



## *Relatório – Robô Kuka* **Elementos de Robótica**

**Giovanni Cesar Meira Barboza – Nº USP 11260996**

**José Rafael Souza do Nascimento – Nº USP 9833426**

**Lucas Real – Nº USP 9833339**

**Programação para desenho de Urso**

**Data: 05/06/2023**

### **1. Desenho Cotado**

Para a realização do desenho foi utilizado o programa de CAD Fusion 360 da Autodesk. Na Figura 1 temos o Urso cotado com todas as dimensões utilizadas. Para a realização de curvas foi utilizado a ferramenta de curva utilizando três pontos, facilitando assim a cotação do desenho. Também foi retirado, no caso das semicircunferências, o ângulo realizado por cada uma delas para ser adicionado no parâmetro CA do código CIRC em Kuka.

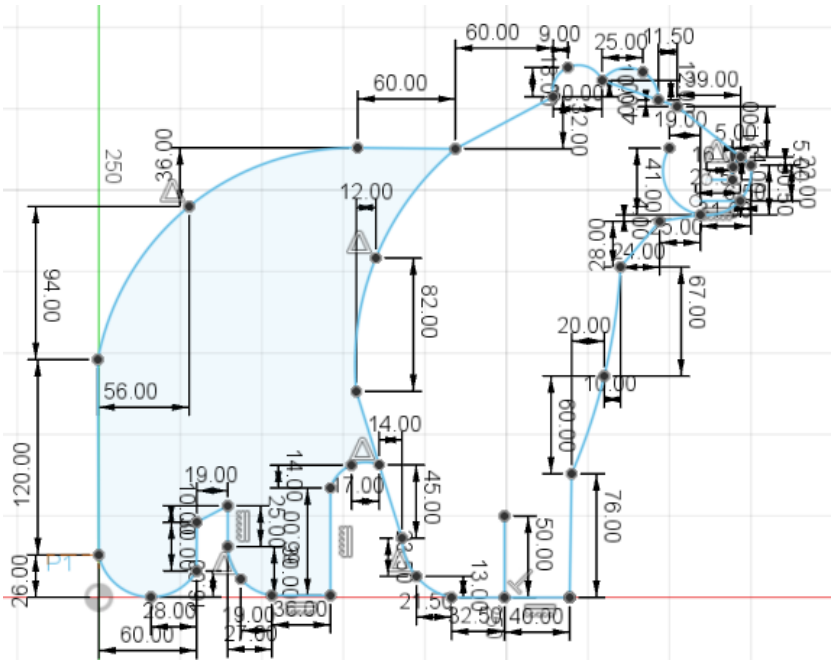


Figura 1: Urso cotado com dimensões e as *constraints* do desenho.

2. Pontos de controle

Foram desenvolvidas algumas tabelas de pontos de controles, em Tabela 1 temos as coordenadas iniciais e finais de cada movimento realizado. O movimento Vazio é o movimento sem contato com o papel e a caneta. O sistema de coordenadas utilizado foi o cartesiano e por isso depois se alterou no FRAME MESA o ângulo para -90°.

Tabela 1: em x e y temos as coordenadas de movimento em mm

x	y	Movimento	x	y	Movimento	x	y	Movimento
0.00	6.50	Inicial	98.50	67.90	Vazio	54.00	0.00	Linha
0.00	36.50	Linha	100.00	66.80	Linha	46.50	9.00	Curva
40.00	68.50	Curva	87.50	69.00	Curva	39.00	32.00	Linha
55.00	68.50	Linha	91.00	61.00	Vazio	55.00	68.50	Curva
70.00	78.00	Linha	98.00	61.00	Linha	42.75	20.50	Vazio
77.50	79.50	Curva	91.00	59.00	Vazio	35.50	16.50	Curva
86.00	76.00	Curva	85.50	57.50	Linha	35.50	0.00	Linha
77.50	79.50	Linha	80.00	50.50	Linha	26.50	0.00	Linha
86.00	76.00	Vazio	72.50	18.50	Curva	20.50	8.00	Curva
90.00	74.40	Linha	72.50	0.00	Linha	20.50	15.00	Linha

98.5 0	67.9 0	Linha	62.50	0.00	Linha	15.0 0	11.5 0	Linha
97.0 0	64.5 0	Curva	62.50	12.5 0	Linha	15.0 0	4.00	Linha
92.5 0	64.5 0	Linha	62.50	0.00	Linha	0.00	6.50	Curva

Tabela 2: Pontos de Controle com as coordenadas em x e y

Ponto	x	y	P15	97	64.5	P31	46.5	9
P1	0	6.5	P16	92.5	64.5	P32	39	32
P2	0	36.5	P17	100	66.8	P33	42.5	52.3
P3	14	60	P18	91	59	P34	55	68.5
P4	40	68.5	P19	87.5	69	P35	42.75	20.5
P5	55	68.5	P20	91	61	P36	38	20
P6	70	78	P21	98	61	P37	35.5	16.5
P7	73	82.5	P22	85.5	57.5	P38	35.5	0
P8	77.5	79.5	P23	80	50.5	P39	26.5	0
P9	84	80.5	P24	77.5	33.5	P40	22	3
P10	86	76	P25	72.5	18.5	P41	20.5	8
P11	77.5	79.5	P26	72.5	0	P42	20.5	15
P12	90	74.4	P27	62.5	0	P43	15	11.5
P13	98.5	67.9	P28	62.5	12.5	P44	15	4
P14	97.5	66.4	P29	54	0	P45	7.5	0
			P30	48.5	3			

### 3. Manual do Programa

A utilização do programa se baseia em um aparelho manual de 5 inputs, onde temos um botão e quatro interruptores. O botão é utilizado para iniciar o desenho (Entrada 1), enquanto os 4 interruptores são utilizados para: terminar o desenho caso solicitado (Entrada 2), definir o lado a ser operado (esquerda ou direita na entrada 3); e dois interruptores para definir quatro tamanhos de imagens (00, 11, 10, 01 – referentes aos tamanhos 50%, 65%, 75% e 85% do desenho original acima, entradas 4 e 5). Entre cada desenho é necessário apertar o botão 1 para continuar a realização, esse tempo é utilizado para alterar os parâmetros de cada desenho utilizando os outros interruptores.

## 4. Listagem do Programa

Abaixo temos o programa utilizado. Foram feitos alguns comentários para melhor interpretação do usuário.

```

&ACCESS RVP
&REL 42
&PARAM TEMPLATE =
C:\KRC\Roboter\Template\vorgabe
&PARAM EDITMASK = *

; caebçalho gerado pelo Kuka

DEF ursoaula( )

; Declaração das variáveis antes do
INI

SIGNAL ENTRADA $IN[4] TO $IN[5]
POS P2
REAL S
INT CONT, CONT_E, CONT_D
FRAME MESA

; Código do Kuka para o INI
;FOLD INI
;FOLD BASISTECH INI
GLOBAL INTERRUPT DECL 3
WHEN $STOPMESS==TRUE DO
IR_STOPM ( )
INTERRUPT ON 3
BAS (#INITMOV,0 )
;ENDFOLD (BASISTECH INI)
;FOLD SPOTTECH INI
USERSPOT(#INIT)
;ENDFOLD (SPOTTECH INI)
;FOLD GRIPPERTECH INI
USER_GRP(0,DUMMY,DUMMY,GDEF
AULT)
;ENDFOLD (GRIPPERTECH INI)
;FOLD USER INI
;Make your modifications
here

;ENDFOLD (USER INI)
;ENDFOLD (INI)

; Código Kuka para o PTP HOME
;FOLD PTP HOME Vel= 100 %
DEFAULT;%{PE}%MKUKATPBASIS,%C
MOVE,%VPTP,%P 1:PTP, 2:HOME, 3:,
5:100, 7:DEFAULT
$BWDSTART = FALSE
PDAT_ACT=PDEFAULT
FDAT_ACT=FHOME
BAS (#PTP_PARAMS,100 )
$H_POS=XHOME
PTP XHOME
;ENDFOLD

; FOLD PTP P1 Vel=100 % PDAT1
Tool[1]:1 Base[0];%{PE}%R
5.6.13,%MKUKATPBASIS,%CMOVE,%
VPTP,%P 1:PTP, 2:P1, 3:, 5:100,
7:PDAT1
$BWDSTART=FALSE
PDAT_ACT=PPDAT1
FDAT_ACT=FP1
BAS(#PTP_PARAMS,100)
PTP XP1
;ENDFOLD

; Primeiro loop para realizar os
desenhos
LOOP
P2=XP1
CONT=1 ; Contador para
saber se já desenhou dois desenhos
em um lado
CONT_E = 1 ; Contador
para saber se já desenhou no lado
Esquerdo
CONT_D = 1 ; Contador
para saber se já desenhou no lado
Direito
MESA={X 0, Y 0,Z 0,A 0,B 0,
C 0} ; Frame criado para utilizar
nosso sistema de variáveis
MESA.X=XP1.X -100 ;
MESA.Y=XP1.Y
MESA.Z=XP1.Z
MESA.A=-90 ;utilizado para
passar do sistema cartesiano para o
sistema kuka global do papel
P2.X=0
P2.Y=0
P2.Z=0

IF $IN[2] THEN ; Caso
switch 2 seja acionado para o
programa
EXIT
ENDIF

IF $IN[3] THEN ; caso do 3
esteja em 1 ele faz do lado direito
MESA.Y=MESA.Y

ELSE ; caso a entrada 3
esteja em 0 faz o lado esquerdo
MESA.Y=MESA.Y + 120

ENDIF

; Caso já tenha desenhado
nos dois lados termina o programa
IF CONT_D > 1 THEN
IF CONT_E > 1 THEN

EXIT
ENDIF
ENDIF

LOOP ; loop para desenhar
dois ursos em um lado do papel

WAIT FOR $IN[1] ; espera
apertar o botão 1 para continuar

;-----
IF $IN[3] THEN ; Bloco
adicionado para não sobrescrever
IF CONT_D > 1 THEN
EXIT
ELSE
CONT_D = CONT_D + 1
ENDIF
ELSE
IF CONT_E > 1 THEN
EXIT
ELSE
CONT_E = CONT_E + 1
ENDIF
ENDIF
;-----

SWITCH ENTRADA

CASE 'B01'
LIN MESA:P2

S=0.85

PLTURSO(P2,S,MESA)
CASE 'B10'

LIN MESA:P2
S=0.75

PLTURSO(P2,S,MESA)
CASE 'B11'

LIN MESA:P2
S=0.65

PLTURSO(P2,S,MESA)
DEFAULT

LIN MESA:P2
S=0.5

PLTURSO(P2,S,MESA)

ENDSWITCH

```

## PMR3502 – Elementos de Robótica

```
IF CONT == 2 THEN ; caso
tenha feito dois ursos de um lado ele
sai do programa
```

```
EXIT
```

```
ELSE
```

```
MESA.X=MESA.X +100 ; Caso já
tenha feito um urso em baixo ele
parte para fazer o outro acima
```

```
CONT=CONT+1
```

```
ENDIF
```

```
ENDLOOP
```

```
ENDLOOP
```

```
;código para o PTP HOME e final do
código.
```

```
;FOLD PTP HOME Vel= 100 %
```

```
DEFAULT;%{PE}%MKUKATPBASIS,%C
```

```
MOVE,%VPTP,%P 1:PTP, 2:HOME, 3:,
5:100, 7:DEFAULT
```

```
$BWDSTART = FALSE
```

```
PDAT_ACT=PDEFAULT
```

```
FDAT_ACT=FHOME
```

```
BAS (#PTP_PARAMS,100 )
```

```
$H_POS=XHOME
```

```
PTP XHOME
```

```
;ENDFOLD
```

```
END
```

```
DEF PLTURSO(P,S,MESA:IN) ;
```

```
subrotina para fazer o plot do urso
```

```
POS P, E, M
```

```
FRAME MESA
```

```
REAL S
```

```
P.Y = P.Y + 6.5*S
```

```
LIN MESA:P
```

```
P.Z = P.Z - 30
```

```
LIN MESA:P
```

```
P.Y = P.Y + 30*S
```

```
LIN MESA:P
```

```
M = P
```

```
M.X = P.X + 14*S
```

```
M.Y = P.Y + 23.5*S
```

```
E = P
```

```
E.X = P.X + 40*S
```

```
E.Y = P.Y + 32*S
```

```
CIRC MESA:M,MESA:E,CA 82.2
```

```
P.X = P.X + 40*S
```

```
P.Y = P.Y + 32*S
```

```
P.X = P.X + 15*S
```

```
LIN MESA:P
```

```
P.X = P.X + 15*S
```

```
P.Y = P.Y + 9.5*S
```

```
LIN MESA:P
```

```
M = P
```

```
M.X = P.X + 3*S
```

```
M.Y = P.Y + 4.5*S
```

```
E = P
```

```
E.X = P.X + 7.5*S
```

```
E.Y = P.Y + 1.5*S
```

```
CIRC MESA:M,MESA:E,CA 180.0
```

```
P.X = P.X + 7.5*S
```

```
P.Y = P.Y + 1.5*S
```

```
M = P
```

```
M.X = P.X + 6.5*S
```

```
M.Y = P.Y + 1*S
```

```
E = P
```

```
E.X = P.X + 8.5*S
```

```
E.Y = P.Y - 3.5*S
```

```
CIRC MESA:M,MESA:E,CA 149.6
```

```
P.X = P.X + 8.5*S
```

```
P.Y = P.Y - 3.5*S
```

```
P.X = P.X - 8.5*S
```

```
P.Y = P.Y + 3.5*S
```

```
LIN MESA:P
```

```
P.Z = P.Z + 30
```

```
LIN MESA:P
```

```
WAIT SEC 1
```

```
P.X = P.X + 8.5*S
```

```
P.Y = P.Y - 3.5*S
```

```
LIN MESA:P
```

```
P.Z = P.Z - 30
```

```
LIN MESA:P
```

```
WAIT SEC 1
```

```
P.X = P.X + 4*S
```

```
P.Y = P.Y - 1.6*S
```

```
LIN MESA:P
```

```
P.X = P.X + 8.5*S
```

```
P.Y = P.Y - 6.5*S
```

```
LIN MESA:P
```

```
M = P
```

```
M.X = P.X - 1.0*S
```

```
M.Y = P.Y - 1.5*S
```

```
E = P
```

```
E.X = P.X - 1.5*S
```

```
E.Y = P.Y - 3.4*S
```

```
CIRC MESA:M,MESA:E,CA 65.0
```

```
P.X = P.X - 1.5*S
```

```
P.Y = P.Y - 3.4*S
```

```
LIN MESA:P
```

```
P.X = P.X - 4.5*S
```

```
LIN MESA:P
```

```
P.Z = P.Z + 30
```

```
LIN MESA:P
```

```
WAIT SEC 1
```

```
P.X = P.X + 6*S
```

```
P.Y = P.Y + 3.4*S
```

```
P.Z = P.Z - 30
```

```
LIN MESA:P
```

```
WAIT SEC 1
```

```
P.X = P.X + 1.5*S
```

```
P.Y = P.Y - 1.1*S
```

```
LIN MESA:P
```

```
M = P
```

```
M.X = P.X - 9.0*S
```

```
M.Y = P.Y - 7.8*S
```

```
E = P
```

```
E.X = P.X - 12.5*S
```

```
E.Y = P.Y + 2.2*S
```

```
CIRC MESA:M,MESA:E,CA 139.4
```

```
P.Z = P.Z + 30
```

```
LIN MESA:P
```

```
WAIT SEC 1
```

```
P.X = P.X - 9.0*S
```

```
P.Y = P.Y - 5.8*S
```

```
LIN MESA:P
```

```
WAIT SEC 1
```

```
P.Z = P.Z - 30
```

```
LIN MESA:P
```

```
WAIT SEC 1
```

```
P.X = P.X + 7*S
```

```
LIN MESA:P
```

```
P.Z = P.Z + 30
```

```
LIN MESA:P
```

```
WAIT SEC 1
```

```
P.X = P.X - 7.0*S
```

```
P.Y = P.Y - 2.0*S
```

```
LIN MESA:P
```

```
P.Z = P.Z - 30
```

```
LIN MESA:P
```

```
WAIT SEC 1
```

```
P.X = P.X - 5.5*S
```

```
P.Y = P.Y - 1.5*S
```

```
LIN MESA:P
```

```
P.X = P.X - 5.5*S
```

```
P.Y = P.Y - 7.0*S
```

```
LIN MESA:P
```

```
M = P
```

```
M.X = P.X - 2.5*S
```

```
M.Y = P.Y - 17.0*S
```

```
E = P
```

```
E.X = P.X - 7.5*S
```

```
E.Y = P.Y - 32.0*S
```

```
CIRC MESA:M,MESA:E,CA 20.1
```

```
P.X = P.X - 7.5*S
```

```
P.Y = P.Y - 32.0*S
```

```
P.Y = P.Y - 18.5*S
```

```
LIN MESA:P
```

```
P.X = P.X - 10.0*S
```

```
LIN MESA:P
```

```
P.Y = P.Y + 12.5*S
```

```
LIN MESA:P
```

```
P.Y = P.Y - 12.5*S
```

```
LIN MESA:P
```

```
P.X = P.X - 7.5*S
```

```
LIN MESA:P
```

```
M = P
```

```
M.X = P.X - 5.5*S
```

```
M.Y = P.Y + 3*S
```

```
E = P
```

```
E.X = P.X - 7.5*S
```

```
E.Y = P.Y + 9*S
```

```
CIRC MESA:M,MESA:E,CA 84.0
```

```
P.X = P.X - 7.5*S
```

```
P.Y = P.Y + 9*S
```

```
P.X = P.X - 7.5*S
```

```
P.Y = P.Y + 23*S
```

```
LIN MESA:P
```

```
M = P
```

```
M.X = P.X + 3.5*S
```

```
M.Y = P.Y + 20.3*S
```

```
E = P
```

```
E.X = P.X + 16*S
```

```
E.Y = P.Y + 36.5*S
```

```
CIRC MESA:M,MESA:E,CA 55.1
```

```
P.Z = P.Z + 30
```

```
P.X = P.X + 16*S
```

```
P.Y = P.Y + 36.5*S
```

```
LIN MESA:P
```

```
WAIT SEC 1
```

```
P.X = P.X - 16*S
```

```
P.Y = P.Y - 36.5*S
```

```
P.X = P.X + 3.75*S
```

```
P.Y = P.Y - 11.5*S
```

```
LIN MESA:P
```

```
P.Z = P.Z - 30
```

```

LIN MESA:P
WAIT SEC 1
M = P
M.X = P.X - 4.75*S
M.Y = P.Y - 0.5*S
E = P
E.X = P.X - 7.25*S
E.Y = P.Y - 4.0*S
CIRC MESA:M,MESA:E,CA 96.9
P.X = P.X - 7.25*S
P.Y = P.Y - 4.0*S
P.Y = P.Y - 16.5*S
LIN MESA:P
P.X = P.X - 9.0*S
LIN MESA:P

```

```

M = P
M.X = P.X - 4.5*S
M.Y = P.Y + 3*S
E = P
E.X = P.X - 6.0*S
E.Y = P.Y + 8*S
CIRC MESA:M,MESA:E,CA 85.7
P.X = P.X - 6.0*S
P.Y = P.Y + 8*S
P.Y = P.Y + 7*S
LIN MESA:P
P.X = P.X - 5.5*S
P.Y = P.Y - 3.5*S
LIN MESA:P
P.Y = P.Y - 7.5*S

```

```

LIN MESA:P
M = P
M.X = P.X - 7.5*S
M.Y = P.Y - 4.0*S
E = P
E.X = P.X - 16.0*S
E.Y = P.Y + 2.5*S
CIRC MESA:M,MESA:E,CA 138.0
P.X = P.X - 16.0*S
P.Y = P.Y + 2.5*S
P.Z = P.Z + 30
LIN MESA:P
END

```

## 5. Resultados

Com os testes no laboratório pudemos revisar o código e testá-lo até obter resultados conforme o esperado. Realizou-se dois desenhos, um de cada lado do papel e cada um com uma escala diferente, 0.5, 0.65, 0.75 e 0.85 do tamanho original. Os resultados do laboratório seguem abaixo na Figura 2 com dois desenhos do lado direito e dois desenhos do lado esquerdo. Os interruptores funcionaram de acordo com o previsto, o robô utilizou para movimentação o comando LIN e o CIRC para semicircunferências. Em nenhum momento a ferramenta se moveu na diagonal, toda saída e retorno ao papel foi feita em linha reta. Foi adicionado um código que impedisse que o programa desenhasse novamente em um mesmo lado do papel e caso ele já tenha escrito nos dois lados ele encerra o programa. Foram utilizadas 11 interpolações circulares e 45 pontos de controle.

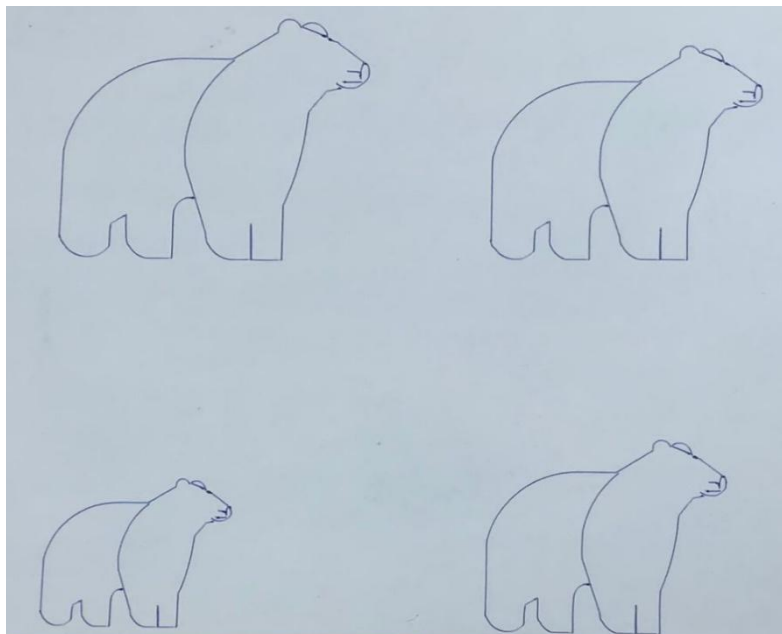


Figura 2: Resultado do teste em laboratório.