

## 7. A linguagem robCOMM<sup>©</sup>

### 7.1. Fundamentos

As linguagens de programação, tal como as linguagens humanas, são definidas através de um conjunto de regras morfológicas, que determinam a sua estrutura, e semânticas que determinam o significado. São utilizadas para facilitar a comunicação homem-máquina, a manipulação de dados e a explicitação de algoritmos.

O objectivo deste trabalho passa, também, por desenvolver uma forma de controlar o comportamento do dispositivo robótico utilizado.

Pretendeu-se criar uma linguagem cujo conjunto de regras pudesse definir comportamentos do equipamento, tornando possível o seu controlo a partir de dispositivos remotos. Assim, a implementação do *robCOMM*, enquanto servidor, implicou também o desenvolvimento de uma linguagem que implementasse os serviços concebidos e possibilitasse a interacção de dispositivos remotos.

Foi, então, desenvolvida a *robCOMM language*, linguagem de implementação do servidor *robCOMM* [ Tabela 4 ].

<b>MOVTOCPOS</b>	Movimento cartesiano
<b>MOVTOJPOS</b>	Movimento em junta
<b>GETCRCPOS</b>	Posição actual do TCP
<b>GETCRJPOS</b>	Configuração actual das juntas
<b>CHECKCPOS</b>	Verifica posições cartesianas
<b>CHECKJPOS</b>	Verifica configurações de junta
<b>GETDIRKIN</b>	Calcula cinemática directa
<b>GETREVKIN</b>	Calcula cinemática inversa
<b>GETREG</b>	Obtêm valores de registos
<b>SETREG</b>	Escreve em registos
<b>SETPATH</b>	Compõe um caminho
<b>MOVTHPTH</b>	Move ao longo de um caminho
<b>LISTTPP</b>	Obtêm a lista de programas TP
<b>RUNTPP</b>	Executa programas TP
<b>GETDIO</b>	Devolve o estado de I/O digitais
<b>GETAIO</b>	Devolve o estado de I/O analógicas
<b>SETDIO</b>	Actua I/O digitais
<b>SETAIO</b>	Actua I/O analógicas
<b>SETTOOLFRM</b>	Especifica uma Tool frame
<b>SETUSERFRM</b>	Especifica um User frame
<b>SETLSPEED</b>	Especifica velocidade linear
<b>MOTIONSTOP</b>	Para o movimento
<b>STOPSERV</b>	Termina o servidor

Tabela 4 – Resumo de intruções robCOMM

## 7.2. Instruções disponíveis

### 7.2.1. MOVTOCPOS e MOVTOJPOS

Sempre que se pensa num manipulador, pensa-se em movimento, fazendo do conjunto de instruções disponíveis, estas como as instruções mais emblemáticas deste trabalho.

Permitem instruir o servidor a efectuar movimentos de tipo variado

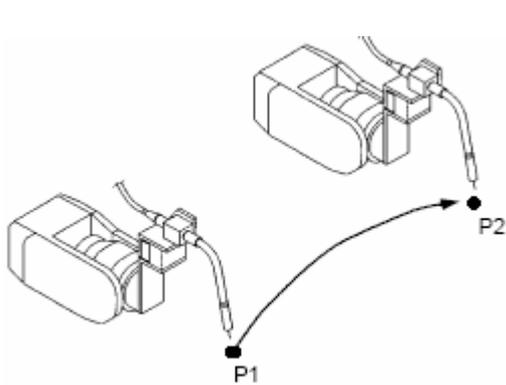


Figura 48 – Movimento em junta [6]

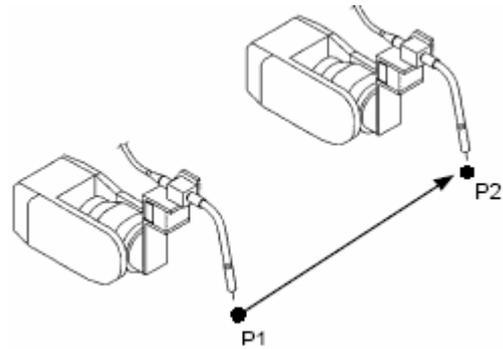


Figura 49 – Movimento linear [6]

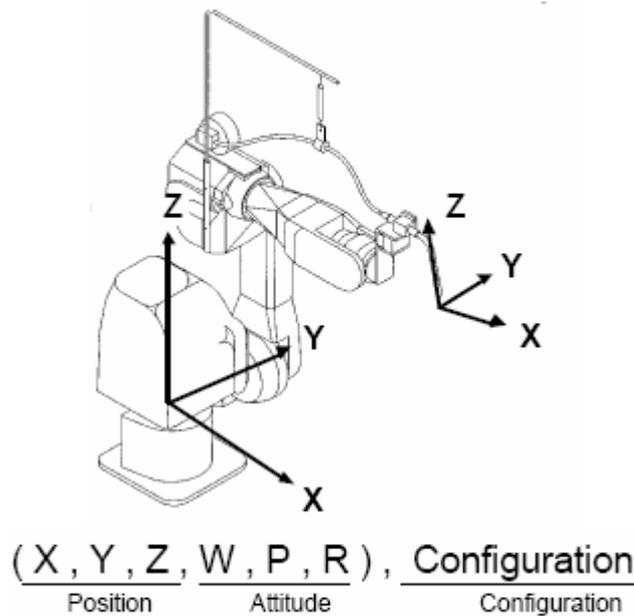


Figura 50 – Composição de ponto cartesiano [6]

( F, L, U, T, 0, 0, 0 )			
Joint placement			Turn number
{ FLIP NOFLIP	{ LEFT RIGHT	{ UP DOWN	{ 1: 180_ to 539_ 0: -179_ to 179_ -1: -539_ to 180_
Flip or no flip of the wrist	Left or right of the arm	Up or down of the arm	Front or back of the arm

Figura 51 – Configuração [6]

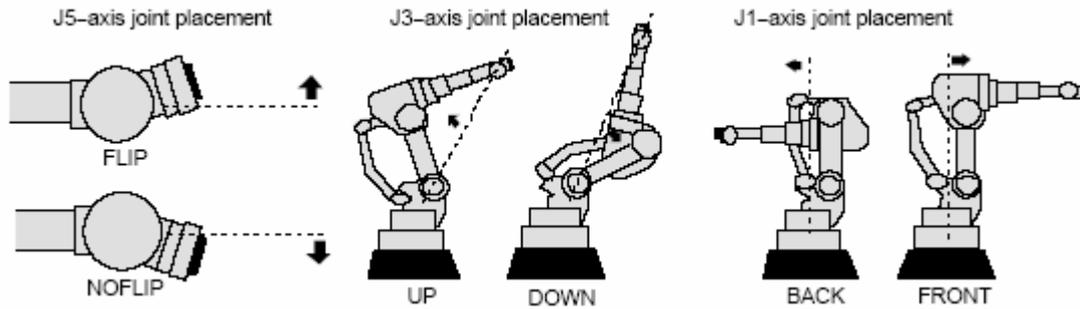


Figura 52 – Redundâncias [6]

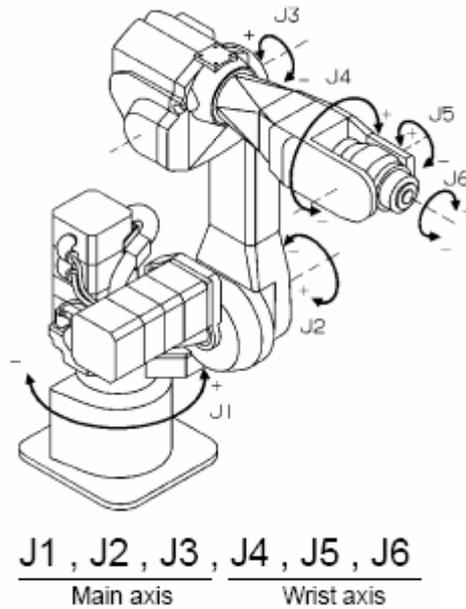
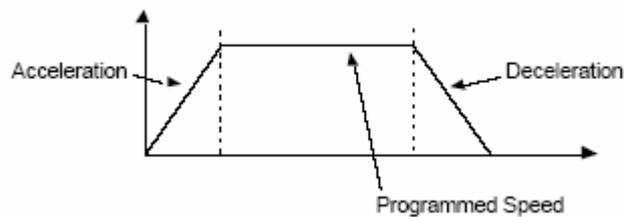


Figura 53 – Composição de configuração de juntas [6]





### 7.2.2. GETCRCPOS e GETCRJPOS

Estes serviços permitem obter a posição actual do TCP do manipulador quer em coordenadas cartesianas, GETCRCPOS, quer em posição angulares de junta, GETCRJPOS.

```

POSITION                               JOINT 30 %
Joint                                   Tool: 1

J1:  0.000 J2:  0.000 J3:
J2:  0.000 J5:  0.000
E1:  *****

[ TYPE ]  JNT  USER  WORLD
    
```

Figura 55 – Ecran TP com posição de juntas [6]

```

POSITION                               JOINT 30 %
World                                   Tool: 1
Configuration: FUT 0
x: 1380.000 y: -380.992 z: 956.895
w:  40.000 p: -12.676 r:  20.000
E1:  *****

[ TYPE ]  JNT  USER  WORLD
    
```

Figura 56 – Ecran TP com posição cartesiana [6]

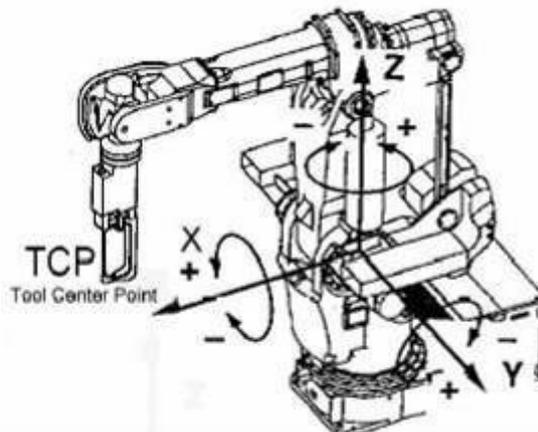


Figura 57 – Posição actual do TCP [6]

De salientar que as coordenadas cartesianas são relativas ao sistema de coordenadas base do manipulador muito embora seja possível configurar o servidor de modo as coordenadas cartesianas sejam relativas a qualquer outro sistema de coordenadas.

#### GETCRCPOS GETCRJPOS

<i>input arg</i>	<i>descrição</i>	<i>tipo</i>
<i>output</i>	<i>Descrição</i>	<i>tipo</i>
X Y Z w p r	Posição cartesiana	String
J1 J2 J3 J4 J5 J6 J7	Posição angular das juntas	String

Tabela 5 – Instruções getCrcPos / getCrjPos

Tabela 6 – Parâmetros getCrcPos / getCrjPos

Para estes comandos não existem argumentos de entrada e o dado de saída é uma cadeia de texto com os valores numéricos da posição actual.

### 7.2.3. CHECKCPOS e CHECKJPOS

Este serviço implementa a validação de posições ou configurações de juntas sobre o volume de trabalho do manipulador.

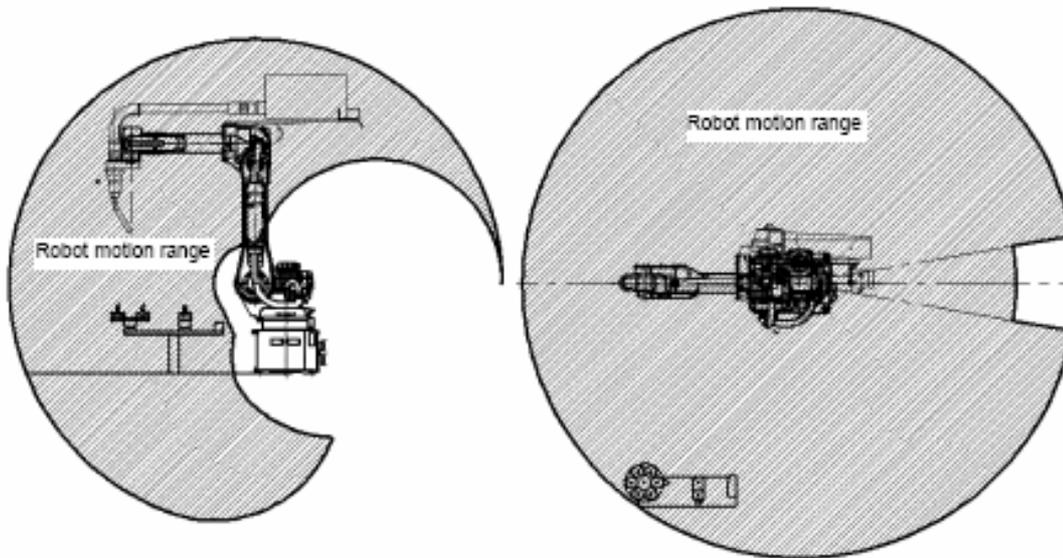


Figura 58 – Exemplo de volume de trabalho de um manipulador industrial [6]

Esta validação é efectuada sempre que se executa um comando de movimento individual, MOVTOCPOS ou MOVTOJPOS, mas não se o comando de movimento for um movimento composto através de um caminho, MOVTHPTH. Assim, dependendo do objectivo ou da necessidade de garantir que uma dada posição destino está dentro do volume de trabalho e é alcançável pelo manipulador, é importante executar este serviço fornecendo as coordenadas cartesianas e orientações da posição a testar, CHECKCPOS, ou a configuração de juntas, CHECKJPOS.

De salientar que a configuração utilizada para o calculo é a configuração actual do manipulador, ou seja, dever-se-á mover previamente o robot para o quadrante da configuração sobre a qual se pretende testar a validade de uma dada posição cartesiana ou configuração de juntas.

**CHECKCPOS** X Y Z w p r  
**CHECKJPOS** J1 J2 J3 J4 J5 J6

<i>input arg</i>	<i>descrição</i>	<i>tipo</i>
X Y Z	Posição cartesiana a verificar	Real
w p r	Orientação da posição a verificar	Real
J1 J2 J3 J4 J5 J6	Configuração de juntas a verificar	Real
<i>output</i>	<i>Descrição</i>	<i>tipo</i>
CHECKCPOS	Validação	String ( OK / NOK )
CHECKJPOS	Validação	String ( OK / NOK )

Tabela 7 – Instrução checkCPos /checkJPos

Tabela 8 – Parâmetros checkCPos / checkJPos

### 7.2.4. GETDIRKIN e GETREVKIN

Estes serviços permitem obter as soluções de cinemática directa para uma dada configuração de juntas ou de cinemática inversa para uma dada posição cartesiana [ Figura 59 ].

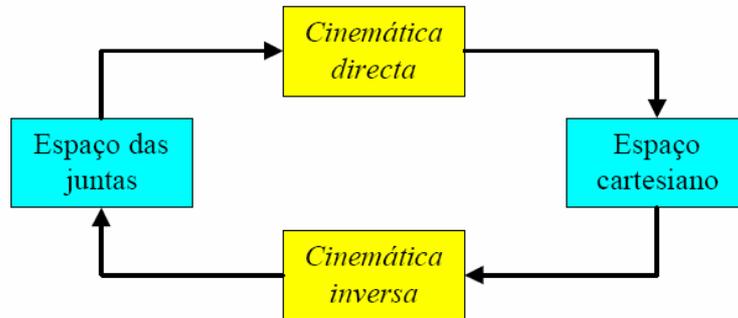


Figura 59 – Cinemática directa e inversa [14]

**GETDIRKIN** J1 J2 J3 J4 J5  
**GETREVKIN** X Y Z w p r

<i>input arg</i>	<i>descrição</i>	<i>tipo</i>
X Y Z w p r	Posição e orientação cartesiana	Real
J1 J2 J3 J4 J5 J6	Configuração de juntas	Real
<i>output</i>	<i>Descrição</i>	<i>tipo</i>
GETDIRKIN	Posição cartesiana	String
GETREVKIN	Configuração de juntas	String

Tabela 9 – Instruções getdirkin / getrevkin

Tabela 10 – Parâmetros getdirkin / getrevkin

### 7.2.5. GETREG

Este serviço permite obter os valores de registos. É possível obter os valores de registos de posição e de registos numéricos mediante a especificação do tipo e número do registo cujos dados se pretende obter.

**GETREG** regType regIndex

<i>input arg</i>	<i>descrição</i>	<i>tipo</i>
regType	Tipo de registo a obter	Real
regIndex	Numero do registo a obter	Real
<i>output</i>	<i>Descrição</i>	<i>tipo</i>
GETREG	Valor do registo	String

Tabela 11 – Instrução getReg

Tabela 12 – Parâmetros getReg

### 7.2.6. SETREG

Este serviço permite atribuir valores a registos. É possível atribuir valores a registos de posição e registos numéricos mediante a especificação do tipo e número do registo cujos dados se pretende obter.

**LISTTP**  
**RUNTP tpPrg**

<i>input arg</i>	<i>descrição</i>	<i>tipo</i>
tpPrg	Nome de programa TP a executar	String
<i>output</i>	<i>Descrição</i>	<i>tipo</i>
LISTTP	Lista de programas	String
RUNTP	Status	String

Tabela 13 – Instruções listTp / runTp

Tabela 14 – Parâmetros listTp / runTp

### 7.2.7. SETPATH e MOVTHPTH

Estes serviços permitem configurar um dado caminho, ie trajectória, através de um conjunto de nodos, ie pontos de passagem, e, posteriormente, mover o TCP do manipulador através desse mesmo caminho.

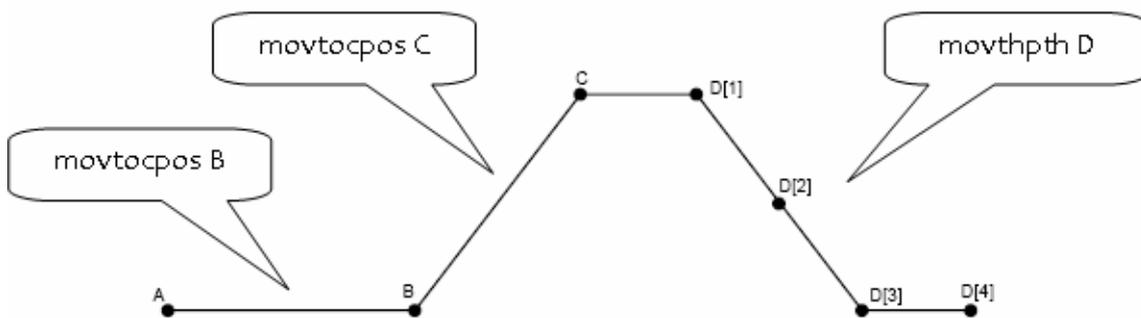


Figura 60 – Instrução movThPth

Relativamente ao SETPATH, é possível especificar a posição cartesiana e respectivas orientações de cada nodo mas é de notar que a configuração utilizada ao longo do movimento é a configuração do manipulador na altura da execução do movimento.

Quanto á instrução MOVTHPTH, é possível especificar a velocidade a utilizar para o movimento através do caminho. Este valor deve ser considerado em mm/s com o máximo de 2000 mm/s e é de notar que passa a ser utilizado como o valor de velocidade utilizado para quaisquer tipos de movimentos seguintes.

**SETPATH X Y Z w p r**  
**MOVTHPTH pathSpeed**

<i>input arg</i>	<i>descrição</i>	<i>tipo</i>
X Y Z	Posição cartesiana a verificar	Real
w p r	Orientação da posição a verificar	Real
pathSpeed	Velocidade	Real

<i>output</i>	<i>Descrição</i>	<i>tipo</i>
SETPATH	Status	String
MOVTHPTH	Status	String

Tabela 15 – Instrução setPath / movThPth

Tabela 16 – Parâmetros setPath / movThPth

### 7.2.8. LISTTP e RUNTP

Estes serviços permitem obter uma lista dos programas TP disponíveis no sistema de ficheiros do controlador e executar um qualquer programa.

**LISTTP**  
**RUNTP tpPrg**

<i>input arg</i>	<i>descrição</i>	<i>tipo</i>
tpPrg	Nome de programa TP a executar	String
<i>output</i>	<i>Descrição</i>	<i>tipo</i>
LISTTP	Lista de programas	String
RUNTP	Status	String

Tabela 17 – Instruções listTp / runTp

Tabela 18 – Parâmetros listTp / runTp

O comando LISTTP, devolve a lista de programas disponível e não dispõe de qualquer argumento de entrada.

Quanto ao comando RUNTP, o único parâmetro de entrada é o nome do programa TP que se pretende executar. Devolverá o estado de execução do comando como OK se tiver sido possível executar o programa solicitado e como NOK se o programa não existir ou não estiver disponível. É possível configurar o servidor de modo a executar este comando de modo síncrono, em que a resposta só é devolvida após a completa execução do programa TP, ou de modo assíncrono, em que a resposta é devolvida logo depois do programa ter sido iniciado.

### 7.2.9. GETDIO e GETAIO

Estes serviços permitem obter o estado ou o valor analógico de entradas digitais ou analógicas respectivamente.

**GETDIO iold**  
**GETAIO iold**

<i>input arg</i>	<i>descrição</i>	<i>tipo</i>
iold	Id da I/O	Inteiro
<i>output</i>	<i>descrição</i>	<i>tipo</i>
GETDIO	Estado digital	String ( 1/0 )
GETAIO	Valor analógico	String ( 0 ... 10000 )

Tabela 19 – Instruções getAio / getDio

Tabela 20 – Parâmetros getAio / getDio

O único parâmetro de entrada é o identificador da entrada de que se pretende obter o respectivo estado ou valor enquanto que o dado de saída é um estado digital se se tratar de uma entrada digital mas, relativamente a entradas analógicas, esta funcionalidade utiliza uma razão proporcional com a resolução de 1/10000 para especificar o respectivo valor. A real amplitude do

valor dependerá da forma como a entrada em questão se encontre configurada mas pretendeu-se que esta questão não fosse tornada transparente ao utilizador remoto que não precisará de conhecer a configuração do hardware para poder obter um valor analógico num intervalo conhecido.

### 7.2.10. SETDIO e SETAIO

Estes serviços permitem atribuir um estado ou um valor analógico a saídas digitais ou analógicas respectivamente.

**SETDIO** *iold* *ioBool*  
**SETAIO** *iold* *ioVal*

<i>input arg</i>	<i>descrição</i>	<i>tipo</i>
<i>iold</i>	Id da I/O	Inteiro
<i>ioBool</i>	Estado digital	1 / 0
<i>ioVal</i>	Valor analógico	Inteiro (0 ... 10000)
<i>output</i>	<i>descrição</i>	<i>tipo</i>
SETDIO	Status	string
SETAIO	Status	String (OK/NOK)

Tabela 21 – Instruções setAio / setDio

Tabela 22 – Parâmetros setAio / setDio

Os argumentos são o identificador da saída e o estado digital ou valor analógico a impor enquanto que o retorno é o estado de execução do comando. Conforme a configuração do servidor, retornará *OK* ou *NOK* consoante a operação tenha sido realizada com ou sem sucesso.

De notar que se se tratar de uma saída analógica, esta funcionalidade utiliza uma razão proporcional com a resolução de 1/10000 para especificar o respectivo valor. A real amplitude do valor dependerá da forma como a saída em questão se encontre configurada mas pretendeu-se que esta questão não fosse tornada transparente ao utilizador remoto que não precisará de conhecer a configuração do hardware para poder impor um valor analógico num intervalo conhecido.

### 7.2.11. SETTOOLFRM

Este serviço permite configurar o *TCP* do manipulador.

Esta é uma funcionalidade essencial na operação de manipuladores industriais e um parâmetro de importância crucial pois é sobre este ponto que as cinemáticas são calculadas e todos os pontos referenciados.

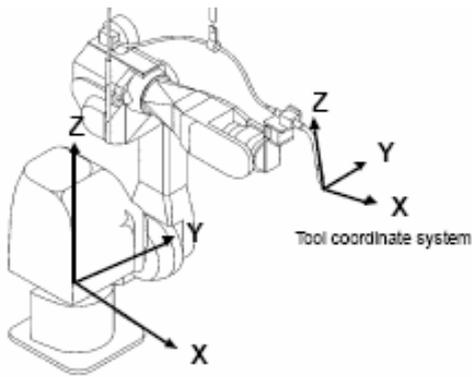


Figura 61 – Instrução setToolFrm [6]

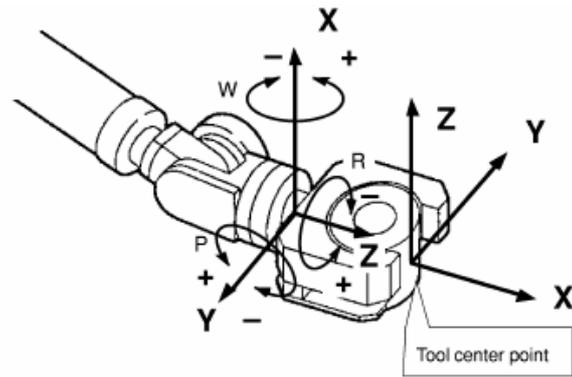


Figura 62 – TCP - Tool Center Point [6]

**SETTOOLFRM** *toolId X Y Z w p r*

<i>input arg</i>	<i>descrição</i>	<i>tipo</i>
<i>toolId</i>	Id da ferramenta	Inteiro
<i>X Y Z</i>	Offset cartesiano da ferramenta relativamente ao TCP <sub>0</sub>	Real
<i>w p r</i>	Orientação da ferramenta relativamente ao sistema de coordenadas World	Real
<i>output</i>	<i>descrição</i>	<i>tipo</i>
SET^TOOLFRM	Status	String

Figura 63 – Instrução setToolFrm

Figura 64 – Parâmetros setAio / setDio

**7.2.12. SETSPEED**

Este serviço permite especificar a velocidade para movimentos simples, lineares ou de junta. O valor de velocidade especificado é válido até ser sobrescrito por este mesmo comando e será sempre atribuído um valor por defeito reduzido da inicialização da aplicação.

Em movimentos de tipo linear, o valor deve ser considerado em mm/s com o máximo de 2000 mm/s enquanto que em movimento de junta, o valor de velocidade é calculado como uma razão proporcional com a resolução de 1/2000 entre o valor atribuído e o valor máximo atingível.

**SETSPEED** *speed*

<i>input arg</i>	<i>descrição</i>	<i>tipo</i>
<i>speed</i>	Velocidade	Real
<i>output</i>	<i>descrição</i>	<i>tipo</i>
SETSPEED	Status	String

Figura 65 – Instrução setSpeed

Figura 66 – Parâmetros setSpeed

### 7.2.13. STOPSERV

Este serviço permite terminar de forma unilateral e definitiva a aplicação servidora, não existindo qualquer possibilidade de recuperar sem que seja utilizada a consola de programação.

#### **STOPSERV**

<i>input arg</i>	<i>descrição</i>	<i>tipo</i>
<i>output</i>	<i>descrição</i>	<i>tipo</i>
STOPSERV	Status	String

---

Figura 67 – Instrução stopServ

Figura 68 – Parâmetros stopServ