comp  
KR 150-2  
KR 180-2  
KR 180 L130-2 CR  
KR 210-2  
KR 240-2  
KR 150-2 K  
KR 180-2 K  
KR 210-2 K  
KR 150 W



#### **bardzo wysokie obciążenia**

KR 360  
KR 500  
KR 1000 TITAN

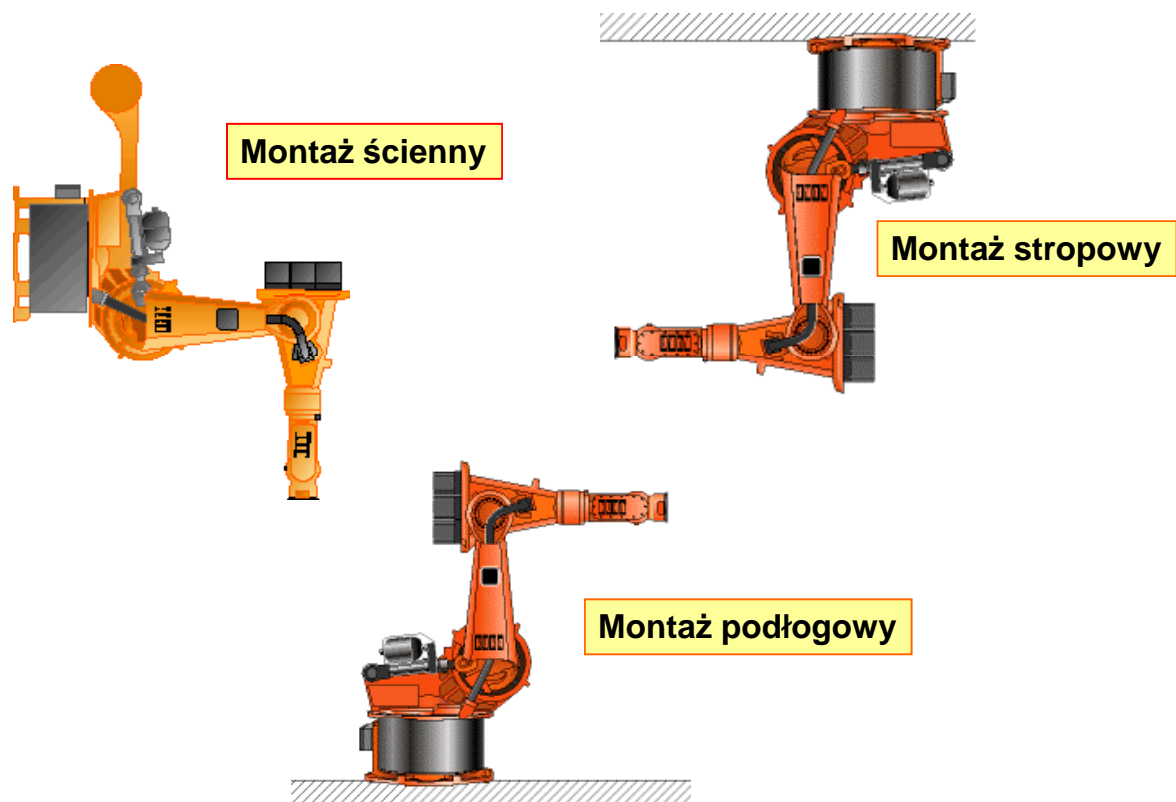


#### **konstrukcje specjalne**

KR 3 SI  
KR 100-2 PA  
KR 180-2 PA  
KR 360 450 PA  
KR 500 570 PA  
KR 100-2 P  
KR 120-2 P  
KR 360 L150 P

2

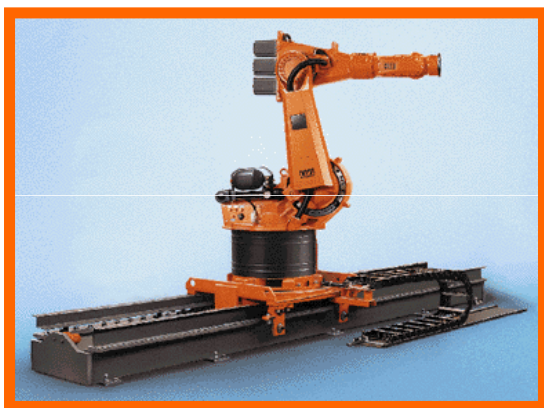
## Możliwości montażu robotów KUKA



3

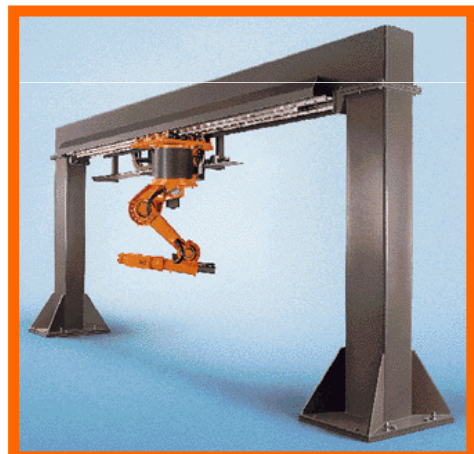
## Jednostki liniowe KUKA

Jednostki liniowe KUKA zwiększające swobodę ruchu  
Możliwość montażu podłogowego i stropowego



Montaż podłogowy

Montaż stropowy



4

## Osobliwości manipulatorów KUKA typu 6R

Standardowa kinematyka sześćoosiowa rozróżnia trzy różne konfiguracje osobliwe. Są to:

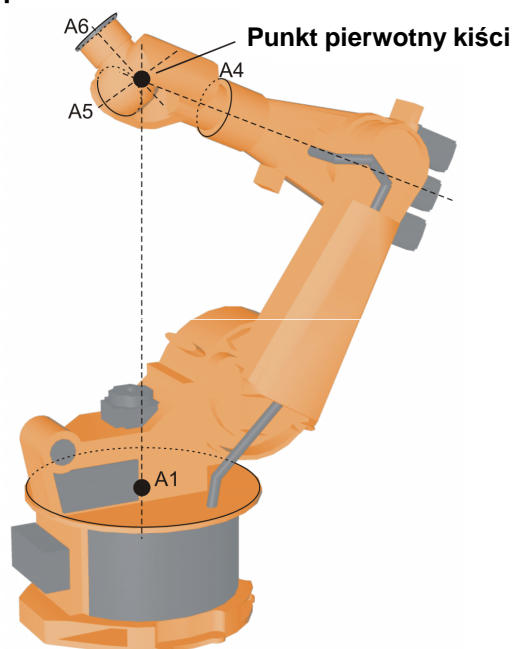
- osobliwość pułapowa,
- osobliwość pozycji wyprostowanej,
- osobliwość osi kiści (nadgarstka).

Osobliwości charakteryzują się między innymi tym, że kinematyka odwrotna (przeliczenie współrzędnych kartezjańskich na współrzędne przegubowe) pomimo ustalonych parametrów „Status” i „Turn”, nie jest jednoznacznie określona. Mała zmiana w układzie kartezjańskim w bezpośrednim sąsiedztwie osobliwości prowadzi do dużych zmian wartości kątowych osi (dużych prędkości ruchu w złączach!).

5

## Osobliwości manipulatorów KUKA typu 6R

### Osobliwość pułapowa

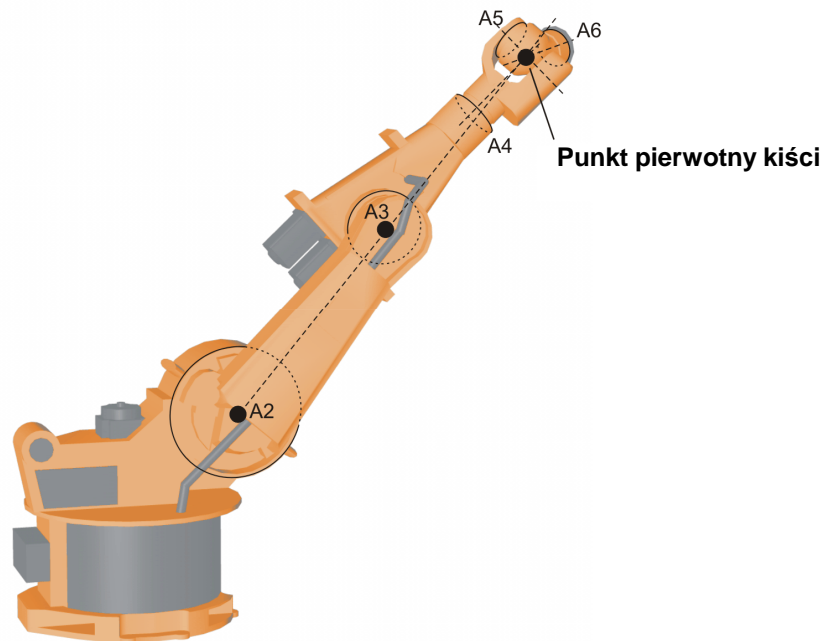


Punkt pierwotny kiści znajduje się w punkcie przecięcia osi A4, A5 i A6, na osi 1.

6

## Osobliwości manipulatorów KUKA typu 6R

### Osobliwość pozycji wyprostowanej

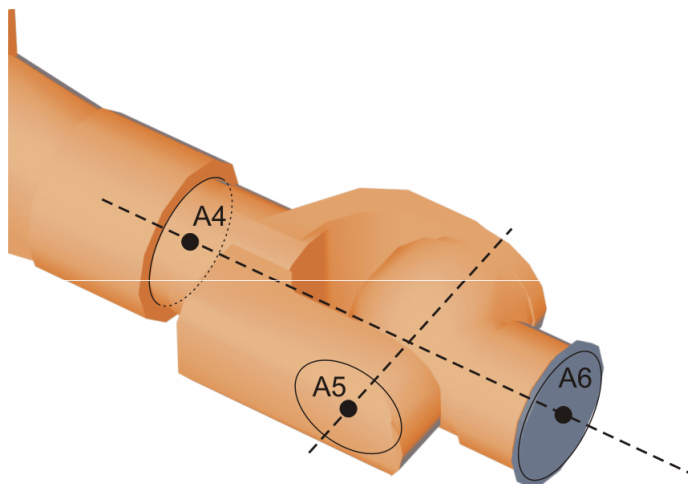


Przedłużenie odcinka A2-A3 przecina punkt pierwotny kiści.

7

## Osobliwości manipulatorów KUKA typu 6R

### Osobliwość osi kiści



W tym przypadku osie 4 i 6 są równoległe względem siebie. Pozycja obu osi nie jest jednoznacznie określona przez kinematykę odwrotną, gdyż istnieje nieskończenie wiele pozycji dla A4 i A6.

8

## Podstawowe elementy systemu robota KUKA

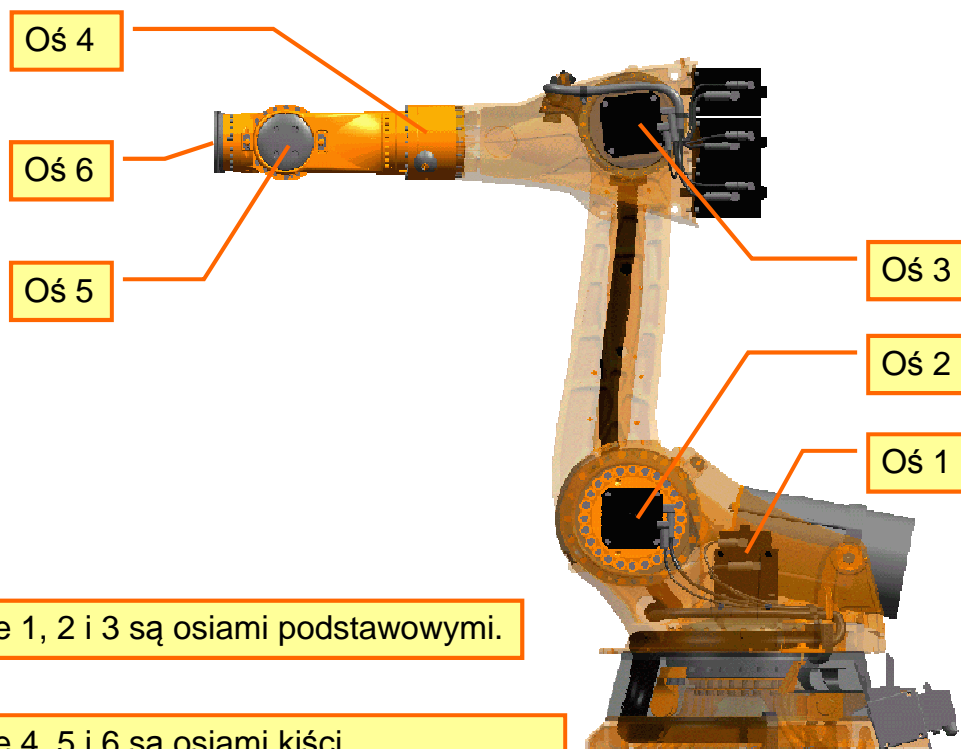
- Ramię manipulacyjne typowo o sześciu stopniach swobody
- Sterownik robota KR C4 / KR C4 compact
- Ręczny panel operatora-programisty KUKA smart PAD



KR AGILUS

9

## Ramię robota KUKA

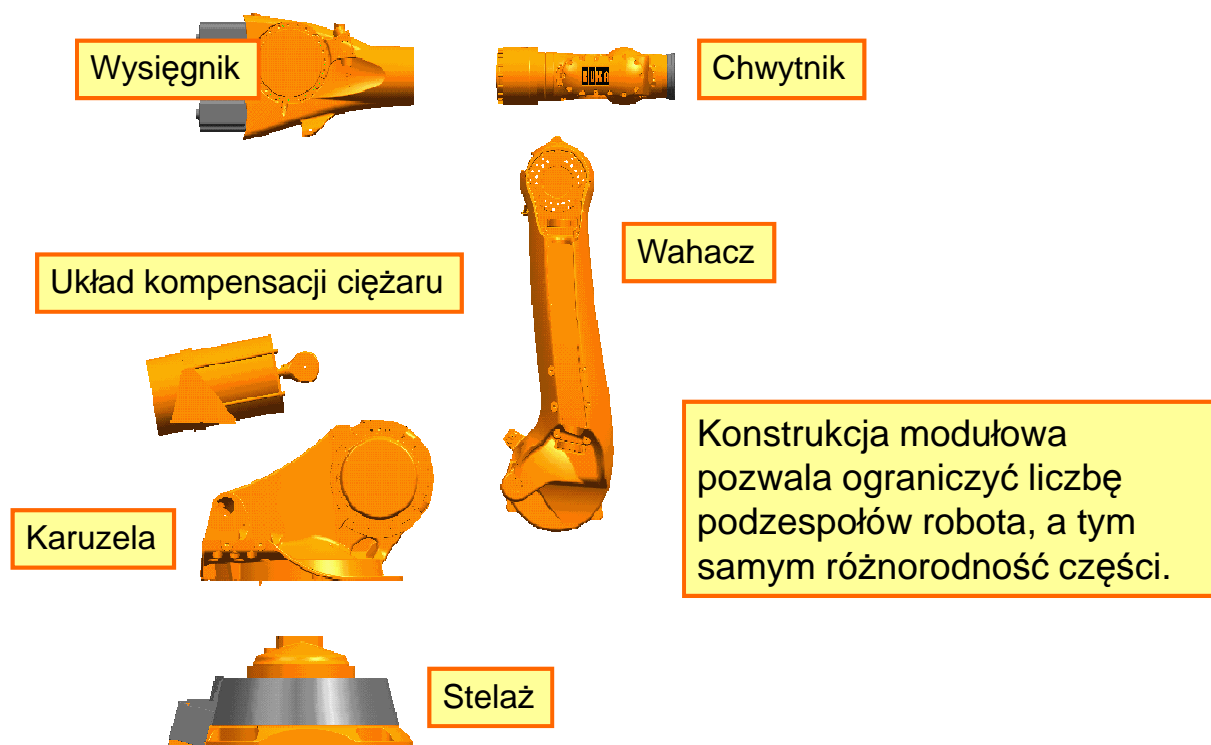


Osie 1, 2 i 3 są osiami podstawowymi.

Osie 4, 5 i 6 są osiami kiści.

10

## Ramię robota KUKA – konstrukcja mechaniczna



11

## Szafa sterownicza KR C4



12

# Szafa sterownicza KR C4

## Interfejsy komunikacyjne

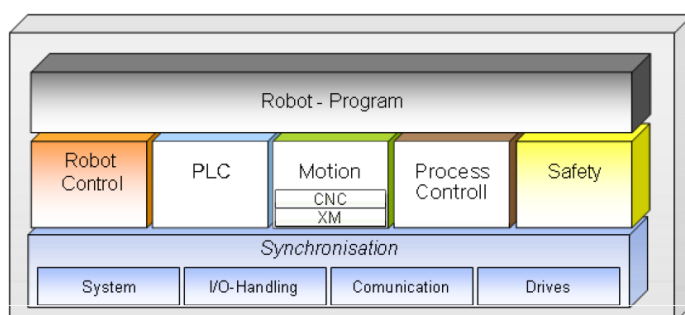


1. Zintegrowane wejścia/wyjścia cyfrowe
2. Magistrale przemysłowe
  - PROFINET
  - PROFIBUS
  - INTERBUS
  - ETHERNET IP
  - Can bus/DeviceNet
3. Ethernet

13

## Moduły i magistrale systemu KR C4

### Główne bloki systemu



#### Moduły związane z systemem sterowania

- RC (Robot Control) – jądro systemu sterownika robota
- Safety – zintegrowany sterownik bezpieczeństwa

#### Moduły opcjonalne dla użytkownika

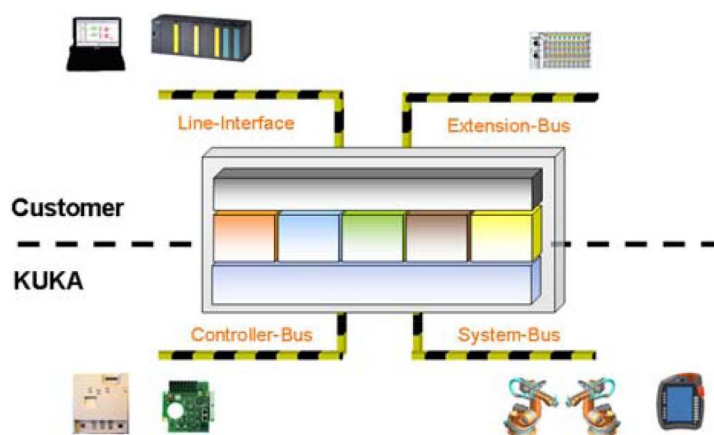
- PLC – SoftPLC do sterowania sekwencyjnego ogólnego przeznaczenia
- Motion – dodatkowe moduły rozszerzeń dla biblioteki KUKA MotionControl
- Process Control – platforma do integracji dodatkowych systemów np. wizyjnych

14



## Moduły i magistrale systemu KR C4

Cztery magistrale systemu oparte o architekturę Ethernet



**KCB** (KUKA Controller Bus) – połączenie z układami obwodów napędów

- RDC (Resolver Digital Converter)
- KPP (KUKA Power Pack)
- KSP (KUKA Servo Pack)
- EMD (Electronic Mastering Device)

15

## Moduły i magistrale systemu KR C4

**KSB** (KUKA System Bus) – do połączenia z

- smartPAD
- SIB (Safety Interface Board)
- Extended SIB
- RoboTeam
- Opcjonalnych modułów dodatkowych KUKA

**KEB** (KUKA Extension Bus) – połączenie z

- EtherCAT I/O
- Urządzenia pośredniczącymi do sieci PROFIBUS, DeviceNet itp.

**KLI** (KUKA Line Interface) – połączenie z

- Sterownikami PLC
- Magistralami PROFINET, Ethernet/IP
- Połączenie sieciowe poprzez protokół TCP/IP do archiwizacji danych, diagnostyki, z wirtualnym panelem, itp.

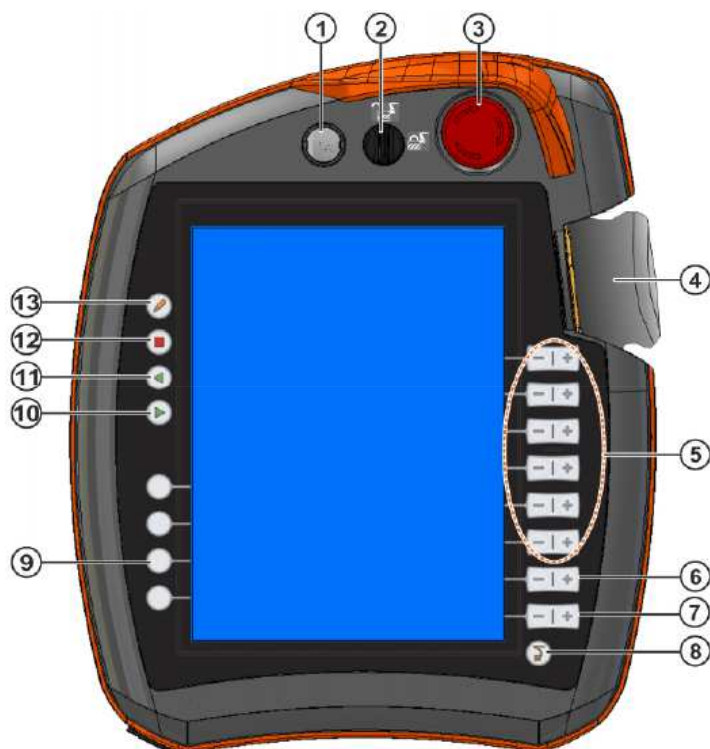
16



# Panel operatora-programisty KUKA smartPAD

Widok z przodu

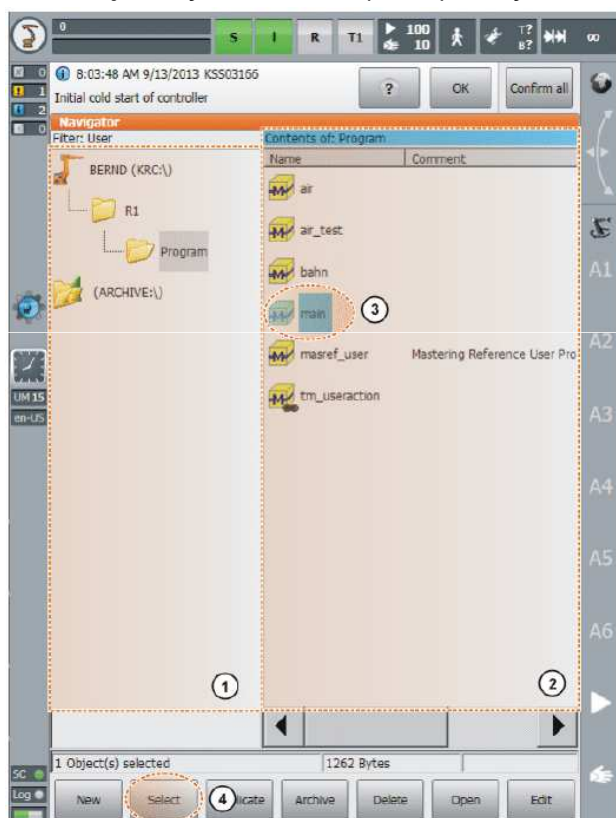
1. Przycisk odłączenia panelu smartPAD
2. Stacyjka otwierająca okno do wyboru trybu pracy
3. Przycisk Emergency STOP
4. Mysz 6D do manipulacji ręcznej
5. Przyciski do manipulacji ręcznej
6. Ustawianie prędkości programowej
7. Ustawianie prędkości manualnej
8. Przycisk wywołania Menu interfejsu użytkownika
9. Przyciski statusowe do obsługi pakietów technologicznych
10. Start programu
11. Start programu do tyłu
12. Stop programu
13. Przycisk do wywołania okna wirtualnej klawiatury



17

# Panel operatora-programisty KUKA smartPAD

Interfejs użytkownika (HMI) dotykowego panelu smartPAD



1. Okno nawigatora – struktura dysków i katalogów
2. Okno nawigatora – struktura wybranego katalogu i plików
3. Zaznaczony plik programu
4. Przycisk do wyboru programu do uruchomienia

18

# Panel operatora-programisty KUKA smartPAD

Widok od tyłu

- 1,3,5 - przyciski zezwolenia (trzypozycyjne)
- 2. Start programu
- 4. Gniazdo USB
- 6. Etykieta identyfikacyjna

- Wciśnięcie i przytrzymanie (w pozycji środkowej) jednego z przycisków zezwolenia umożliwia załączenie napędów robota w trybie T1 oraz T2 i wykonanie ruchu ręcznego lub testowanie programu (w połączeniu z przyciskiem Start)








19

## Komunikaty systemu KRC

Okno komunikatów – za pośrednictwem tego okna układ sterowania komunikuje się z operatorem. **Prawidłowe uruchomienie systemu wymaga potwierdzenia komunikatów z tym związanych!**



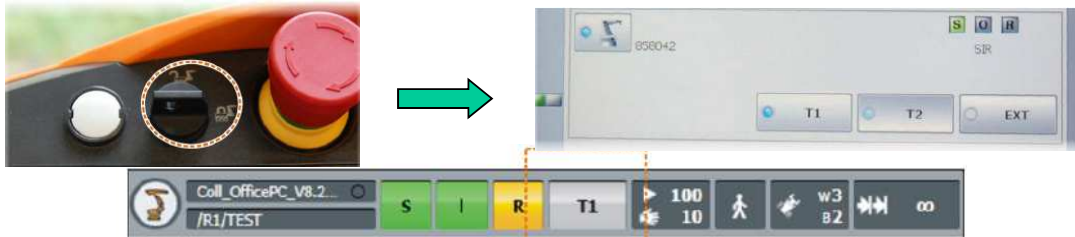
- 1 - okno komunikatów
- 2 - liczniki komunikatów

-  Informacja, status - np. "WYŁĄCZENIE AWARYJNE"
-  Wskazówka - np. "Klawisz START konieczny"
-  Potwierdzenie - np. "Potwierdź WYŁĄCZENIE AWARYJNE"
-  Oczekiwanie - np. "Wait for \$IN[1]==True "
-  Dialog - np. "Czy rzeczywiście nadpisać punkt?"

20

# Tryby pracy robotów KUKA

Cztery tryby pracy T1, T2, AUT i AUT EXT



Tryb T1 (Manual Reduced Velocity)

- tryb do ręcznej manipulacji, uczenia, programowania i testowania
- prędkość programowa i w ruchu ręcznym ograniczona do 250mm/s

Tryb T2 (Manual High Velocity)

- tryb przeznaczony do testowania programów
- prędkość realizacji ruchów odpowiada zaprogramowanej prędkości
- sterowanie ręczne niemożliwe

Tryb AUT (Automatic)

- tryb pracy automatycznej bez zewnętrznych sterowników nadrzędnych
- prędkość realizacji ruchów odpowiada zaprogramowanej prędkości
- sterowanie ręczne niemożliwe

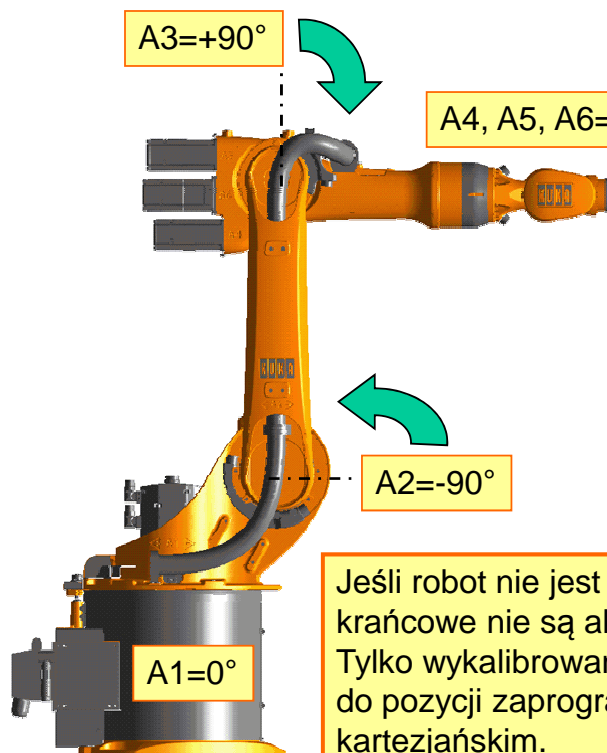
Tryb AUT EXT (Automatic External)

- tryb do współpracy robotów z nadrzędnym sterownikiem np. PLC
- prędkość realizacji ruchów odpowiada zaprogramowanej prędkości
- sterowanie ręczne niemożliwe

21

## Kalibracja robota KUKA

Cel kalibracji



- Podczas kalibracji robota osie ustawiane są w zdefiniowanym położeniu mechanicznym, tzw. **mechanicznym położeniu zerowym**.
- Gdy robot znajduje się w mechanicznym położeniu zerowym, zapisana zostaje bezwzględna wartość czujnika dla każdej osi.

Jeśli robot nie jest skalibrowany, programowe wyłączniki krańcowe nie są aktywne. Tylko wykalibrowany robot może wykonywać przesuw do pozycji zaprogramowanych i poruszać się w układzie kartezjańskim.

22

# Kalibracja robota KUKA

## Środki umożliwiające przeprowadzenie kalibracji



Elektroniczny czujnik pomiarowy (EMT)

Podczas kalibracji z użyciem EMT układ sterowania robota automatycznie ustawia mechaniczną pozycję zerową. W przypadku stosowania czujnika zegarowego należy tego dokonać przy pomocy osiowego przesuwu ręcznego.

W celu umożliwienia dokładnego ustawienia w mechanicznym punkcie zerowym, stosuje się czujnik zegarowy lub elektroniczny czujnik pomiarowy (EMT).



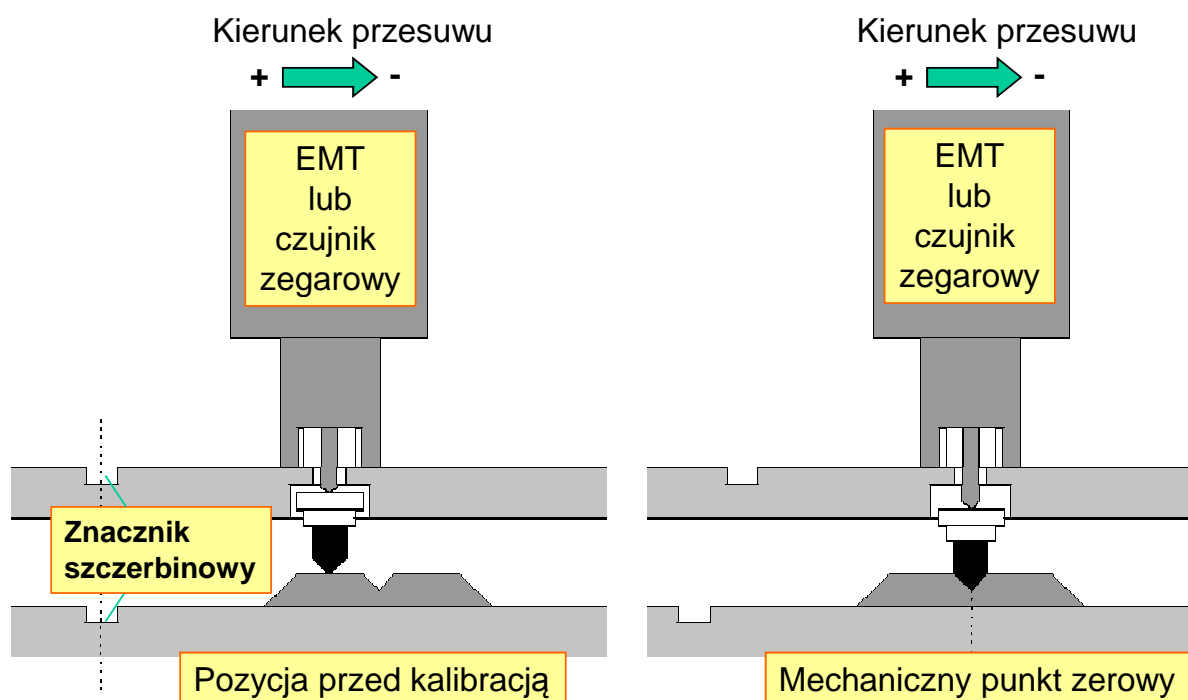
Czujnik zegarowy

23

# Kalibracja robota KUKA

Wybór kalibracji    Start-up>Master>EMD>Standard>Set Mastering

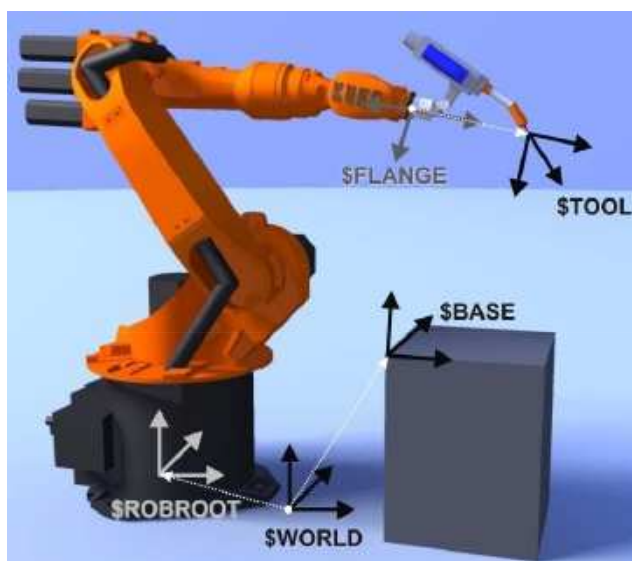
## Schemat przesuwu osi podczas kalibracji



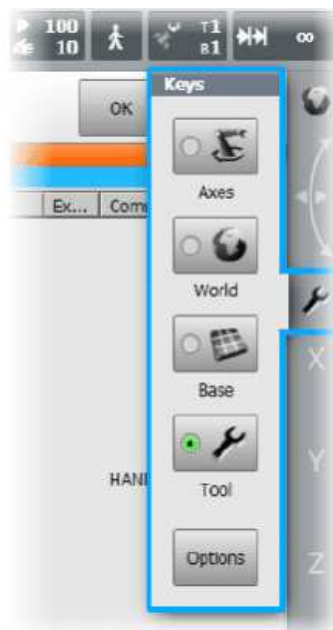
24

# Układy współrzędnych w systemie robota KUKA

## Układy kartezjańskie



- WORLD | world coordinate system
- ROBROOT | robot base coordinate system
- BASE | base coordinate system
- FLANGE | flange coordinate system
- TOOL | tool coordinate system



25

# Układy współrzędnych w systemie robota KUKA

## Układy do manipulacji robotem



- **Układ współrzędnych konfiguracyjnych - przesuw osiowy**  
Każda oś robota może być przesuwana pojedynczo w kierunku dodatnim lub ujemnym.



- **Układ współrzędnych uniwersalnych (WORLD)**  
Stacjonarny, prostokątny układ współrzędnych z początkiem typowo przy podstawie robota.



- **Układ współrzędnych TOOL**  
Prostokątny układ współrzędnych z początkiem w wybranym punkcie narzędzia.



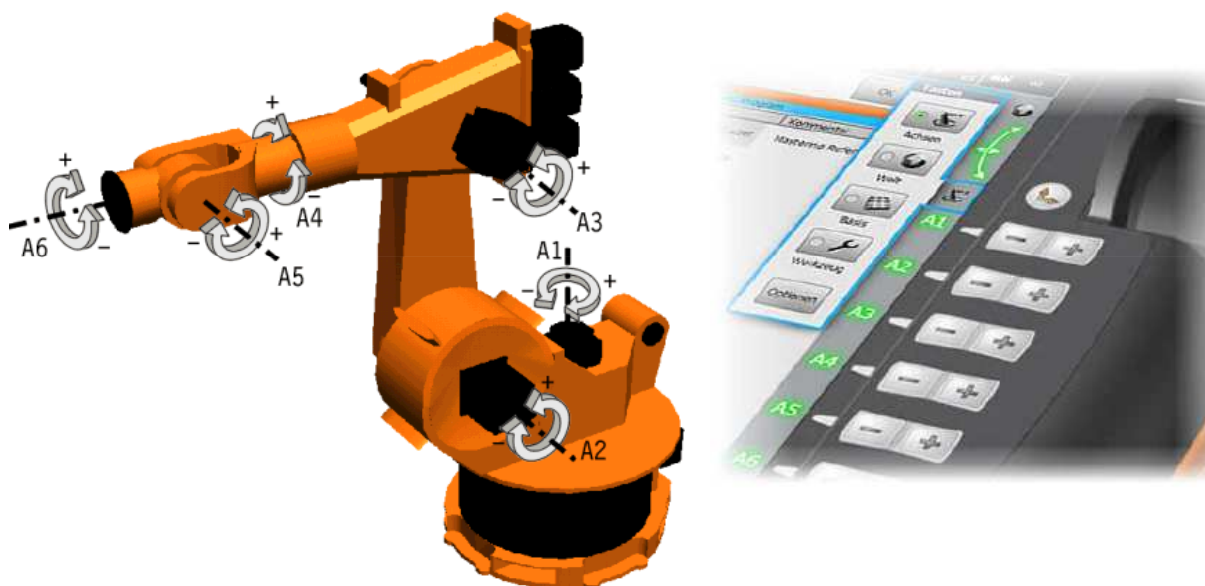
- **Układ współrzędnych BASE**  
Prostokątny układ współrzędnych z początkiem znajdującym się na przedmiocie przeznaczonym do obróbki.

26



## Układy współrzędnych w systemie robota KUKA

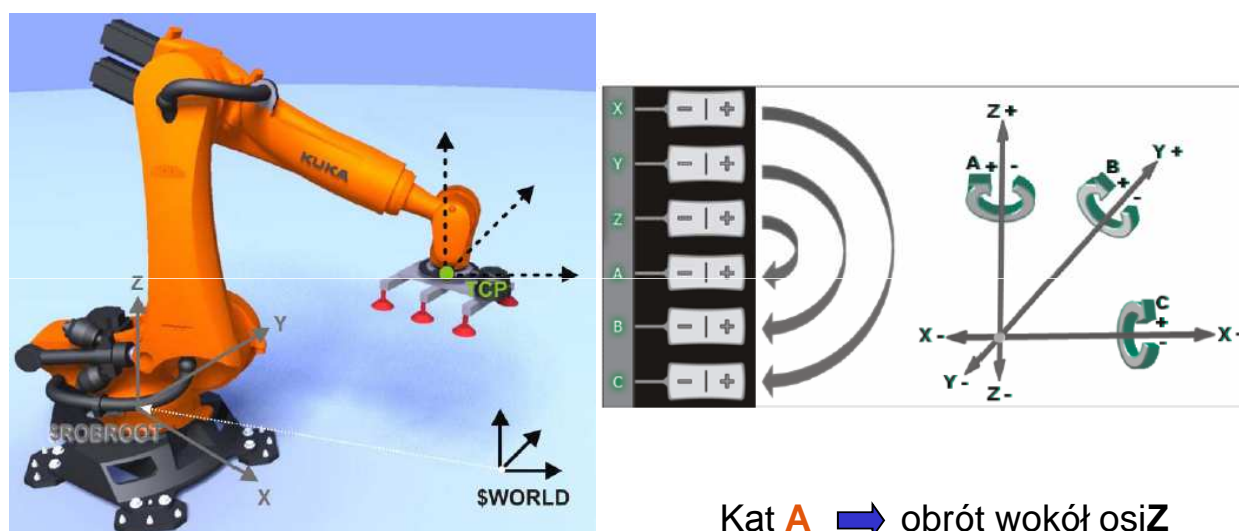
### Ręczny przesuw osiowy we współrzędnych konfiguracyjnych



27

## Układy współrzędnych w systemie robota KUKA

### Przesuw we współrzędnych kartezjańskich w układzie stacjonarnym (WORLD)

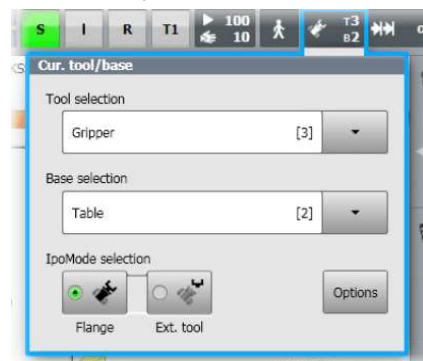
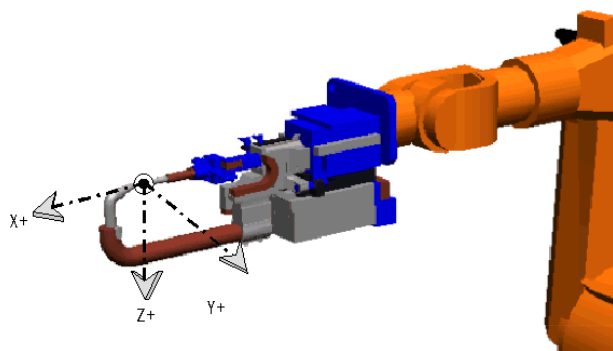


- Kąt **A** ➡ obrót wokół osi **Z**
- Kąt **B** ➡ obrót wokół osi **Y**
- Kąt **C** ➡ obrót wokół osi **X**

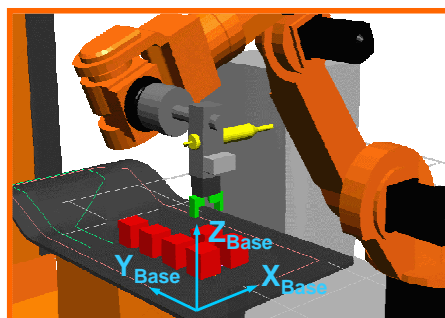
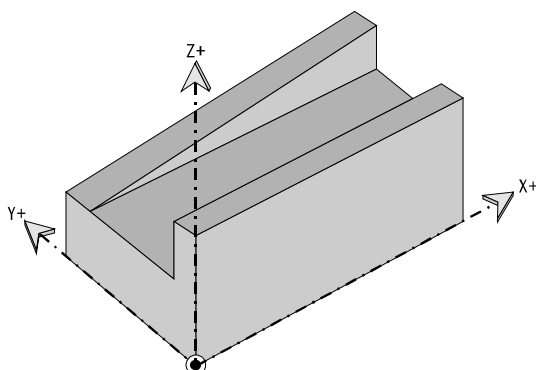
28

# Układy współrzędnych w systemie robota KUKA

## Przesuw we współrzędnych kartezjańskich w układzie narzędzia (TOOL)



## Przesuw we współrzędnych kartezjańskich w układzie zadania (BASE)

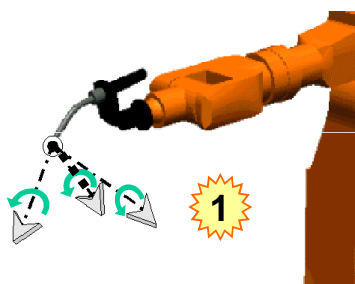


29

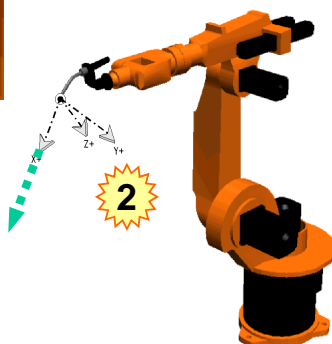
## Pomiar układów narzędzi roboczych (TOOL)

Celem pomiaru układu związanego z narzędziem TCP (Tool Center Point) jest zdefiniowanie **kartezjańskiego układu współrzędnych**, którego początek znajduje się w punkcie ustalonym przez użytkownika. W systemie może być zdefiniowanych 16 układów narzędzi TOOL.

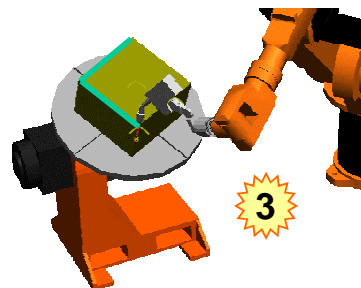
Korzyści wynikające z pomiaru narzędzia:



Początek układu i orientacja osi



Kierunek uderzenia



Prowadzenie końcówki narzędzia z zadaną prędkością po zdefiniowanym torze

30



# Pomiar układów narzędzi roboczych (TOOL)

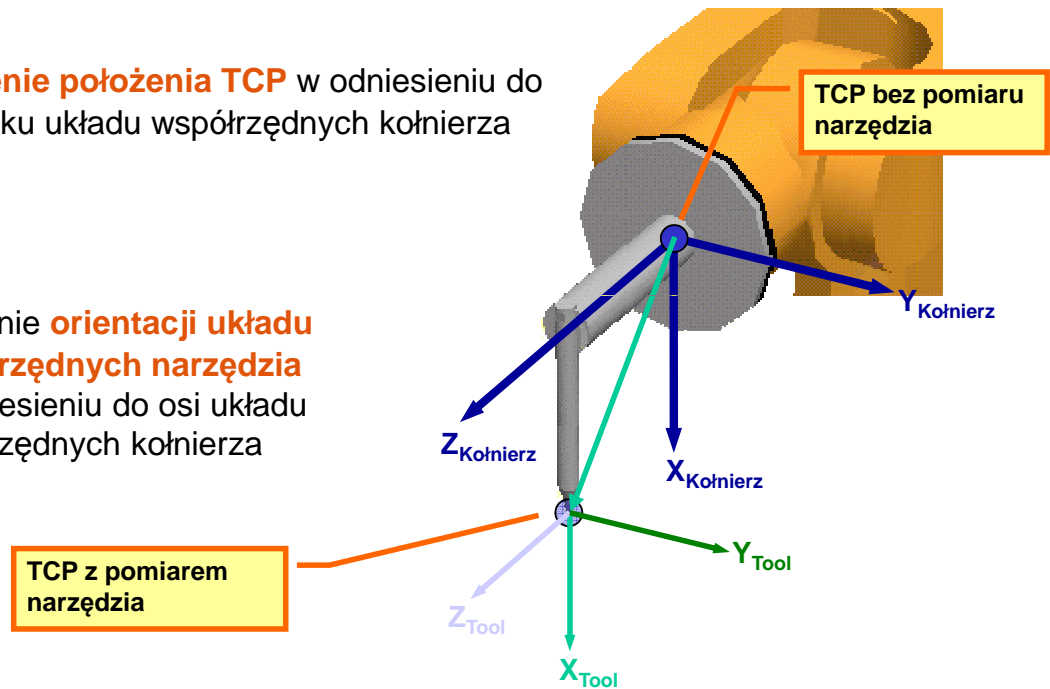
## Ogólny przebieg pomiaru narzędzia

### 1. krok:

Ustalenie **położenia TCP** w odniesieniu do początku układu współrzędnych kołnierza

### 2. krok:

Ustalenie **orientacji układu współrzędnych narzędzia** w odniesieniu do osi układu współrzędnych kołnierza



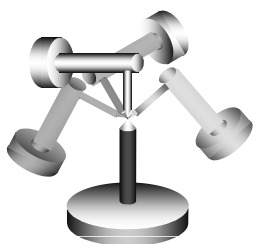
31

# Pomiar układów narzędzi roboczych (TOOL)

## Metody pomiaru układu narzędzia

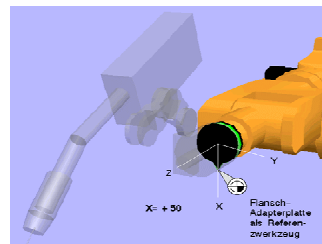
Start-Up>Calibrate>Tool>...

### 1. Pomiar TCP



XYZ 4-punktowy

lub



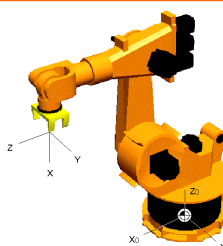
XYZ referencyjny

### 2. Pomiar orientacji



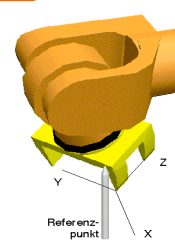
ABC uniwersalny 5D

lub



ABC uniwersalny 6D

lub



ABC 2-punktowy

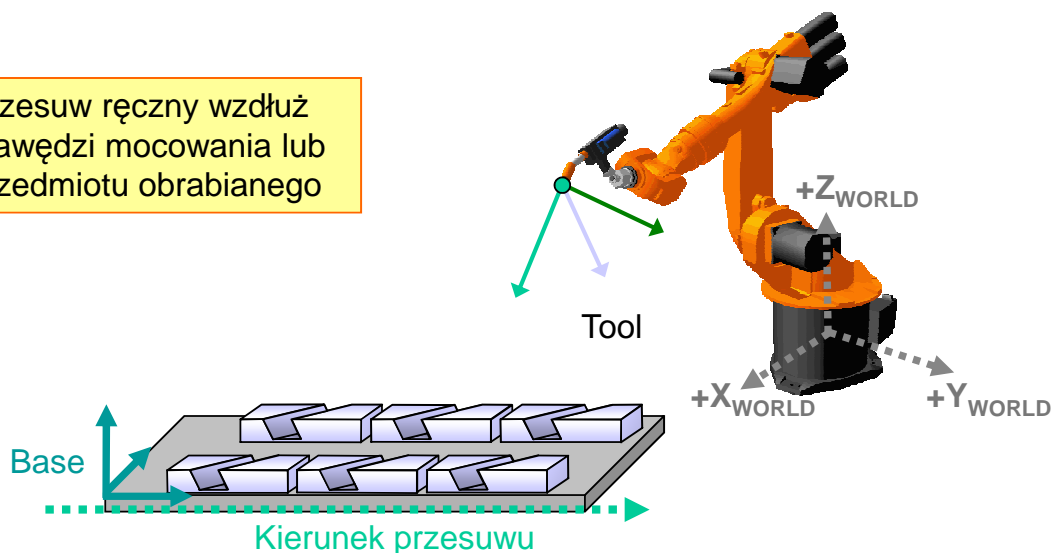
32

## Pomiar układów bazowych (BASE)

Celem pomiaru układu bazowego związanego z realizowanym zadaniem jest zdefiniowanie **kartezjańskiego układu współrzędnych**, którego początek znajduje się w punkcie odniesienia ustalonym przez użytkownika. W systemie mogą być zdefiniowane 32 układy bazowe BASE.

Korzyści wynikające z pomiaru narzędzia:

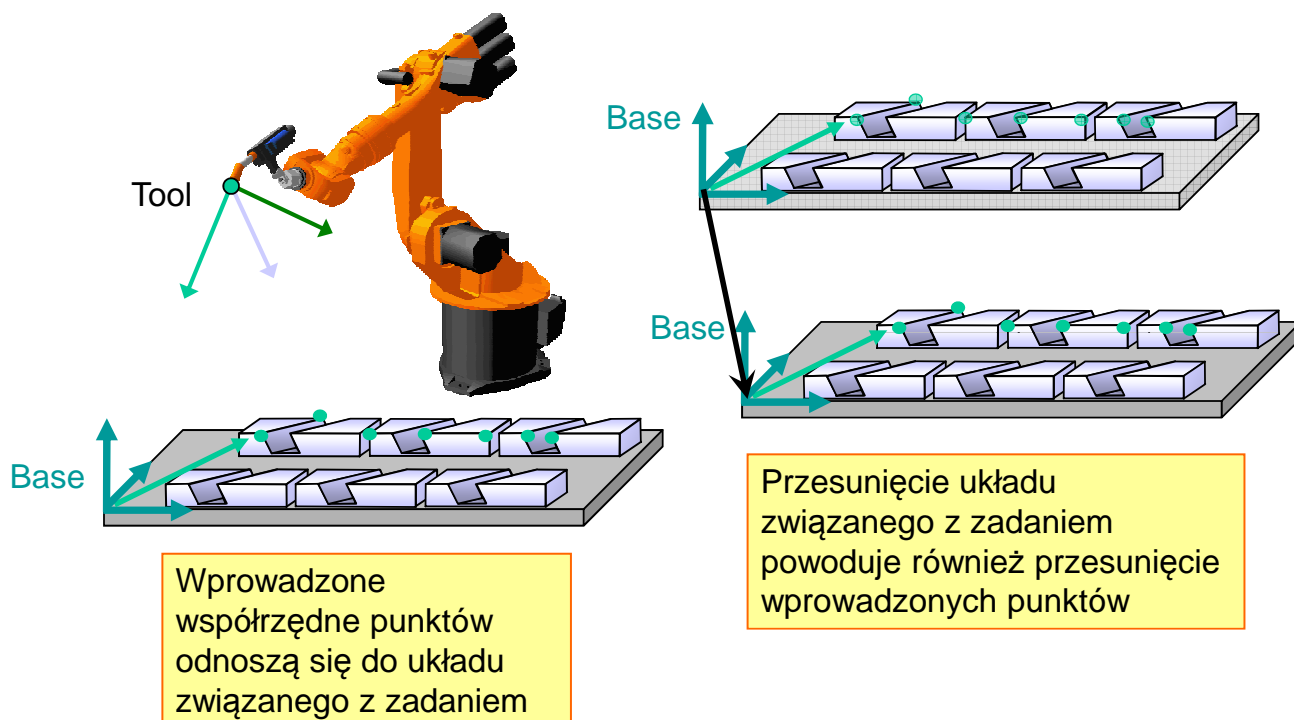
Przesuw ręczny wzdłuż krawędzi mocowania lub przedmiotu obrabianego



33

## Pomiar układów bazowych (BASE)

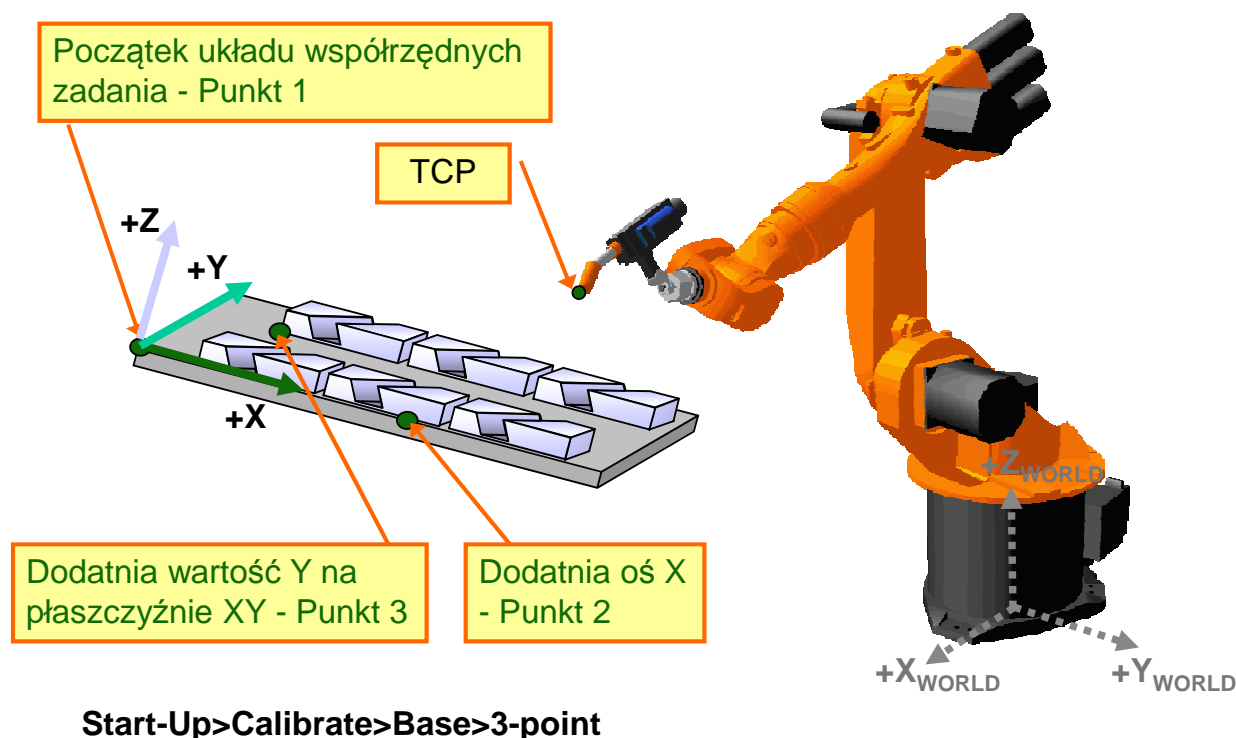
Korzyści wynikające z pomiaru narzędzia:



34

## Pomiar układów bazowych (BASE)

Metoda 3-punktowa pomiaru układu BASE związanego z zadaniem



35

## Utworzenie i wybór programu

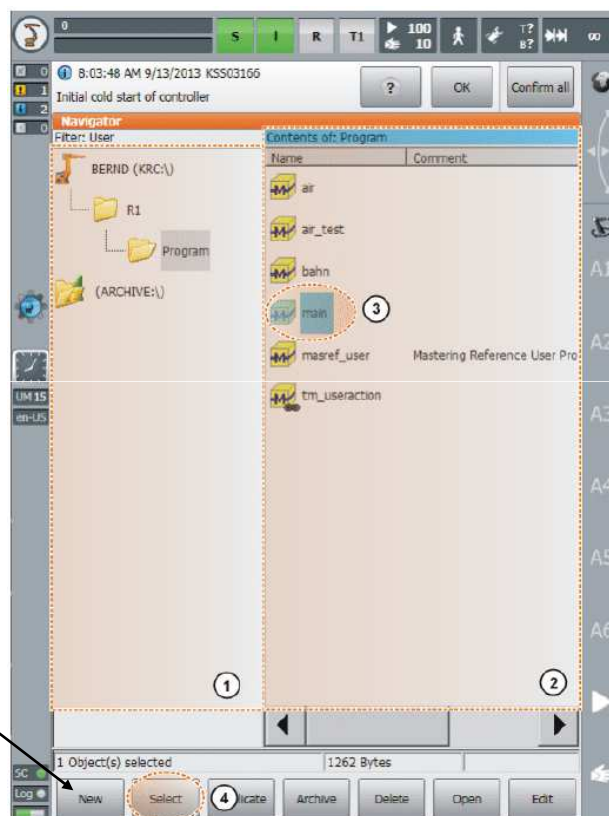
### Okno nawigatora

1. Okno nawigatora – struktura dysków i katalogów
2. Okno nawigatora – struktura katalogu
3. Zaznaczony moduł programu
4. Przycisk wyboru programu do uruchomienia

Edycja i testowanie programów – dwa tryby:

- Otwarcie programu do uruchomienia (poprzez Select) – umożliwia programowanie ruchów z formularzy i jednocześnie testowanie programu (widoczny wskaźnik programu).
- Otwarcie programu tylko do edycji (poprzez Open) – umożliwia programowanie ruchów z formularzy oraz pisanie i edycję programu w języku KRL (tryb eksperta).

Przycisk utworzenia nowego programu



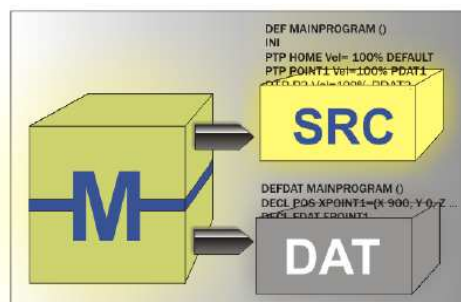
36

# Utworzenie i wybór programu

## Moduł programu

### 1. Kod programu SRC

```
DEF MAINPROGRAM ()  
INI  
PTP HOME Vel= 100% DEFAULT  
PTP POINT1 Vel=100% PDAT1 TOOL[1] BASE[2]  
PTP P2 Vel=100% PDAT2 TOOL[1] BASE[2]  
...  
END
```



### 2. Lista danych DAT

```
DEF DAT MAINPROGRAM ()  
DECL E6POS XPOINT1=(X 900, Y 0, Z 800, A 0, B 0, C 0, S 6, T 27, E1  
0, E2 0, E3 0, E4 0, E5 0, E6 0)  
DECL F DAT FPOINT1 ...  
...  
ENDDAT
```

Możliwe operacje za pomocą przycisków programowych:

- Duplicate/Copy
- Delete
- Rename

37

# Uruchamianie i testowanie programu

## Tryby wykonywania programu



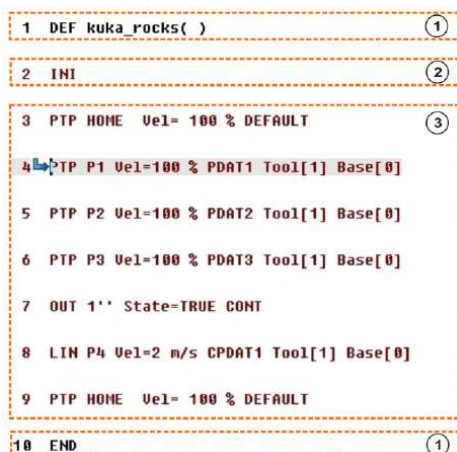
**GO** – program jest wykonywany w sposób ciągły aż do jego końca.



**STEP** – program jest wykonywany krokowo, każde polecenie (formularz, blok instrukcji w postaci zakładki) jest wykonywane osobno i wymaga po zakończeniu ruchu ponownego wciśnięcia przycisku Start.



**SINGLE STEP** – program jest wykonywany krokowo w sposób inkrementalny, linia po linii niezależnie od ich zawartości.  
Po wykonaniu każdej linii programu wymagane jest ponowne wciśnięcie przycisku Start.  
Tryb dostępny tylko dla użytkownika Expert !



## Struktura programu

1. Nagłówek programu oraz koniec programu
2. Sekcja inicjalizacji wewnętrznych zmiennych i parametrów programu (musi być wykonana)
3. Sekcja programu z poleceniami ruchu, obsługą we/wy itp. Zalecane pierwsze i ostatnie polecenie to ruch do bezpiecznej pozycji HOME (widoczny wskaźnik programu)

38

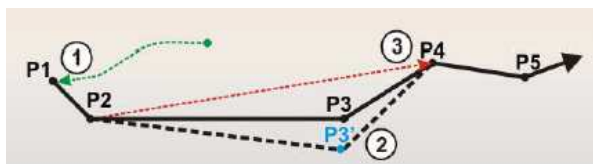
# Uruchamianie i testowanie programu

## Ruch BCO w programie

- **BCO** (Block **C**oincidence) oznacza zgranie zdarzeń w czasie/przestrzeni.
- Ruch BCO jest wymagany w celu zapewnienia zgodności bieżącej konfiguracji robota z pozycją wynikającą z wybranego bloku programu. Planowanie ścieżki jest tylko możliwe, gdy bieżąca pozycja robota jest zgodna z pozycją zaprogramowaną – pierwszy ruch w programie musi być PTP.
- Ruch BCO jest realizowany w przypadku wyboru programu, restartu (Reset) programu, pierwszy ruch przy testowaniu programu, po modyfikacji programu lub wyboru nowego bloku programu.
- Ruch BCO zawsze realizowany jest z ograniczoną prędkością.

Przykłady ruchu BCO:

1. wybór lub restart programu
2. modyfikacja poleceń ruchu
3. wybór bloku programu



39

# Uruchamianie i testowanie programu

## Ikona opisująca stan programu

|  |  |
|--|--|
|  | Program nie jest wybrany do uruchomienia                                     |
|  | Wskaźnik programu znajduje się na pierwszej linii programu (typowo blok INI) |
|  | Wybrany program jest uruchomiony   |
|  | Wybrany i uruchomiony program został zatrzymany                              |
|  | Program się zakończył, wskaźnik programu znajduje się na jego końcu          |

## Ustawianie prędkości



40

# Uruchamianie programu

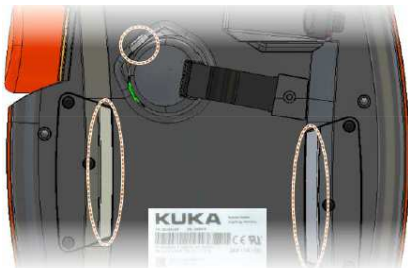
## Procedura uruchomienia programu

1. Wybór programu



2. Ustawienie prędkości programowej POV

3. Wciśnięcie przycisku zezwolenia na ruch



4. Wykonanie poleceń sekcji INI oraz realizacja pierwszego ruchu BCO (przycisk Start)



5. Osiągnięcie pierwszej pozycji i zatrzymanie robota – komunikat „Programmed path reached (BCO)”

6. Uruchomienie programu

Tryby T1 i T2 – przycisk Start

Tryb AUT – wymaga załączenia napędów



41

# Programowanie ruchów robota KUKA

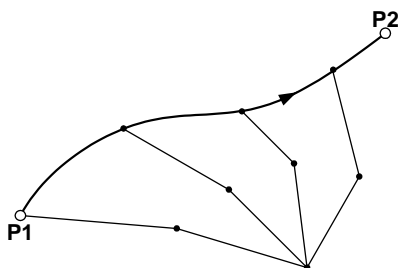
## Rodzaje ruchów (interpolacji ruchu)

Ruch  
osiowy

Ruchy  
odnoszące się do toru (CP), SPLINE

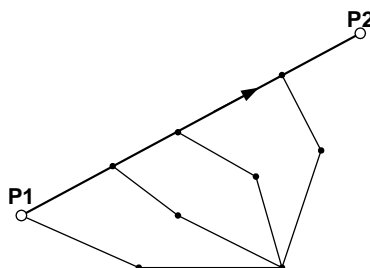
### PTP

(od punktu do punktu):  
Narzędzie przesuwane  
jest po najszybszym  
torze do punktu  
końcowego.



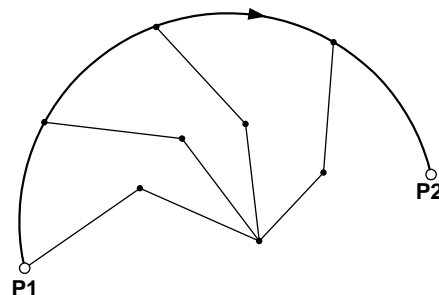
### LIN (ruch liniowy):

Prowadzenie narzędzia ze  
zdefiniowaną prędkością  
po najkrótszym torze  
(prostej).



### CIRC (ruch kołowy):

Prowadzenie narzędzia  
ze zdefiniowaną  
prędkością po torze  
kołowym.

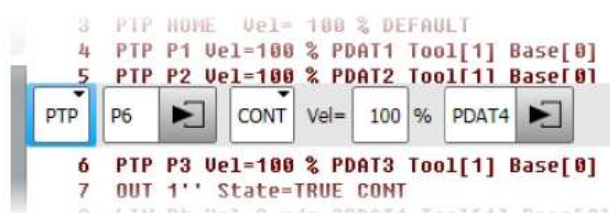


42

# Programowanie ruchów robota KUKA

## Formularze do wprowadzanie poleceń ruchu

```
3 PTP HOME Vel= 100 % DEFAULT
4 PTP P1 Vel=100 % PDAT1 Tool[1] Base[0]
5 PTP P2 Vel=100 % PDAT2 Tool[1] Base[0]
```

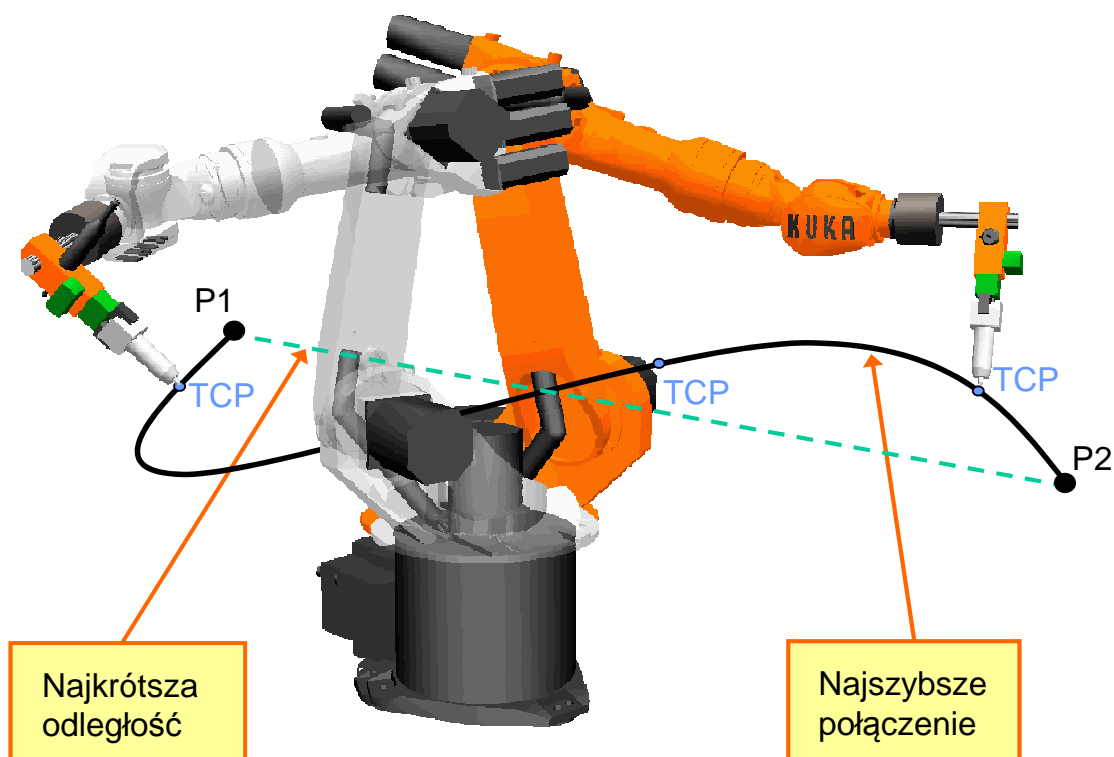


1. Typ ruchu PTP, LIN lub CIRC
2. Nazwa punktu nadawana automatycznie (można wprowadzić własną), wybranie pola otwiera okno edycji układów współrzędnych
3. Pole wyboru ruchu dokładnego (puste) lub przybliżonego (CONT)
4. Prędkość ruchu wyrażona w:
  - 1..100 % dla ruchów PTP względem ustawionej prędkości POV
  - 0,001..2 m/s dla ruchów typu CP
5. Pole parametrów ruchu: otwiera okno do ustawienia przyspieszenia, odległości przybliżania i sposobu prowadzenia narzędzia

43

# Programowanie ruchów robota KUKA

## Ruch od punktu do punktu PTP



44



# Programowanie ruchów robota KUKA

## Ruch PTP

### Commands>Motion>PTP

PTP P1 CONT Vel= 100 % PDAT1

1 Tool: TOOL\_DATA[1] 2 Base: BASE\_DATA[1]

3 External TCP: False 4 Collision detection: False

Frames

1. Wybór układu narzędzia [1]-[16].
2. Wybór układu bazowego [1]-[32]
3. Wybór zewnętrznego nieruchomego narzędzia:
  - False: narzędzie zamontowane na kołnierzu (ruchome)
  - True: narzędzie nieruchome umieszczone w otoczeniu robota
4. Detekcja kolizji za pomocą obliczania momentów w poszczególnych osiach
  - True: momenty są obliczane
  - False: momenty nie są obliczane, detekcja kolizji niemożliwa

45

# Programowanie ruchów robota KUKA

## Ruch PTP

1 Acceleration: 100 [%] 2 Approximation distance: 100 [mm]

Motion parameter

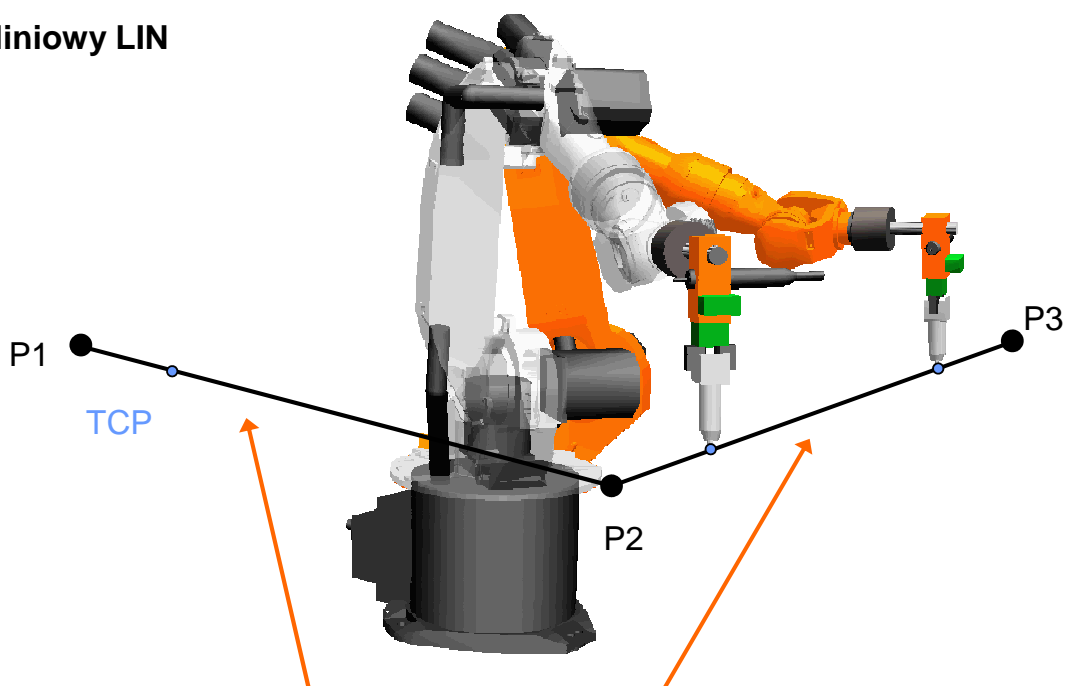
1. Przyspieszenie ruchu w % 1..100
2. Gdy wybrany jest parametr ruchu przybliżonego CONT można ustawić odległość przybliżania w zależności od konfiguracji ruchów:
  - %: 1..100
  - mm: 1..1000

Zakończenie edycji formularza ruchu - **CmdOK**

46

# Programowanie ruchów robota KUKA

## Ruch liniowy LIN

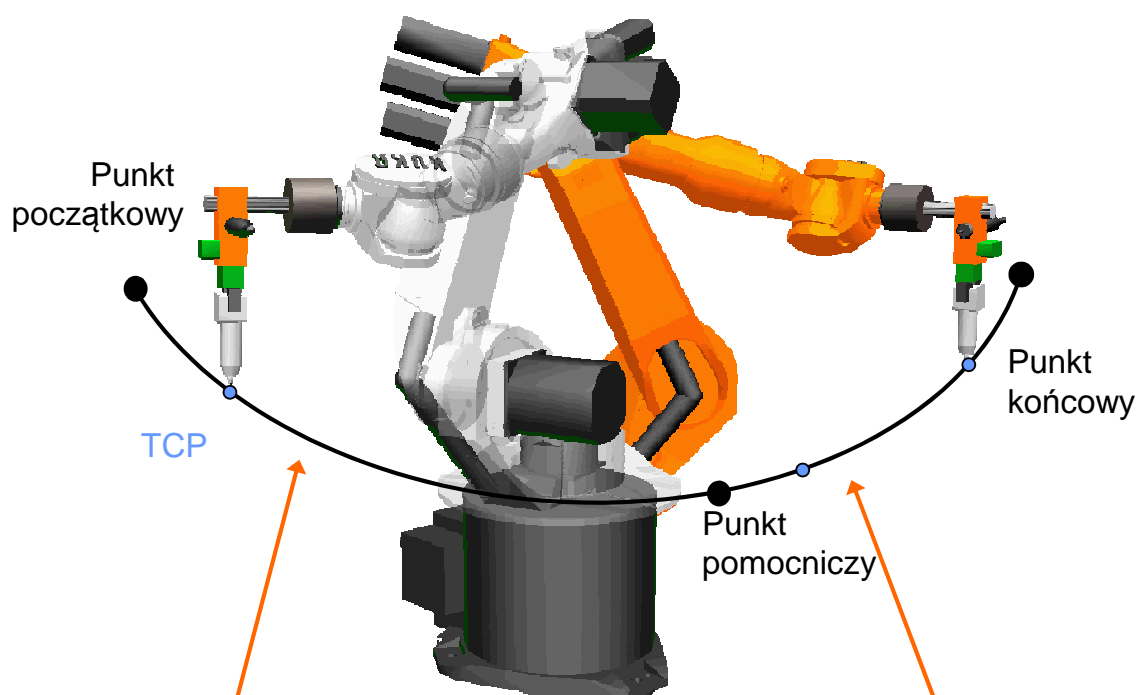


TCP poruszany jest po najkrótszym torze do punktu końcowego

47

# Programowanie ruchów robota KUKA

## Ruch po łuku okręgu CIRC



TCP przesuwany jest po łuku kołowym do punktu końcowego

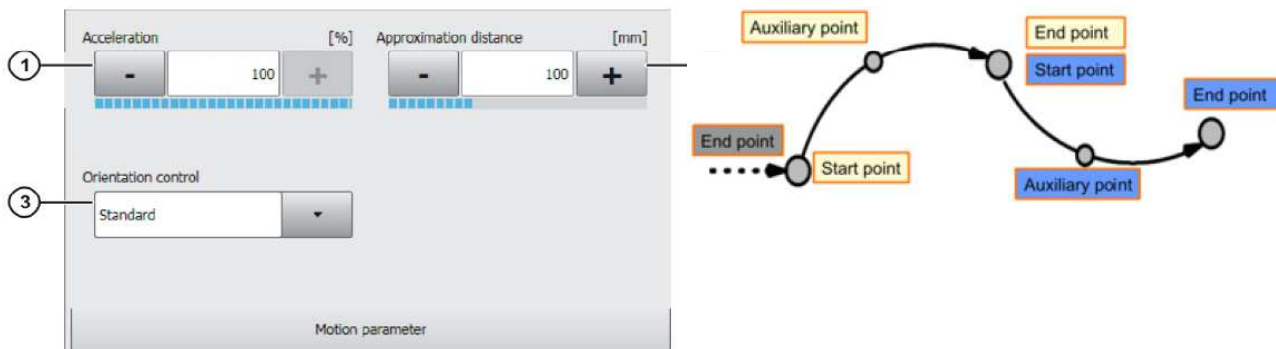
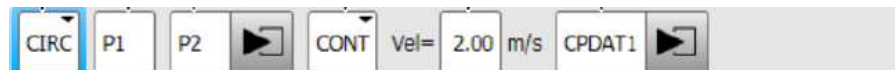
48

# Programowanie ruchów robota KUKA

## Ruch liniowy LIN Commands>Motion>LIN



## Ruch kołowy CIRC Commands>Motion>CIRC



3. Wybór sposobu prowadzenia (orientowania) narzędzia: Standard, Wrist PTP, Constant Orientation

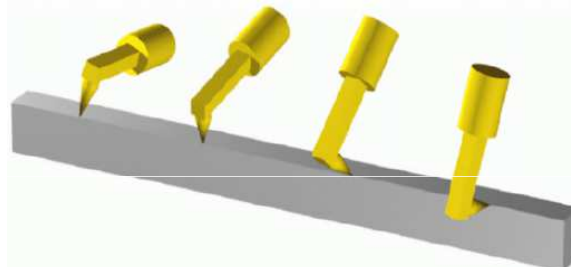
49

# Programowanie ruchów robota KUKA

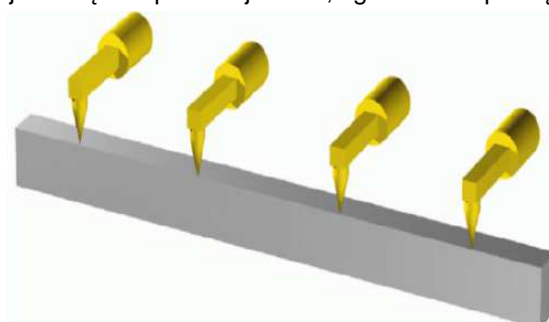
## Prowadzenie narzędzia w ruchu liniowym LIN

**Standard** – standardowe prowadzenie narzędzia, orientacja zmienia się w sposób ciągły podczas ruchu.

**Wrist PTP** – orientacja również zmienia się w sposób ciągły, zmiana orientacji następuje przez liniową zmianę kątów osi nadgarstka (uniknięcie problemów osobliwości nadgarstka).



**Constant** – orientacja narzędzia pozostaje stała, zgodnie z zapamiętaną w punkcie początkowym.

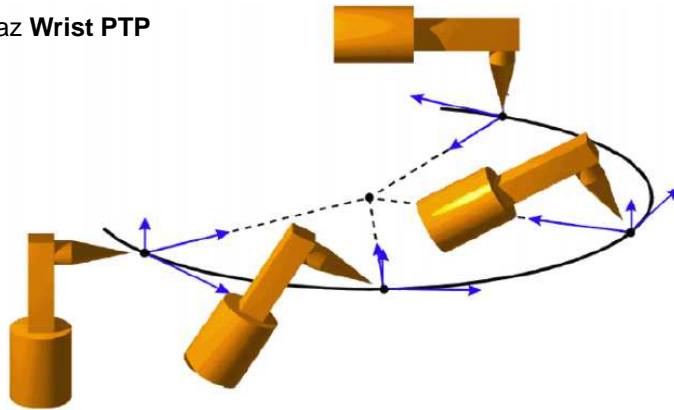


50

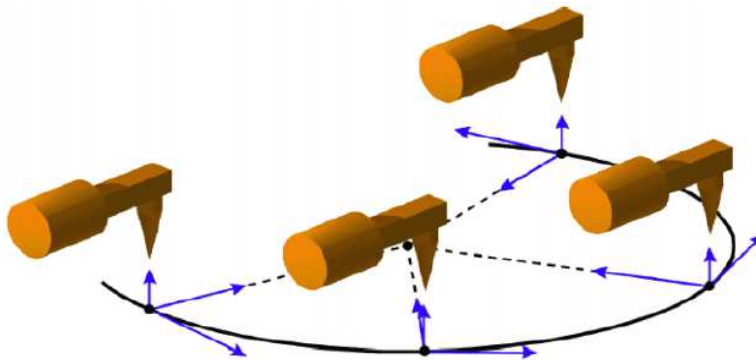
# Programowanie ruchów robota KUKA

## Prowadzenie narzędzia w ruchu kołowym CIRC

Standard oraz Wrist PTP



Constant



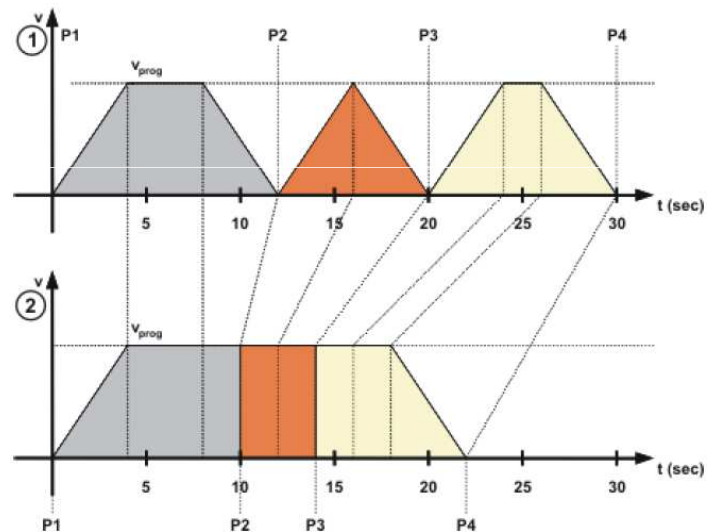
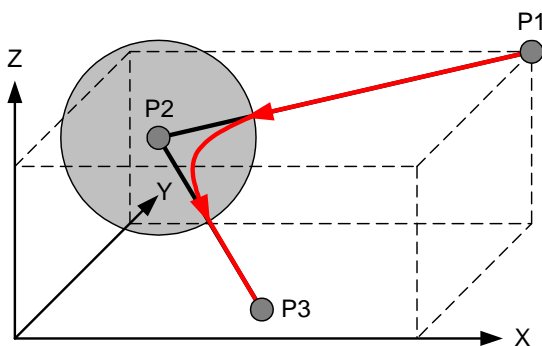
51

# Programowanie ruchów robota KUKA

## Przybliżanie ruchów

Podczas przybliżania brak jest precyzyjnego dojazdu do punktów a robot nie jest hamowany.

- Zalety:
- oznacza to mniejsze zużycie mechanizmów i energii
  - krótsze czasy cykli



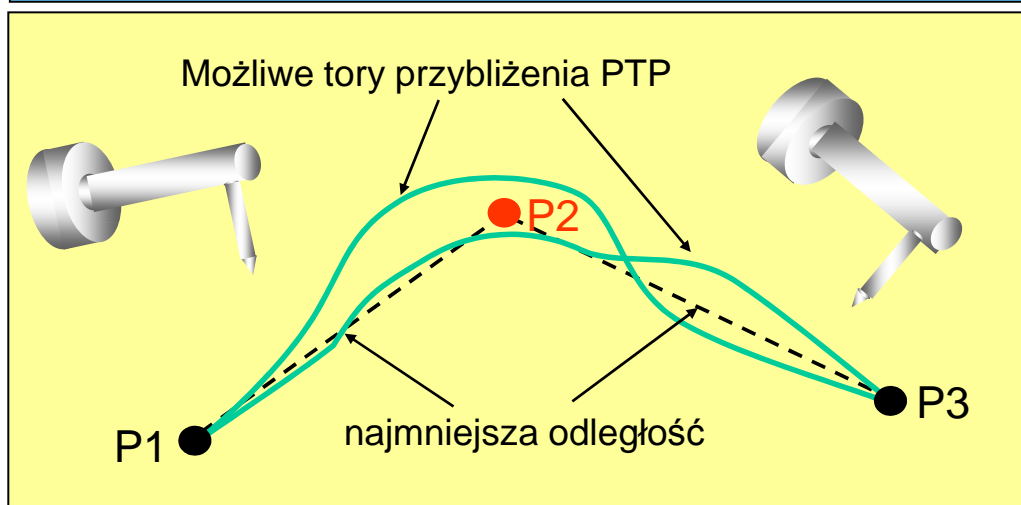
52

# Programowanie ruchów robota KUKA

## Przybliżanie ruchów PTP

Ruch PTP  
z przybliżeniem

→ P2 to  
punkt przybliżenia



Wartość odległości przybliżenia (wyrażona w %) zadaje wielkość obszaru przybliżenia. Trajektoria **nie** może być ustawiana przez operatora i jest również **nieprzewidywalna**!

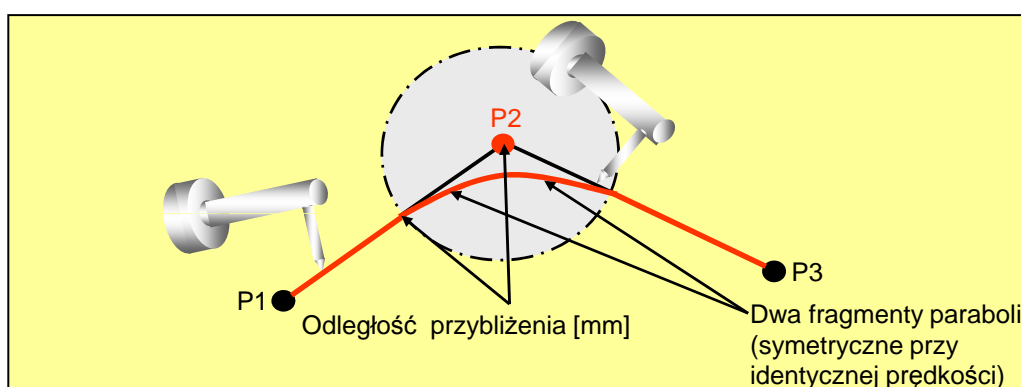
53

# Programowanie ruchów robota KUKA

## Przybliżanie ruchów LIN

Ruch LIN  
z przybliżeniem

→ P2 to  
punkt przybliżenia



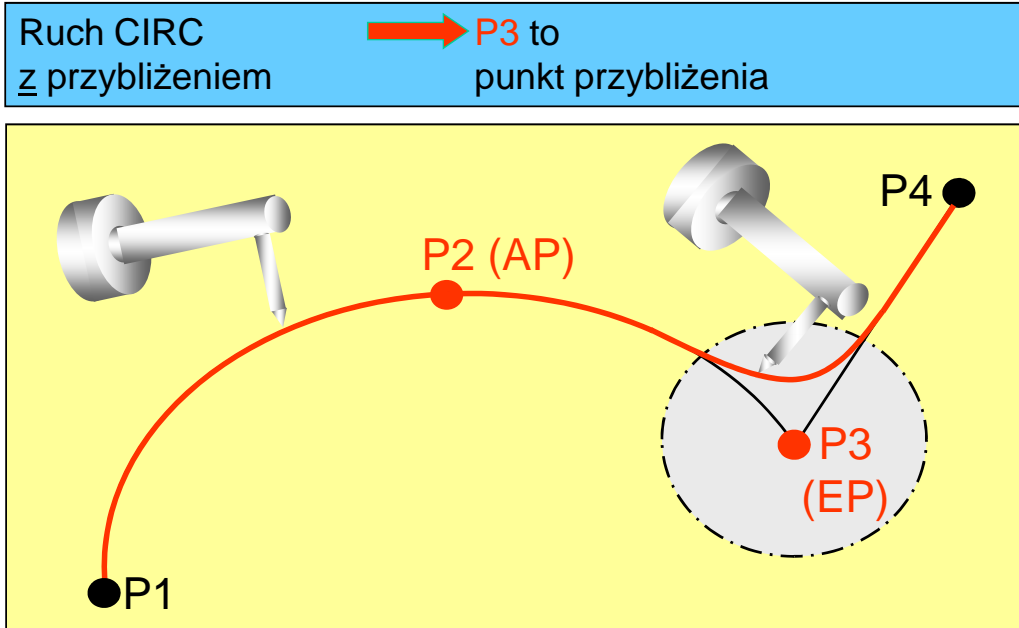
Wartość odległości przybliżania (wyrażona w mm) definiuje odległość pomiędzy punktem końcowym a punktem, w którym zaczyna się ruch przybliżony.

Wynikająca z tego trajektoria **nie** jest łukiem okręgu. Dotyczy to również komendy ruchu CIRC.

54

# Programowanie ruchów robota KUKA

## Przybliżanie ruchów CIRC



Uzyskana trajektoria ruchu przybliżonego **nie** jest łukiem okręgu, podobnie jak w ruchu LIN (połączenie dwóch parabol).

55

# Programowanie ruchów robota KUKA

## Przebieg programu w przód

### Czym jest przebieg programu w przód:

Widoczny na ekranie podczas wykonywania programu **licznik przebiegu głównego** (biały pasek) zawsze pokazuje rekord, który jest właśnie wykonywany. **Licznik przebiegu w przód** nie jest widoczny i **znajduje się** o trzy rekordy ruchowe (ustawienie standardowe) **przed** licznikiem przebiegu głównego.

### Jaką funkcję spełnia licznik przebiegu w przód:

Aby móc obliczyć trajektorię, np. ruchu przybliżonego, konieczne jest wczytywanie danych planowania toru poprzez licznik przebiegu w przód. Przetwarzane są nie tylko dane ruchu, lecz również instrukcje arytmetyczne i sterujące peryferiami.

### Co wpływa na licznik przebiegu w przód:

Instrukcje i dane, które mają wpływ na peryferia (np. instrukcje wejścia/wyjścia) powodują zatrzymanie przebiegu w przód. Jeśli licznik przebiegu w przód zostanie zatrzymany, nie można wykonać przybliżenia.

56

## Programowanie logiki robota KUKA

Odczytywanie wejść i ustawianie wyjść służy do komunikacji układu sterowania robota z otaczającymi urządzeniami peryferyjnymi (np. narzędzia, elementy wykonawcze, czujniki itd.)



Układ sterowania robota



Peryferia

- Funkcje oczekiwania
- Funkcje przełączające
- Funkcje przełączające zależne od czasu i odległości

57

## Programowanie logiki robota KUKA

Funkcja oczekiwania zależna od czasu WAIT

Commands>Logic>WAIT

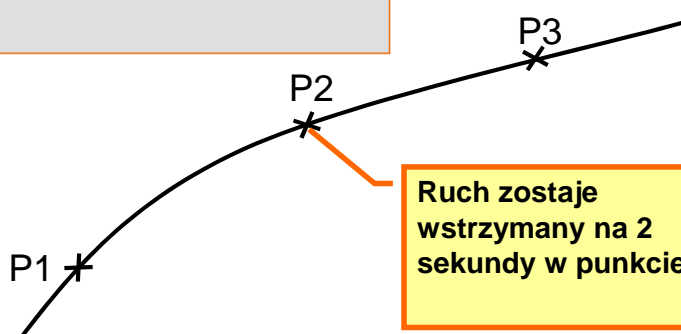
Po wybraniu "WAIT", można wprowadzić czas oczekiwania. Komenda ta powoduje zawsze wstrzymanie przebiegu, również przy wartości 0 sekund.



Czas oczekiwania w sekundach

Przykład:

```
PTP P1 Vel=100% PDAT1  
PTP P2 Vel=100% PDAT2  
WAIT Time=2 sec  
PTP P3 Vel=100% PDAT3
```



Ruch zostaje wstrzymany na 2 sekundy w punkcie P2.

58



# Programowanie logiki robota KUKA

## Funkcja oczekiwania zależna od sygnału WAIT FOR

Commands>Logic>WAIT FOR

1. Dodanie zewnętrznej operacji logicznej AND, OR, EXOR z możliwością negacji (wybór przez odpowiednie przyciski menu)
2. Dodanie wewnętrznej operacji logicznej AND, OR, EXOR z możliwością negacji (wybór przez odpowiednie przyciski menu).
3. Sygnał na który funkcja oczekuje – IN, OUT, CYCFLAG, TIMER, FLAG
4. Numer sygnału 1-4096
5. Nazwa sygnału (o ile istnieje i tylko dla trybu EXPERT)
6. Sposób realizacji oczekiwania:
  - CONT: oczekiwanie realizowane w przedbiegu programu
  - puste pole: zatrzymanie przedbiegu i oczekiwanie po realizacji wcześniejszej linii programu

Wprowadzone w funkcji oczekiwania wyrażenie logiczne zwraca TRUE (wartość 1) lub FALSE (wartość 0)

59

# Programowanie logiki robota KUKA

## Funkcja sterująca wyjściami OUT

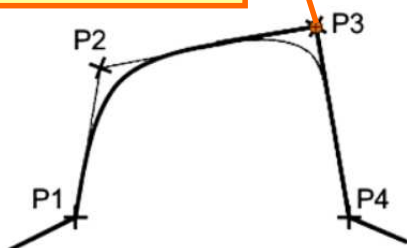
Commands>Logic>Out>OUT

1. Numer wyjścia 1..4096
2. Nazwa sygnału (tylko dla trybu EXPERT)
3. Stan: TRUE lub FALSE
4. Sposób realizacji ustawiania:
  - CONT: ustawienie w przedbiegu programu
  - puste pole: zatrzymanie przedbiegu i ustawienie po realizacji wcześniejszej linii programu

```

LIN P1 Vel=0.2 m/s CPDAT1
LIN P2 CONT Vel=0.2 m/s CPDAT2
LIN P3 CONT Vel=0.2 m/s CPDAT3
OUT 5 'rob_ready' State=TRUE
LIN P4 Vel=0.2 m/s CPDAT4
    
```

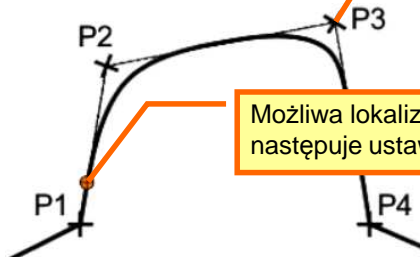
Punkt w którym następuje ustawienie wyjście 5



```

LIN P1 Vel=0.2 m/s CPDAT1
LIN P2 CONT Vel=0.2 m/s CPDAT2
LIN P3 CONT Vel=0.2 m/s CPDAT3
OUT 5 'rob_ready' State=TRUE CONT
LIN P4 Vel=0.2 m/s CPDAT4
    
```

Punkt P3 jest przybliżony



Możliwa lokalizacja w której następuje ustawienie wyjście 5

60

# Programowanie logiki robota KUKA

## Funkcja sterująca wyjściami PULSE

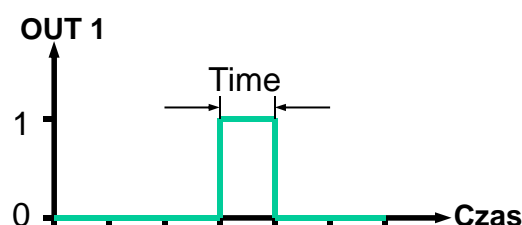
Commands>Logic>Out>PULSE

1 2 3 4 5

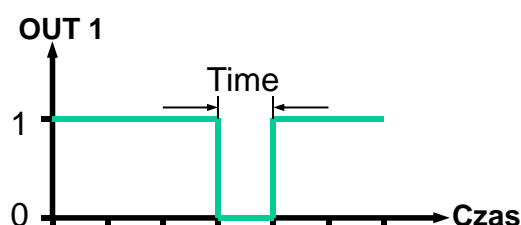
PULSE 12 test\_output State= TRUE CONT Time= 0.15 sec

1. Numer wyjścia 1..4096
2. Nazwa sygnału (tylko dla trybu EXPERT)
3. Stan: TRUE (poziom wysoki) lub FALSE (poziom niski)
4. Sposób realizacji ustawiania:
  - CONT: ustawienie w przedbiegu programu
  - puste pole: zatrzymanie przedbiegu i ustawienie po realizacji wcześniejszej linii programu
5. Czas trwania impulsu 0,10 .. 3,00 s

Poziom "high" przełącza: **STATE=TRUE**



Poziom "low" przełącza: **STATE=FALSE**



61

# Programowanie logiki robota KUKA

## Funkcje sterujące wyjściami typu SYN

Commands>Logic>Out>SYN OUT

Commands>Logic>Out>SYN PULSE

Opcja **START/END**

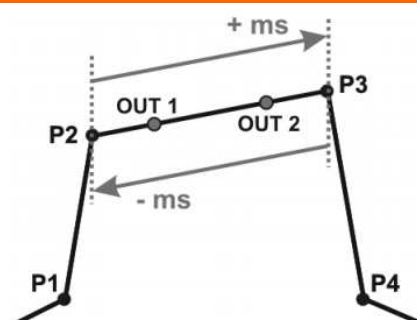
1 2 3 4 5

SYN OUT 9 SIGNAL 9 State= TRUE at START Delay= 20 ms

1. Numer wyjścia 1..4096
2. Nazwa sygnału (tylko dla trybu EXPERT)
3. Stan: TRUE lub FALSE
4. Opcja: PATH, START, END
  - **START**: przełączanie jest względem punktu startowego
  - **END**: przełączanie jest względem punktu końcowego bloku ruchu
5. Opóźnienie przełączenia -1000 .. 1000 ms

```

LIN P1 VEL=0.3m/s CPDAT1
LIN P2 VEL=0.3m/s CPDAT2
SYN OUT 1 '' State= TRUE at START Delay=20ms
SYN OUT 2 '' State= TRUE at END Delay=-20ms
LIN P3 VEL=0.3m/s CPDAT3
LIN P4 VEL=0.3m/s CPDAT4
    
```



62

# Programowanie logiki robota KUKA

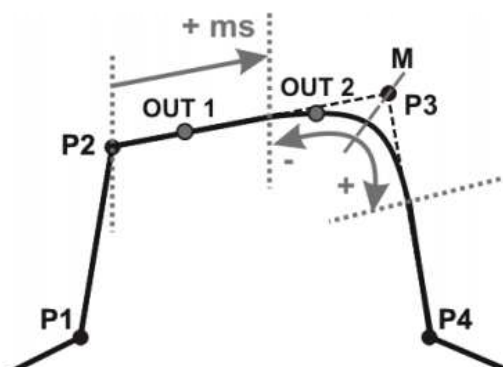
## Funkcje sterujące wyjściami typu SYN

Opcja **START/END**

Punkt końcowy P3 przybliżany

```

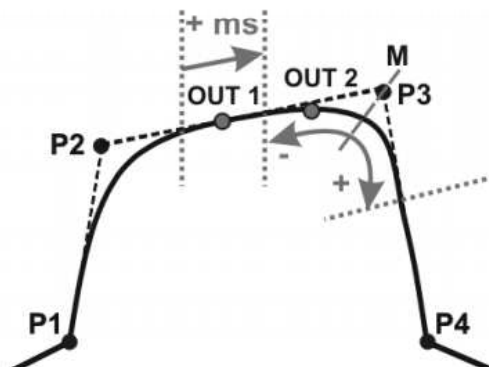
LIN P1 VEL=0.3m/s CPDAT1
LIN P2 VEL=0.3m/s CPDAT2
SYN OUT 1 '' State= TRUE at START Delay=20ms
SYN OUT 2 '' State= TRUE at END Delay=-20ms
LIN P3 CONT VEL=0.3m/s CPDAT3
LIN P4 VEL=0.3m/s CPDAT4
    
```



Punkt początkowy P2 i końcowy P3 przybliżany

```

LIN P1 VEL=0.3m/s CPDAT1
LIN P2 CONT VEL=0.3m/s CPDAT2
SYN OUT 1 '' State= TRUE at START Delay=20ms
SYN OUT 2 '' State= TRUE at END Delay=-20ms
LIN P3 CONT VEL=0.3m/s CPDAT3
LIN P4 VEL=0.3m/s CPDAT4
    
```

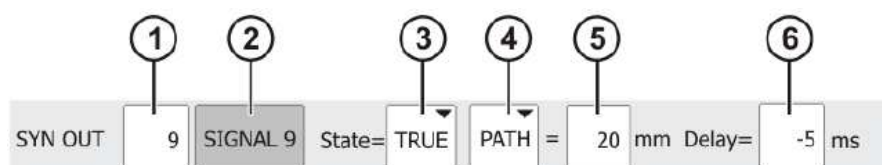


63

# Programowanie logiki robota KUKA

## Funkcje sterujące wyjściami typu SYN

Opcja **PATH**



1. Numer wyjścia 1..4096
2. Nazwa sygnału (tylko dla trybu EXPERT)
3. Stan: TRUE lub FALSE
4. Opcja: PATH, START, END
  - **PATH**: przełączenie jest realizowane względem punktu końcowego bloku ruchu
5. Odległość od punktu końcowego bloku ruchu -2000 .. 2000 mm
6. Opóźnienie załączenia -1000 .. 1000 ms

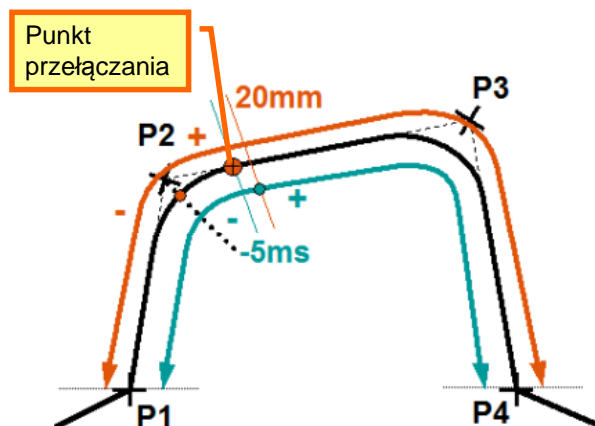
64

# Programowanie logiki robota KUKA

## Funkcje sterujące wyjściami typu SYN Opcja PATH

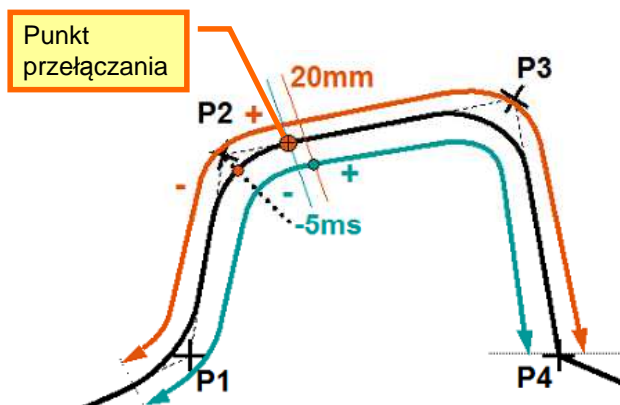
Punkt przełączania dla P2 i P3 przybliżanego

Ograniczenia przełączania stanu wyjścia



```

LIN P1 VEL=0.3m/s CPDAT1
SYN OUT 1 '' State= TRUE at START PATH=20mm Delay=-5ms
LIN P2 CONT VEL=0.3m/s CPDAT2
LIN P3 CONT VEL=0.3m/s CPDAT3
LIN P4 VEL=0.3m/s CPDAT4
    
```



```

LIN P1 CONT VEL=0.3m/s CPDAT1
SYN OUT 1 '' State= TRUE at START PATH=20mm Delay=-5ms
LIN P2 CONT VEL=0.3m/s CPDAT2
LIN P3 CONT VEL=0.3m/s CPDAT3
LIN P4 VEL=0.3m/s CPDAT4
    
```

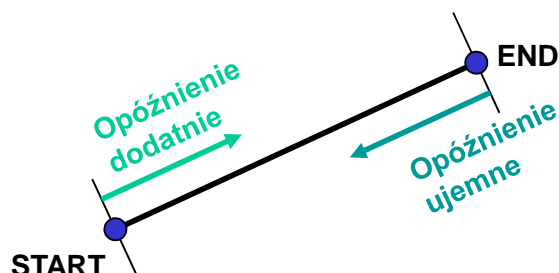
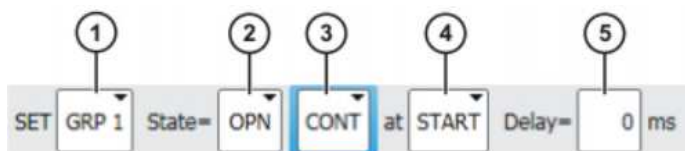
65

## Programowanie chwytaka robota KUKA

W systemie można skonfigurować do 16 chwytaków. Do konfiguracji dostępne jest funkcjonalnie 5 różnych typów chwytaków

### Commands>GripperTech>Gripper

Zmiana stanu chwytaka w ruchu przybliżonym

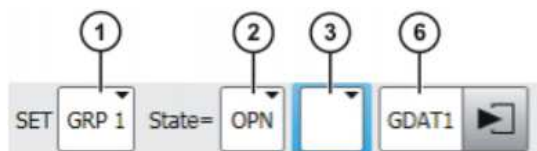


1. Nazwa chwytaka ze zbioru chwytaków zdefiniowanych w systemie
2. Wybór stanu chwytaka
3. Zmiana stanu w trakcie ruchu przybliżonego (**CONT**) lub po zatrzymaniu w punkcie (puste pole)
4. Opcja dostępna tylko w trybie **CONT**, punkt odniesienia
  - **START**: przełączenie jest realizowane względem punktu początkowego
  - **END**: przełączenie jest realizowane względem punktu końcowego
5. Opcja dostępna tylko w trybie **CONT**. Czas opóźnienia zmiany stanu chwytaka względem wybranego punktu -200 .. 200 ms

66

## Programowanie chwytaka robota KUKA

Zmiana stanu chwytaka w punkcie



6. Pole dostępne w trybie ruchu dokładnego, opcje dodatkowe kontroli stanu chwytaka

Kontrola stanu chwytaka w punkcie w trybie ruchu dokładnego (opcje pola nr 6)



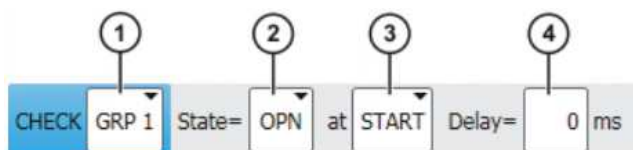
1. Czas opóźnienia po którym ruch jest wznowiany
2. Opcja:
  - **OFF**: brak kontroli stanu chwytaka (domyślne)
  - **ON**: kontrola stanu chwytaka poprzez badanie stanu wejścia skojarzonego z chwytakiem (kontynuacja programu robota nastąpi po ustawieniu wejścia zgodnie z ustawionym stanem chwytaka)

67

## Programowanie chwytaka robota KUKA

Kontrola stanu chwytaka

**Commands>GripperTech>Check Gripper**



1. Nazwa chwytaka ze zbioru chwytaków zdefiniowanych w systemie
2. Wybór stanu chwytaka do zbadania
3. Opcje:
  - **START**: sprawdzanie stanu w punkcie początkowym ruchu
  - **END**: sprawdzanie stanu w punkcie końcowym ruchu
4. Czas oczekiwania do zmiany stanu chwytaka względem wybranego punktu

Komenda jest tylko wtedy wykonywana, jeśli poprzedza ona rekord ruchowy.

Jeśli sprawdzany warunek nie jest spełniony następuje zatrzymanie robota.

68