Kinematical and Dynamical Models of KR 6 KUKA Robot, Including the Kinematic Control in a Parallel Processing Platform

Chapter · April 2010

DOI: 10.5772/9330 · Source: InTech

3 authors:

John Archila

46 PUBLICATIONS 34 CITATIONS

SEE PROFILE

Max Dutra

Federal University of Rio de Janeiro

126 PUBLICATIONS 667 CITATIONS

SEE PROFILE

Fernando Augusto de Noronha Castro Pinto

Federal University of Rio de Janeiro

34 PUBLICATIONS 113 CITATIONS

SEE PROFILE

Kinematical and Dynamical Models of KR 6 KUKA Robot, including the kinematic control in a parallel processing platform

John Faber Archila Díaz
Universidad Industrial de Santander. UIS
Colombia
Max Suell Dutra and Fernando Augusto de Noronha Castro Pinto
Universidade Federal do Rio de Janeiro, UFRJ
Brazil

1. Introduction

This chapter presents the study and modelling of KR 6 KUKA Robot, of the Robotics Laboratory, Federal University of Rio de Janeiro, see fig 1. The chapter shows the CAD model (Computer Aided Design), the direct kinematics, the inverse kinematics and the inverse dynamical model. The direct kinematic is based in the use of homogeneous matrix. The inverse kinematics uses the quadratic equations model. The dynamical model is based on the use of Euler-Lagrange equations, using the D-H (Denavit-Hartenberg) algorithm and taking into account the inertia tensor, which was found with help of CAE tools (Computer Aided Engineering), On the other hand the Jacobian of robot manipulator is present, it's necessary for the kinematic control. The chapter finishes with the implementation of the inverse kinematic in one parallel processing platform and analyzes its performance.

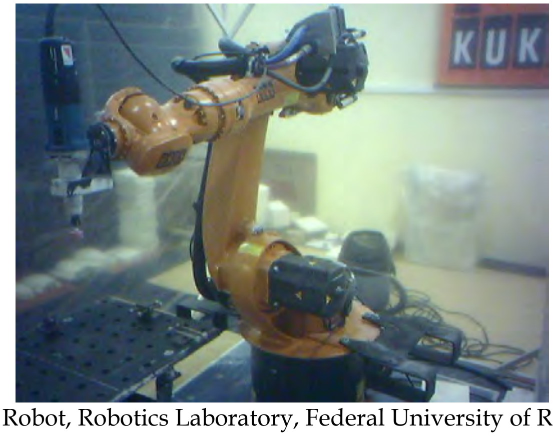


Fig. 1. KR 6 KUKA Robot, Robotics Laboratory, Federal University of Rio de Janeiro UFRJ.

Space movement representation

For the representation of space movements there are several methods such as rotation matrix, vectors, quaternions, roll pitch and yaw, Euler angles, homogenous matrix, among others (Barrientos, 1997). The selected method used for the developing of the direct kinematic model in this work is the homogeneous matrix. The basic concepts for mathematical models formulations are:

Homogeneous Matrix

Homogeneous matrices are 4X4 matrixes, which can represent rotations, translations, scales and perspectives (Ollero, 2001). In general, the homogeneous matrices represent linear transformations. The general form is presented in equation (1)

$$A = \begin{bmatrix} [R(3\times 3)] & [T(3\times 1)] \\ [P(1\times 3)] & [E(1\times 1)] \end{bmatrix} \tag{1}$$

$R(3\times 3)$ Corresponds to a matrix of three rows by three columns, representing rotations.

$T(3\times 1)$ Corresponds to an array of three rows by a column that represents translation.

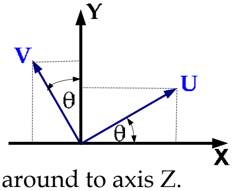
$P(3\times 1)$ Represents a vector of a row of three columns representing the perspective.

$E(1\times 1)$ Corresponds to a scalar that represents the scale of the transformation.

For this case $\vec{P} = \vec{0}$ and $E = 1$

Principal homogeneous matrix

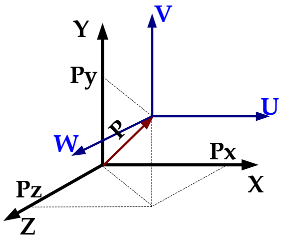
Rotation around the Z axis, figure 2.



$$\begin{bmatrix} Cos(\theta) & -Sin(\theta) & 0 & 0 \\ Sin(\theta) & Cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Fig. 2. Rotation around to axis Z.

Translation Px, Py, Pz, figure 3.



$$\begin{bmatrix} 1 & 0 & 0 & P_x \\ 0 & 1 & 0 & P_y \\ 0 & 0 & 1 & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Fig. 3.Translation

The movements in the space are represented by a series of rotations and translations, these rotations and translations, can be represented as a homogeneous matrix multiplication.

Direct Kinematics

The direct kinematics is the robot kinematic model. In this model, the movements of the robot (coordinates of degrees of freedom) are given and the final positions are found. See Figure 4.

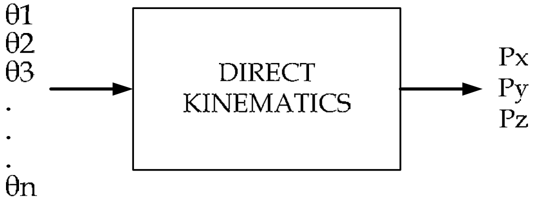


Fig. 4. Direct Kinematics

To find the direct kinematic model, using the homogeneous matrix method, is necessary to make the moves of coordinated system from the fixed base until the last link. For each movement, homogeneous matrices are obtained and the final result is the product of these matrices.

Inverse kinematics

The inverse kinematics seeks the coordinates of each degree of freedom based on the final position of the robot. Figure 4.

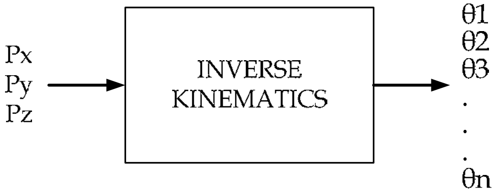


Fig. 5. Inverse Kinematics

The methods used are: the geometric method using the quadratic equation (Dutra, 2006) and the gradient method

Path planning

For the robot operation is necessary to identify different working positions, coordinates and the paths that allow performing a specific task. This work is makes with path planning methods.

Dynamic

In order to the robot control is necessary to know the dynamic model and formulate the dynamic control strategies. To find the dynamic model of the KUKA KR6 the Euler Lagrange model was used. (Kurfles, 2005).

2. Modelling

For the Robot modelling, CAD models and mathematical models were developed. The mathematical models are for the kinematics and dynamics of the manipulator and are presented below.

2.1 CAD Model

The Figure 6 shows the model developed in the CAD software Solid Edge®, based on the KR6 KUKA Robot of Robotics Laboratory at UFRJ, taking into account its main geometric characteristics.

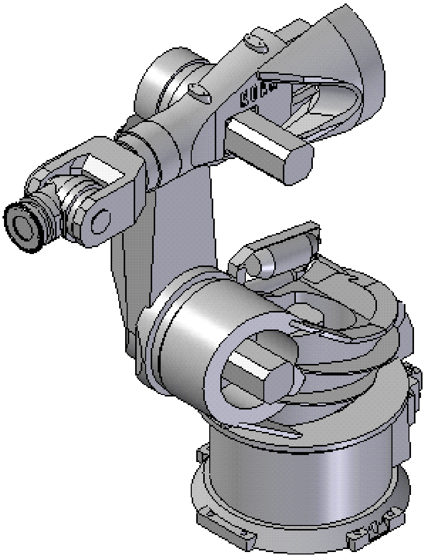


Fig. 6. CAD model of the KR6 KUKA Robot

2.2 Direct Kinematics

The manipulator kinematics model is based on the use of homogeneous matrix for this purpose; coordinated systems are located in a convention proposed by the authors. Supported by recommendations of the Denavit-Hartenberg algorithm (Denavit & Hartenberg, 1955). The convention adopted is:

- On the spin axis of each joint, locate the z-axis direction such that positive rotations are counter clockwise. See Figure 7.
- The x-axis is located parallel to each link, oriented to the follow coordinate system. See Figure 7.

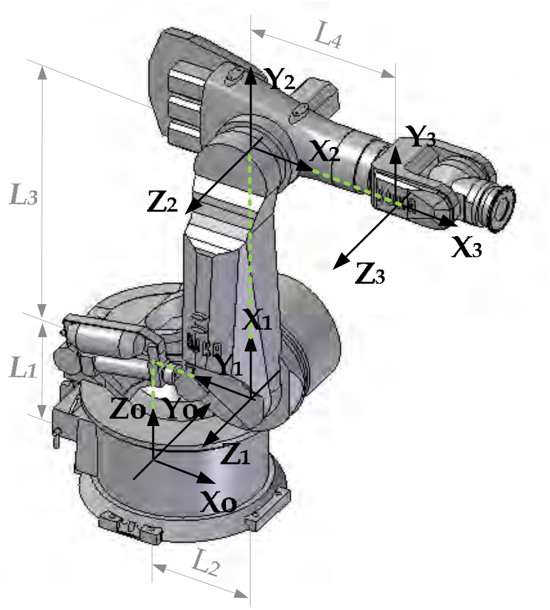


Fig. 7. Reference coordinate systems, for the KR6 KUKA Robot.

The proposed convention, seeks to define all positive rotations of joints when the rotation direction is counter clockwise and to make the translations always on the x-axis in positive direction.

The generated movements for going from one frame to another are mathematically represented by homogeneous matrix transformations and follow the particular geometry of the robot link to link:

1. R(Zo,θo)*T(Zo,L1)
2. T(Xo´,L2)*R(Xo´,π/2)*R(Zo´, π/2)
3. R(Z1,θ1)*T(X1,L3) *R(Z1´, -π/2)
4. R(Z2,θ2)*T(X2,L4)

The full kinematic model is presented in equation (2):

$$\begin{aligned} T = {} & R(Z_o,\theta_0)\times T(Z_o,L_1)\times T(X'_o,L_2) \\ & \times R\left(X'_o,\frac{\pi}{2}\right)\times R\left(Z'_o,\frac{\pi}{2}\right)\times R(Z_1,\theta_1) \\ & \times T(X_1,L_3)\times R\left(Z'_1,-\frac{\pi}{2}\right)\times R(Z_2,\theta_2) \\ & \times T(X_2,L_4) \end{aligned} \tag{2}$$

2.2 Inverse Kinematics

To obtain the inverse kinematic model the geometry of the robot is used, using the quadratic equation method (Dutra, 2006), obtaining the solution for different configurations of the robot in closed form. The first step is to characterize the robot in a vectorial model, for any position, see Figure 8.

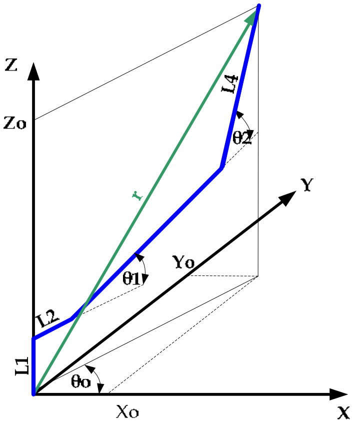


Fig. 8. Vectorial model for inverse kinematics.

Based on the figure 8, the vector sum is developed and the algebraic equations for each joint are obtained:
For the joint θ_1 equation (3)

$$\theta_1 = Sin^{-1}\left(\frac{-B \pm \sqrt{B^2 - 4AC}}{2A}\right) \tag{3}$$

Were

$$\begin{aligned} D &= L_4{}^2 - L_3{}^2 - X^2 - Z^2 \\ A &= 4X^2L_3{}^2 + 4Z^2L_3{}^2 \\ B &= 4ZL_3D \\ C &= D^2 - 4X^2L_3{}^2 \end{aligned} \tag{4}$$

For the joint θ_2 equation (5)

$$\theta_2 = Sin^{-1}\left(\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}\right) - \theta_1 \tag{5}$$

were:

$$\begin{aligned} d &= L_3{}^2 - L_4{}^2 - X^2 - Z^2 \\ a &= 4X^2L_4{}^2 + 4Z^2L_4{}^2 \\ b &= 4ZL_4D \\ c &= d^2 - 4X^2L_4{}^2 \end{aligned} \tag{6}$$

The above equations, equation (3) to equation (6), represent the inverse kinematic model in closed form and providing solutions for different settings of KR 6 KUKA Robot in the workspace.

2.3 Numerical method to the inverse kinematics

For the implementation of inverse kinematic model in a parallel processing platform, the gradient method was proposed in order to observe the platform potential for the application in the control of the manipulator. The general form of the method corresponds to equation (7)

$$\theta_{n+1} = \theta_n - J^{-1}(\theta_n) \times f(\theta_n) \tag{7}$$

Were $J^{-1}(\theta_n)$ correspond to inverse jacobian of the robot, which can be seen in equation (8) and the functions $f(\theta_n)$ depends of kinematical direc model, equation (9)

$$J(\theta_n) = \left[\frac{\partial f_i(\theta)}{\partial \theta_j} \right] = \begin{bmatrix} \frac{\partial f_1(\theta)}{\partial \theta_1} & \frac{\partial f_1(\theta)}{\partial \theta_2} & \frac{\partial f_1(\theta)}{\partial \theta_3} \\ \frac{\partial f_2(\theta)}{\partial \theta_1} & \frac{\partial f_2(\theta)}{\partial \theta_2} & \frac{\partial f_2(\theta)}{\partial \theta_3} \\ \frac{\partial f_3(\theta)}{\partial \theta_1} & \frac{\partial f_3(\theta)}{\partial \theta_2} & \frac{\partial f_3(\theta)}{\partial \theta_3} \end{bmatrix} \tag{8}$$

$$\begin{aligned} f_1(\theta_1, \theta_2, \theta_3) &= a_1 \\ f_2(\theta_1, \theta_2, \theta_3) &= a_2 \\ f_3(\theta_1, \theta_2, \theta_3) &= a_3 \end{aligned} \tag{9}$$

According to the equations (8) and (9), it is necessary to determine the Jacobian of the robot and develop the algorithm to find the corresponding angles for each position of the manipulator, the functions of equations (9) are made explicit in the set of equations (10):

$$\begin{aligned} f_1 &= L_2 \times Cos(\theta_0) \\ &\quad + L_3 \times Cos(\theta_1) \times Cos(\theta_0) \\ &\quad + L_4 \times (Cos(\theta_1 + \theta_2) \times Cos(\theta_0) - X_o \\ f_2 &= L_2 \times Seno(\theta_0) \\ &\quad + L_3 \times Cos(\theta_1) \times Seno(\theta_0) \\ &\quad + L_4 Cos(\theta_1 + \theta_2) \times Seno(\theta_0) - Y_o \\ f_3 &= L_1 + L_3 \times Seno(\theta_1) \\ &\quad + L_4 \times Seno(\theta_1 + \theta_2) - Z_o \end{aligned} \tag{10}$$

The equations (10), correspond to the solution for the direct kinematics of the robot.

2.4 Inverse Dynamic

The dynamic model is based on the Euler Lagrange equations (11). Specifically in the model presented by Fu (1987), it considers the inertial effects of the manipulator by means of the inertia tensor. The use of the transformation matrices is an advantage by the fact that its derivatives can be obtained as a linear combination of a constant matrix multiplied by the original matrix.

$$T_i = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_i}\right) - \frac{\partial L}{\partial \theta_i} \tag{11}$$

were:

L corresponds to the Lagrangian (kinetic energy less potential energy) equation (12)

$$L\left(\theta(t),\dot{\theta}(t)\right) = K\left(\theta(t),\dot{\theta}(t)\right)\text{-U}\left(\theta(\text{t})\right) \tag{12}$$

The model proposed by Fu (1987) is presented in equation (13)

$$T_i = \sum_{k=1}^{n} D_{ik}\ddot{\theta}_k + \sum_{k=1}^{n}\sum_{m=1}^{n} h_{ikm}\dot{\theta}_k\dot{\theta}_m + c_i \tag{13}$$

The model represented in a matrix form is shown in equation (14)

$$T(t) = D(\theta(t))\ddot{\theta}(t) + h(\theta(t),\dot{\theta}(t)) + c(\theta(t)) \tag{14}$$

were:

$T(t) = \left[T_1(t) \quad T_2(t) \quad ...... \quad T_n(t)\right]^T$ Vector torque, size nx1.

$\theta(t) = \left[\theta_1(t) \quad \theta_2(t) \quad ...... \quad \theta_n(t)\right]^T$ Vector of joint positions, size nx1.

$\dot{\theta}(t) = \left[\dot{\theta}_1(t) \quad \dot{\theta}_2(t) \quad ...... \quad \dot{\theta}_n(t)\right]^T$ Vector of angular Velocities, size nx1.

$\ddot{\theta}(t) = \left[\ddot{\theta}_1(t) \quad \ddot{\theta}_2(t) \quad ...... \quad \ddot{\theta}_n(t)\right]^T$ Vector of angular acceleration, size nx1.

$D_{ik} = \sum_{j=\max(i,k)}^{n} Tr\left(U_{jk}J_jU_{ji}^T\right) \qquad i,k = 1,2,....,n$ Inertia matrix, size nxn

$$J_i = \begin{bmatrix} \frac{-I_{xx}+I_{yy}+I_{zz}}{2} & I_{xy} & I_{xz} & m_i\bar{x}_i \\ I_{xy} & \frac{I_{xx}-I_{yy}+I_{zz}}{2} & I_{yz} & m_i\bar{y}_i \\ I_{xz} & I_{yz} & \frac{I_{xx}+I_{yy}-I_{zz}}{2} & m_i\bar{z}_i \\ m_i\bar{x}_i & m_i\bar{y}_i & m_i\bar{z}_i & m_i \end{bmatrix}$$ (15) Inertia Tensor, size 4x4.

$h(\theta,\dot{\theta}) = \left[h_1 \quad h_2 \quad ...... \quad h_n\right]^T$ Vector of Coriolis and centrifugal force, Size nx1

$h_i = \sum_{k=1}^{n}\sum_{m=1}^{n} h_{ikm}\dot{\theta}_k\dot{\theta}_m \qquad i = 1,2,....,n$

$h_{ikm} = \sum_{j=\max(i,k,m)}^{n} Tr\left(U_{jkm}J_jU_{ji}^T\right) \quad i,k,m = 1,2,....,n$

$h_i = \sum_{k=1}^{n}\sum_{m=1}^{n} h_{ikm}\dot{\theta}_k\dot{\theta}_m \qquad i = 1,2,....,n$

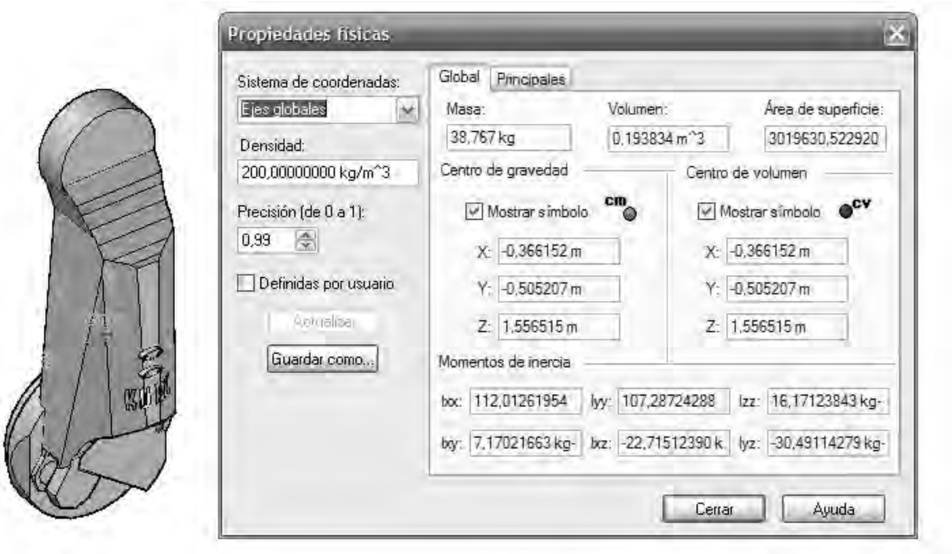
$c(\theta) = \begin{bmatrix} c_1 & c_2 & \ldots\ldots & c_n \end{bmatrix}^T$ Gravity forces vector size nx1

$c_i = \sum_{j=i}^{n} \left(-m_j g U_{ji} \, {}^j r_j \right) \qquad\qquad i = 1, 2, \ldots., n$

From the direct kinematic model, presented in equation (2), is applied the model of Fu , to which is necessary determine Ujk matrices, the inertia tensor Ji for each link, the inertia effects D, the matrix hi and hijk of Coriolis and centrifugal acceleration, the position vector R and the gravitational vectors force C.
To calculate the matrix U_{jk} is used the canonical equation (16):

$$U_{jk} = \frac{\partial \, {}^0 A_j}{\partial \theta_k} = {}^0 A_{j-1} Q_i \, {}^{j-1} A_k \tag{16}$$

To determine the inertia tensor of each link the CAD model was used, obtaining the inertia moments around of the reference system used in the assembly module of the CAD software. As an example is presented the case of link 2, the software presents the inertia in the way shown in Figure 8.


Propiedades físicas

Sistema de coordenadas: Ejes globales

Densidad: 200,00000000 kg/m^3

Precisión (de 0 a 1): 0,99

□ Definidas por usuario

Global | Principales

Masa: 38,767 kg

Volumen: 0,193834 m^3

Área de superficie: 3019630,522920

Centro de gravedad

☑ Mostrar símbolo

X: -0,366152 m

Y: -0,505207 m

Z: 1,556515 m

Centro de volumen

☑ Mostrar símbolo

X: -0,366152 m

Y: -0,505207 m

Z: 1,556515 m

Momentos de inercia

Ixx: 112,01261954 Iyy: 107,28724288 Izz: 16,17123843 kg-

Ixy: 7,17021663 kg- Ixz: -22,71512390 k Iyz: -30,49114279 kg-

Fig. 8. Inertia obtained in Solid Edge®.

The Table 1 was obtained using the software solid edge®, showing data of inertia and centroids.

m_2	38,767	Kg
I_{xx2}	112,0126	Kg-m^2
I_{yy2}	107,2872	Kg-m^2
I_{zz2}	16	Kg-m^2

I_{xy2}	7,1702	Kg-m^2
I_{xz2}	-22,7151	Kg-m^2
I_{yz2}	-30,4911	Kg-m^2
X_c	-0,3661	M
Y_c	-0,505	M
Z_c	1,55	m

Table 1. Inertial and centroidal data for link two.

To obtain the inertia tensor, equation (15), it is necessary to make two changes to the inertia tensor data obtained with solid edge®. The first change is a translation of the reference base system to the reference of the joint and the second change is to orient the base reference system according to the reference system selected in the kinematic model. Figure 9

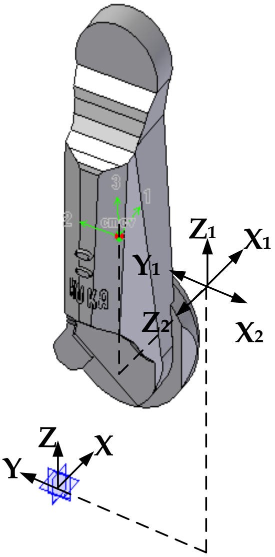


Fig. 9. Reference system for link two

To the translation of the references systems it is necessary to apply the Steiner theorem (Beer, 1980), the distances to apply the theorem are presented in Table 2

X axis2	0	M
Y axis2	-0,505	M
Z axis2	1,35	M

Table 2. Distances to Steiner theorem application.

The inertia tensor for the link two after making changes based on the equation (15) is presented in Table 3

2,295	4,062	7,170	0
4,062	53,840	22,715	7,753
7,170	22,715	1,111	14,195
0	7,753	14,195	38,767

Table 3. Inertia tensor of link two.

For the determination of the matrix D (matrix of inertial effects). It is necessary to use the Equation (17)

$$D(\theta) = \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{21} & D_{22} & D_{23} \\ D_{31} & D_{34} & D_{33} \end{bmatrix} \tag{17}$$

were:

$$D_{11} = Tr\left(U_{11}J_1U_{11}^T\right) + Tr\left(U_{21}J_2U_{21}^T\right) + Tr\left(U_{31}J_3U_{31}^T\right)$$

$$D_{12} = D_{21} = Tr\left(U_{22}J_2U_{21}^T\right) + Tr\left(U_{32}J_3U_{31}^T\right)$$

$$D_{13} = D_{31} = Tr\left(U_{33}J_3U_{31}^T\right)$$

$$D_{22} = Tr\left(U_{22}J_2U_{22}^T\right) + Tr\left(U_{32}J_3U_{32}^T\right)$$

$$D_{23} = D_{32} = Tr\left(U_{33}J_3U_{32}^T\right)$$

$$D_{33} = Tr\left(U_{33}J_3U_{33}^T\right)$$

For the determination of vector h, vector of Coriolis and centrifugal forces, the equation (18) is proposed. This equation presents the angular velocities independently through the matrix *Hi,v*, for purposes of calculation and simulation.

$$h_i = \dot{\theta}^T H_{i,v}\dot{\theta} \tag{18}$$

were:

$$H_{i,v} = \begin{bmatrix} h_{i11} & h_{i12} & h_{i13} \\ h_{i21} & h_{i22} & h_{i23} \\ h_{i31} & h_{i32} & h_{i22} \end{bmatrix}$$

$$h_{ikm} = Tr\left(U_{jkm}J_jU_{ji}^T\right) \quad j = \max(i,k,m)$$

$$U_{jkm} = \frac{\partial U_{jk}}{\partial \theta_m} = {}^0A_{k-1}Q_k\,{}^{k-1}A_{m-1}Q_m\,{}^{m-1}A_i \quad j \geq k \geq m$$

$$U_{jkm} = \frac{\partial U_{jk}}{\partial \theta_m} = {}^0A_{m-1}Q_m\,{}^{m-1}A_{k-1}Q_k\,{}^{k-1}A_i \quad j \geq m \geq k$$

$$U_{jkm} = \frac{\partial U_{jk}}{\partial \theta_m} = 0 \qquad \text{se} \quad j < k \ \text{ ou } \ j < m$$

$$U_{jkm} = \frac{\partial U_{jk}}{\partial \theta_m} = \frac{\partial}{\partial \theta_m}\frac{\partial^0 A_j}{\partial \theta_k}$$

Thus the vector *h* of centrifugal and Coriolis forces is:

$$h_i = \dot{\theta}^T H_{i,v} \dot{\theta}$$

$$\begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \dot{\theta}_1 & \dot{\theta}_2 & \dot{\theta}_2 \end{bmatrix} \begin{bmatrix} h_{111} & h_{112} & h_{113} \\ h_{121} & h_{122} & h_{123} \\ h_{131} & h_{132} & h_{122} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_2 \end{bmatrix} \\ \begin{bmatrix} \dot{\theta}_1 & \dot{\theta}_2 & \dot{\theta}_2 \end{bmatrix} \begin{bmatrix} h_{211} & h_{212} & h_{213} \\ h_{221} & h_{222} & h_{223} \\ h_{231} & h_{232} & h_{222} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_2 \end{bmatrix} \\ \begin{bmatrix} \dot{\theta}_1 & \dot{\theta}_2 & \dot{\theta}_2 \end{bmatrix} \begin{bmatrix} h_{311} & h_{312} & h_{313} \\ h_{321} & h_{322} & h_{323} \\ h_{331} & h_{332} & h_{322} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_2 \end{bmatrix} \end{bmatrix}$$

For the Determination of the gravity force vector C:

$$c(\theta) = \begin{bmatrix} c_1 & c_2 & c_3 \end{bmatrix}^T$$

$$c_i = \sum_{j=i}^{3} \left(-m_j g U_{ji} \, {}^j\overline{r}_j \right) \qquad i = 1,2,3$$

$$c_1 = -m_1 g U_{11} \, {}^1\overline{r}_1 - m_2 g U_{21} \, {}^2\overline{r}_2 - m_3 g U_{31} \, {}^3\overline{r}_3$$

$$c_2 = -m_2 g U_{22} \, {}^2\overline{r}_2 - m_3 g U_{32} \, {}^3\overline{r}_3$$

$$c_3 = -m_3 g U_{33} \, {}^3\overline{r}_3$$

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} -m_1 g U_{11} \, {}^1\overline{r}_1 - m_2 g U_{21} \, {}^2\overline{r}_2 - m_3 g U_{31} \, {}^3\overline{r}_3 \\ -m_2 g U_{22} \, {}^2\overline{r}_2 - m_3 g U_{32} \, {}^3\overline{r}_3 \\ -m_3 g U_{33} \, {}^3\overline{r} \end{bmatrix}$$

The vector r in the reference system of rotation axes is:

$${}^1r_1 = \begin{bmatrix} -0.0052 \\ 0.0026 \\ 0.369 \\ 1 \end{bmatrix} \qquad {}^2r_2 = \begin{bmatrix} 0 \\ 0.2 \\ 0.366 \\ 1 \end{bmatrix} \qquad {}^3r_3 = \begin{bmatrix} 0 \\ 0.454 \\ -0.00015 \\ 1 \end{bmatrix}$$

$$g = \begin{bmatrix} 0 & 0 & -g & 0 \end{bmatrix}$$

The total system is then as follows:

$$T(t) = D(\theta(t))\ddot{\theta}(t) + h(\theta(t), \dot{\theta}(t)) + c(\theta(t))$$

$$\begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} = \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{21} & D_{22} & D_{23} \\ D_{31} & D_{34} & D_{33} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix}$$

$$+ \begin{bmatrix} \begin{bmatrix} \dot{\theta}_1 & \dot{\theta}_2 & \dot{\theta}_2 \end{bmatrix} \begin{bmatrix} h_{111} & h_{112} & h_{113} \\ h_{121} & h_{122} & h_{123} \\ h_{131} & h_{132} & h_{122} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_2 \end{bmatrix} \\ \begin{bmatrix} \dot{\theta}_1 & \dot{\theta}_2 & \dot{\theta}_2 \end{bmatrix} \begin{bmatrix} h_{211} & h_{212} & h_{213} \\ h_{221} & h_{222} & h_{223} \\ h_{231} & h_{232} & h_{222} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_2 \end{bmatrix} \\ \begin{bmatrix} \dot{\theta}_1 & \dot{\theta}_2 & \dot{\theta}_2 \end{bmatrix} \begin{bmatrix} h_{311} & h_{312} & h_{313} \\ h_{321} & h_{322} & h_{323} \\ h_{331} & h_{332} & h_{322} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_2 \end{bmatrix} \end{bmatrix}$$

$$+ \begin{bmatrix} -m_1 g U_{11} {}^1\overline{r}_1 - m_2 g U_{21} {}^2\overline{r}_2 - m_3 g U_{31} {}^3\overline{r}_3 \\ -m_2 g U_{22} {}^2\overline{r}_2 - m_3 g U_{32} {}^3\overline{r}_3 \\ -m_3 g U_{33} {}^3\overline{r} \end{bmatrix}$$

3. Control

3.1 Kinematic Control

For the robot control, the first stage is to make a kinematical control. The different types of Kinematical control are shown below:

Control of position in open loop

This control is appropriated in the case of low operation velocities or in case that the dynamics of the robot does not perturb the behavior of the manipulator. The figure 10 presents the open loop kinematics control.

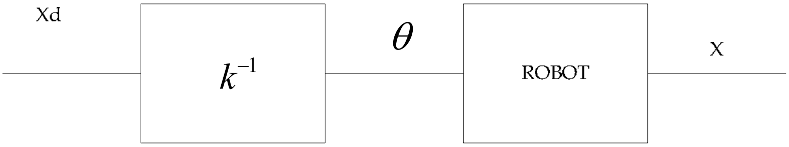


Fig. 10. Block Diagram to open loop kinematics control

In the figure 10 the operator K^{-1} corresponds to inverse kinematics model.

Control of position in closed loop

To make the kinematical position control iterative in a closed loop, it is required to work whit the inverse jacobian J^{-1} as is shown in the figure 11. The operator f(.) corresponds to the inverse kinematic model.

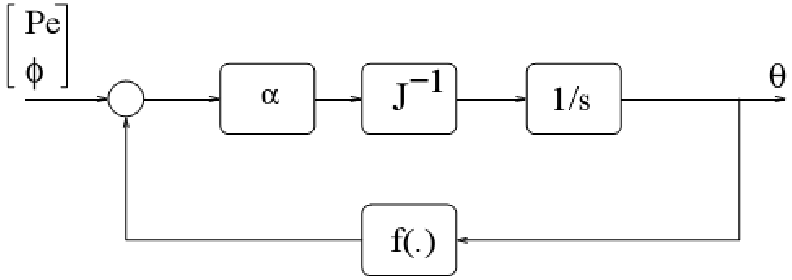


Fig. 11. Iterative control position

Joint Control

Other option is to make a joint level control. In this case, the control is represented in the equation (19) in state variables.

$$\begin{bmatrix} v \\ w \end{bmatrix} = J(\theta)\dot{\theta} \tag{19}$$

Considering that most of the robots usually have a speed control loop at the level of joints: where for one input $u = \dot{\theta}_d$ and a high-gain control $k \longrightarrow \infty$ is found, the error tends to zero $e \longrightarrow 0$ and consequently $u \approx \dot{\theta}$, the desired angular velocity. Figure 12.

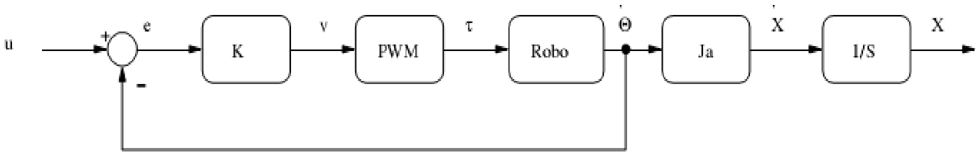


Fig. 12. Joint velocity control

Therefore, the manipulator motion can be described by $\dot{\theta}_i = u_i$, $i = 1,...,n$, were u_i is a signal applied to the motor speed of the i-th joint. Then the control system to be considered is presented in the equation (20):

$$\begin{bmatrix} v \\ w \end{bmatrix} \approx J(\theta)u \tag{20}$$

For the problem of trajectory tracking and to avoid errors in steady state, the control law u can be chosen as a proportional control adding one feed forward term.
Equation (21) and figure 13

$$\overline{u} = \dot{x}_d + K\left(x_d - x\right) \tag{21}$$

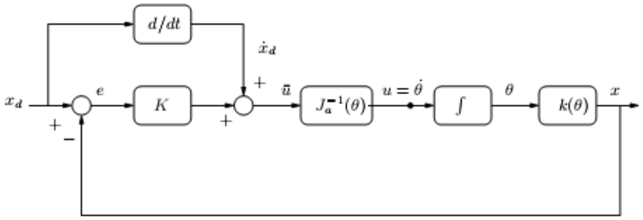


Fig. 13. Control law.

3.2 Control in a parallel processing platform

The Parallel Processing is the ability of an entity to carry out multiple operations or tasks simultaneously, obtaining a better performance and achieving lower response time. There are different platforms of parallel processing like DSPs (Digital Signal processor), FPGAs (Field Programmable Gate Array), among others. In this work the Inverse kinematic model was implemented into a DSP and a microprocessor used to compare its performance. To implement the inverse kinematics model an algorithm based on the gradient method or Newton's method were used (Tsai, 1999). In this case the algorithm calculates the position error and the number of iterations needed to get the result. It is notable that the bigger the number of iterations, the bigger the calculation time will be for the control system, (Archila & Dutra, 2007), which is the response speed limit allowed by the robot and/or the task to realize. An overview of the algorithm is presented in Figure 14.

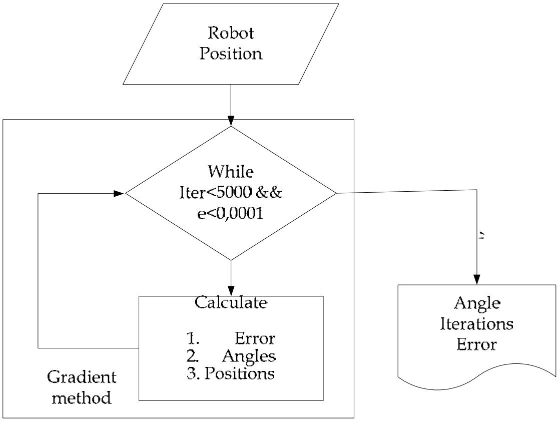


Fig. 14. Algorithm for inverse kinematics.

4. Numeric Simulation

To validate the kinematic and dynamic models Matlab 7.0® was used evaluating the behaviour of each mathematical model to be applied in the KUKA robot controller. The validation of the models works with oblique trajectories passing through two singularities of the robot geometry.

4.1 Computational model to validate the direct kinematics model

In the validations of this model, the input data are angular positions for each joint and the positions for terminal elements are obtained, Figure 15. In this case the joint 2 and 3 takes angular increments up to a 90 degrees rotation in each link.

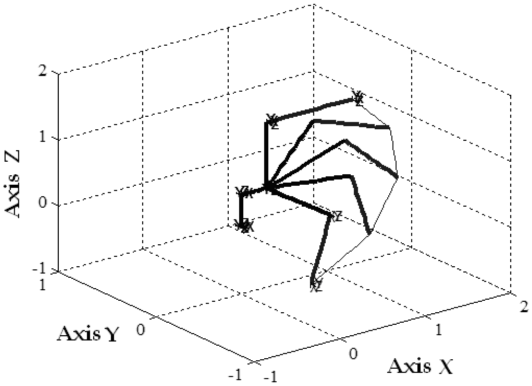


Fig. 15. Direct kinematic model.

4.2 Computational model to validate the inverse kinematics model

The validation of this model requires knowledge of the spatial positions for the robot, the validation used an oblique path obtaining the following results Figure 16.

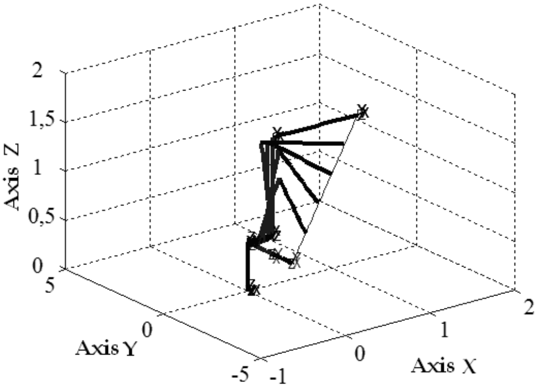


Fig. 16. Inverse kinematic model.

Figure 16 shows that the inverse kinematics model follows the path requested, providing the angular positions needed to achieve the desired positions even for singular points in the workspace of the manipulator.

4.3 Computational model to validate the dynamical model

To validate the dynamic model, the input data were oblique trajectories and tasks that require the motion of the terminal element with constant speed, working load of 16 kg which corresponds to the maximum load recommended by KUKA Roboter (2005). Through the validated inverse kinematics algorithm the angular positions are calculated. Once obtained, the angular positions, it is calculated the angular velocities using the Jacobian inverse, obtaining the following joint speeds Figure 17.

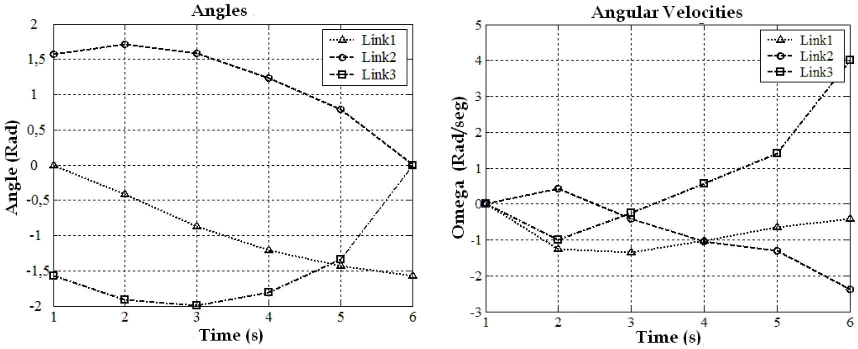


Fig. 17. Angles and angular velocity for the joints.

The angular accelerations required to perform the requested trajectory were calculated and the values are presented in Figure 18.

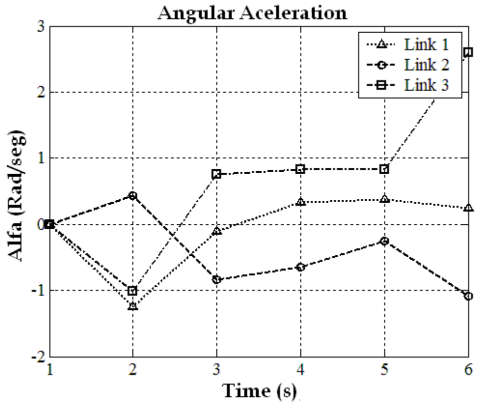


Fig. 18. Angular accelerations obtained in the joints.

With the data of angular positions, angular velocities, angular accelerations and loads applied on the dynamic model of equation (14), the following values of the torques of the robot actuators are obtained, Figure 19.

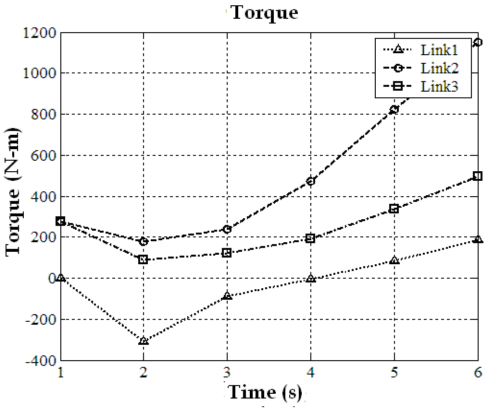


Fig. 19. Torques obtained in the joints.

The data provided by KUKA Roboter (2005) are presented in Table 4, It shows the operational limits of the robot.

	KUKA KR 6
Maximum Torque	3400 Nm
Total mass	206 Kg + 16 Kg
Maximum Velocity	600 Degrees/sec
Repeatability	+/- 0,1 mm

Table 4. Technical data to KR6 KUKA Robot

4.4 Implementation in a parallel processing platform

The inverse kinematics algorithm was implemented in VisualDSP®, Matlab® and C® to observe its performance, evaluating the processing time and the position error.

Processing Time

To evaluate the processing time, oblique trajectories were used in the same way as in the case of the dynamic model validation. The processing times obtained are presented in Figure 20 which clearly shows that the DSP has the shortest processing time.

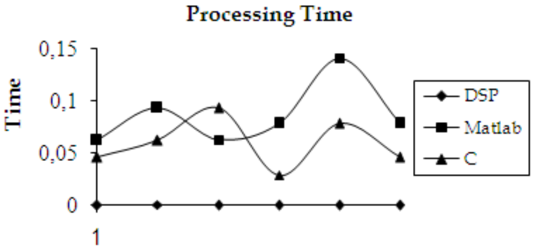


Fig. 20. Comparation of processing time

Evaluation of processing time

The Figure 20 shows that the DSP processing time is the shortest. However, we must assess whether it is appropriate to implement it as a Possible driver. For this purpose it is necessary to review the technical data of the robot to know its response speed. This data can be seen in Table 5, where the main information are: the encoder pulses and maximum angular velocity to which the robot can move, due to technical characteristics of the servo motors used to its operation.

Encoder	512	Pulses/Rev
	1,422	Pulses/Degree
W Max	600	Degrees/sec
Frequency	853,33	Pulses/sec
Period	0,00117	sec/pulse
	0,703	Degrees/pulse
	0,0123	Radians/pulse

Table 5. Technical data to calculus of response speed of KR6 KUKA Robot

The Figure 21 shows that the DSP processing time is appropriated to be implemented as a controller because it offers a maximum response of 0.55 ms, and the robot required 1.17 ms.

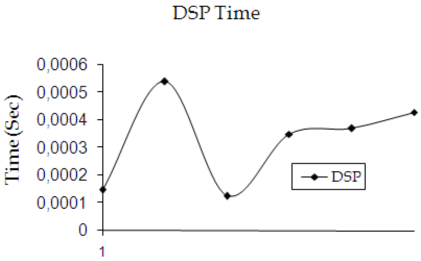


Fig. 21. DSP Processing time

Position Error

Another important characteristic to evaluate, is the position error which corresponds to two main factors: the first one is the quantization error and the second one the error due to the calculation method. The Figure 25 shows the position errors. The comparison parameter corresponds to the repeatability of the robot, which for this case is 0.1 mm in accordance **to** Figure 22.

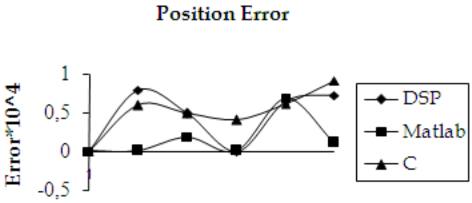


Fig. 22. Position error

In the figure 22, the maximum position error of the DSP is 0.08 mm.

5. Conclusions

The work presents the direct kinematics model of the robot KUKA KR 6, which was evaluated, obtaining suitable results for the kinematic control implemented.

The inverse kinematics model using the quadratic equation was shown to be an appropriated method. It presents the possible configurations of the robot in order to achieve the desired position to follow a trajectory.

The calculation of the inertia tensor was performed with the aid of CAD software, which gives us the principal inertia moments, products of inertia, and the centroid location of each link of the robot

Is Important to show that the inertia tensor obtained needs to be understood and translated to the appropriate reference system for the dynamic model.

The dynamic model was evaluated, obtaining values of 1250 Nm of torque to the oblique trajectory at constant speed work-rate. Obtaining as a maximum speed 12.7 RPM in the joint two, working well within the parameters of the KUKA Roboter.

The performance evaluation of the DSP was adequate, obtaining a processing time of 0.55 ms, appropriate for the given operation.

The position error found corresponds to the calculation method and the quantization error

The maximum error value found was 0.08 mm which is below the repeatability value given by KUKA Roboter.

6. References

Archila J. F, Dutra M. S., (2007). Design and construction of a SCARA Type Manipulator, implementing a control system, International Congress of Mechanical Engineering COBEM 2007. Brasilia.

Barrientos, A., (2001). Fundamentos de robótica, McGraw Hill, pp 15 – 38

Beer F., (1980) Mecânica Vetorial para Engenheiros, McGraw H, pp 399 – 447.

Denavit, J., Hartenberg R. S., (1955) A Kinematic Notation for Lower- Pair Mechanism Based on Matrices, Journal of Applied Mechanics 22: 215 –221.

Dutra, M. S., (2006). Notas de Aula Mecanismos, Universidade Federal do Rio de Janeiro UFRJ.

Fu K. S., (1987) Control, Sensing, Vision, and Intelligence, McGraw Hill, pp 82 – 102.

KUKA Roboter. (2005)Technical specifications Manual.

Kurfles, T., (2005). Robotics and Automation Handbook. CRC Press, pp 26 - 84

Ollero, A., (2001). Robótica, Manipuladores y robots móviles, Primera edición, Alfaomega, Barcelona, pp 43 - 80

Tsai L. W., (1999) Robot Analysis. The Mechanics of Serial and Parallel Manipulators. Editorial John Wiley & Sons, Inc, pp 55 – 72.

Visual DSP ++ 4.5, C/C++ Compiler and library Manual for sharc, Analog Device.