TRAJECTORY PLANNING WITH REDUNDANT COOPERATIVE ROBOTICS SYSTEMS

Cristiane Pescador Tonetto, cris.tonetto@emc.ufsc.br
Altamir Dias, altamir@emc.ufsc.br
Universidade Federal de Santa Catarina
Departamento de Engenharia Mecânica
Laboratório CAD/CAM
Campus Universitário - Trindade
88040-900 – Florinópolis, SC, Brasil

Abstract. *The world trend in industrial automation is to use, more and more, cooperative robotic manipulators to perform some special tasks. Cooperative robots should work together to accomplish the tasks interactively. A system of cooperative robot is composed to operate in several environments like: in larger workspace, to get more flexibility in task programming, and to provide more weight lift capacity, as compared with a system where several single robots perform the same task separately. The paper presents a methodology based on the screw theory and Kirchhoff-Davies method to program the cooperative robotic manipulators kinematics. The screw theory is very appropriate to plan and to control the relative moves between manipulators links. It follows from the Kirchhoff-Davies method how to establish the instant joints position and speeds, as well as to compute the speed of the end effector on the workspace. It gives the necessary equations to evaluate the inverse kinematics of system. The Kirchhoff-Davies method is appropriate, initially, to be applied only to closed kinematics chains, so, the method has to be extended to work with open chains. It is done by using an auxiliary technique, known as Assur virtual chains, which adds "virtual or imaginary" chains, by connecting the robot ground or part to the effector, in order to close an open chain. The robotic system chain are now closed by the virtual chain, and can be considered an integrated part of the whole chain and controlled by impelling the desired movement for the end effector or even other joints. The same technique can be, now, extended again, to control secondary tasks as collision avoidance, paths optimization and singularity dodging. In order to verify the Kirchhoff-Davies method in redundant robotic systems, a method based on the right-side Jacobian pseudo-inverse is applied to get the inversion of non-square matrices. It is possible to find the local minimum of the joints speed norm. The trajectory planning of each end-effectors of cooperating robots is made off-line. The methodology explored in this paper was used to simulate two robots working cooperatively, each one with seven degrees of freedom, to load a body in the workspace. The system is modeled as redundant closed kinematic chains.*
Keywords: *cooperating manipulators, redundant robots system, trajectory planning.*

1. INTRODUCTION

The use of several robots executing tasks cooperatively offers some advantages, such as the reduction of total production time, operation flexibility, greater weight lift capacity. Some authors are currently working on this area, such as Lewis (1996) that proposes the relative Jacobian for two planar robots, in order to plan optimal trajectories. Owen, Croft and Benhabib (2003) assume the whole system as a single redundant robot and the way to plan trajectory for two robots. Dourado (2005) had researched the kinematic and trajectory planning for planar cooperative robots; and Ribeiro, Guenther and Martins (2007) have proposed a Cooperation Jacobian for cooperative robots.

A cooperative robots system is defined as a robotic system composed by several robot manipulators, acting collectively in order to perform one or more tasks. So, cooperative robots programming involves to know how to generate automatically the robots program to different configurations, taking in account its geometry and how to define perfectly the task specifications.

The main goal of this paper is to present a methodology to compute the joints speeds for a redundant robots cooperative system, while performing a given task. Redundant robots have more degrees of freedom than is needed to execute a task, leading naturally a undetermined system solution. So, to override this initial indetermination, it is needed to find out a single solution for the system. In order to accomplish a initial solution for single redundant robots, a pseudo-inverse method was developed. In this paper the pseudo-inverse method is extended to redundant cooperative robotics systems by using the screw theory, the Kirchhoff-Davies method and the Assur virtual chains. When, the method is applied to the cooperative robotics system, it is possible to compute the joint speeds for all redundant robots that compose a system, having an initial information of the each robot and the tasks to be performed.

2. TOOLS TO COMPUTE THE DIFFERENTIAL KINEMATICS

2.1 Screw Theory

The screw theory is based on the application of the Chasles' theorem. The Chasles' theorem establishes that the displacement of a rigid body can be represented as a rotation around a fixed axis and a translation in the direction of the same axis. The combination of the translation and rotation movements simultaneously of a rigid body is called screw displacement (Tsai, 1999). It can be applied to robotic system, where the screw displacement represents both joints rotation and prismatic displacements. So, thinking in this way, it is possible to represent two different types of robots joints, once it defines the relative displacements of links in a single way, and so being possible to overcome pose questions in a generalized way.

Figure 1 depicts the screw displacement of a point P on the space from the P_1 to the position P_2. These displacements can be represented by a rotation of an angle of θ over the screw axis followed by a translation of t on the same axis.

In the same figure, the s axis is, also, a way to represent the point P movements. Since s does not necessarily pass through the origin, the vector s_o denotes the distance of s to the origin O. The S_p point is the intersection between the plane of P_1 and p_r^2 with the s axis. Since the transformation from P_1 to p_r^2 is a rotation by the s axis, this plane is normal to the axis.

The vector p_1 denotes the initial position of the point P referenced to the coordinate's system O, the vectors r_1 and r_2 are same size vectors that define the initial and final position of P according to the screw displacement. The p_2 denotes the position of P after the translation, according to the reference system O (Tsai, 1999).

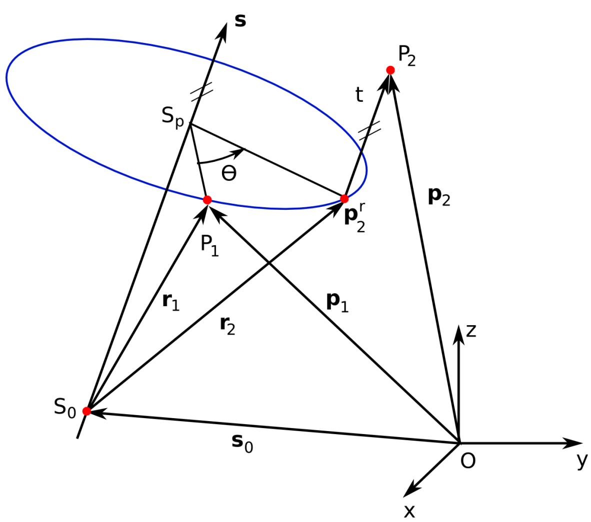


Figure 1. A screw displacement representation of a point P.

In a system composed by many rigid bodies, the s_i vector denotes the unitary vector given the direction of rotation and translation of the screw axis, where i refers to the index of the link or fixed body. The s_{oi} vector denotes the s_i vector position according to a fixed coordinate system to a reference link. The angle θ measures the rotation of the point P around s and t is the length of the displacement of P along s (Tsai,1999). For convenience, the vectors s_i and s_{oi} shall be chosen perpendicular, and expressed by equation:

$$s_{oi}^T s_i = 0 \tag{1}$$

Also, the Mozzi theorem states that the speed of the points of a rigid body relative to a reference system $O(x, y, z)$ can be represented by a differential rotation around a fixed axis and a simultaneous differential translation on the same axis (Tsai, 1999). Thus, any movement of a rigid body can be considered a screw displacement.

Let's consider that the instantaneous movement of a rigid body relative to a inertial system is composed by a pair of vectors $(\omega, v_p)^T$. So, $\omega = (\omega_x, \omega_y, \omega_z)$ denotes the angular speed of a body relative to the chosen coordinate system and the vector $v_p = (v_{px}, v_{py}, v_{pz})$ represents the linear speed of a point P that is in the body (Bonilla, 2004).

Once more, the instantaneous movement can be decomposed as an amplitude and a normalized axis, when there is only rotational or translational movement. If the move is a rotation, the amplitude is the magnitude of the angular velocity $||\omega||$, and when the move is a translation, the amplitude is the magnitude of the linear velocity $||v_p||$.

Let's define a screw $ as the geometric element defined by a directed line on the space and by a scalar parameter h of unitary length. The normalized screw of a joint i is given by the matrix:

$$\hat{\$} = \begin{bmatrix} s_i \\ s_{oi} \times s_i + h s_i \end{bmatrix} \tag{2}$$

where h is the screw pitch and $s_{oi} \times s_i$ is the vector product between s_{oi} and s_i vectors.

It is always possible to verify if the movement is a rotation around the s_i axis or if it is a translation over the s_i axis. The screw that represents the instantaneous movement can be:

$$\$ = (\omega, v_p) = \hat{\$}\dot{q} \tag{3}$$

The screws associate the differential displacement between two bodies related to a reference coordinate system. The compute the differential kinematics between two links in mechanism or robotic chain is generally done by using revolute or translational joint. In 3D chain, where is common to use different types of joints, like spherical joints, those joints can be decomposed in three different revolute joints.

To apply this method it is usually to take some rules, like: if the movement is related to a revolute joint, there is no translation, so, the pitch of the screw is null ($h = 0$), and the screw displacement is given by the equation:

$$\hat{\$} = \begin{bmatrix} s_i \\ s_{oi} \times s_i \end{bmatrix} \tag{4}$$

In the other hand, if just a prismatic joint is used, there is no rotation movement ($\theta = 0$), it is only translative; so the pitch of the screw is infinite ($h = \infty$), and the screw displacement is given by:

$$\hat{\$} = \begin{bmatrix} 0 \\ s_i \end{bmatrix} \tag{5}$$

The resulting screw of the displacement of the end effector can be given by adding the screws of each one of the joints, that is:

$$\$_E = \sum_{i=1}^{n} \hat{\$}_i \dot{q}_i \tag{6}$$

where the i^{th} normalized screw is described related to the reference coordinate system, $\dot{q}_i$ is the magnitude of the displacement of the rigid body i and $\$_E$ is the resulting screw that represents the end effector movement according to the reference coordinate system. Then the displacements of the joints represented by normalized screws and their magnitudes are computed (Tsai, 1999).

2.2 Kirchhoff-Davies Method

The Kirchhoff mesh rule, applied for electric circuits on electrical engineering, establishes that the algebraic sum of all potential differences on a closed electric circuit equals null. Davies adapted this mesh rule for mechanism kinematics computation on closed kinematic chains, so that the sum of relative speeds between two adjacent links throughout any kinematic chain is null (Davies, 1981). This is known as the Kirchhoff-Davies method, and it makes possible to establish an instantaneous relationship between the speeds of all joints of a closed kinematic chain using the speed representation with screws. This relation is named as the equation of constraint in manipulators kinematic chain (Bonilla, 2004), (Dourado, 2005) and (Simas, 2008).

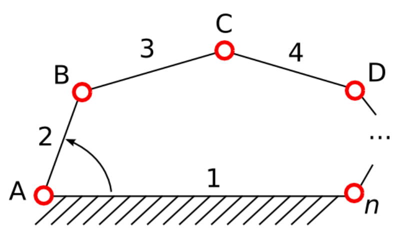


Figure 2. n bars mechanism.

The application of the Kirchhoff-Davies method is better understood with an example with a n bars and n joints mechanism (Figure 2). So, suppose that the screw $\$_A$ represents the movement of link 2 related to the link 1, thus, the screw $\$_i$ represents the movement of the link $\$_{i+1}$ related to the i link, so that $1 < i < n$. The screws $\$_A$, $\$_B$,$\$_C$ ... $\$_n$ represent the kinematics pairs $A, B, C, ...n$, respectively.

The link 2 motion related to link 1 is represented by the screw $\$_A$, and the link 3 movement related to link 1 is represented by the algebraic sum of the screws $\$_A + \$_B$. As the link 1 is fixed to the ground its motion related to itself is null and can be written, as shown in the equation 7, applying the Kirchhoff-Davies method for closed chains.

$$\$_A + \$_B + \$_C + \cdots + \$_n = 0 \tag{7}$$

The vector 0 is a vector of (3×1) dimension. Changing its screws to its normalized axis $\hat{\$}$ and by the speed magnitude $\dot{q}$ leads to:

$$\hat{\$}_A \dot{q}_A + \hat{\$}_B \dot{q}_B + \hat{\$}_C \dot{q}_C + \cdots + \hat{\$}_n \dot{q}_n = 0 \tag{8}$$

where $\hat{\$}_i$ represents the normalized screw of the screw $\$_i$ and $\dot{q}_i$ represents the magnitude of the speed related to the i joint, so that $i \in A, B, C \ldots, n$.

The matrix format is:

$$\begin{bmatrix} \hat{\$}_A & \hat{\$}_B & \hat{\$}_C & \ldots & \hat{\$}_n \end{bmatrix}_{(3 \times n)} \begin{bmatrix} \dot{q}_A \\ \dot{q}_B \\ \dot{q}_C \\ \vdots \\ \dot{q}_n \end{bmatrix}_{(n \times 1)} = 0_{(3 \times 1)} \tag{9}$$

And generally, the constraint equation is represented in a compact way by:

$$N\dot{q} = \overrightarrow{0} \tag{10}$$

where N is the network matrix $(3 \times n)$ that includes the normalized screws and $\dot{q}$ is the matrix $(n \times 1)$ that includes the speed magnitudes associated to the system joints.

The equation 10 can be rewritten in order to find a way to solve the differential kinematics on the joint space of the kinematic chain. It is also a way to find the speed magnitudes of the passive joints as a function of the actuated joints. So, it assumes that the matrices of equation 10 can be divided in two different groups: the joints with known speed magnitudes (primary joints) and on the ones with unknown magnitudes (secondary joints), that is:

$$\dot{q} = \begin{bmatrix} \dot{q}_s \vdots \dot{q}_p \end{bmatrix}^T \tag{11}$$

Reassembling the N network matrix according to the magnitudes leads to:

$$N = \begin{bmatrix} N_s \vdots N_p \end{bmatrix} \tag{12}$$

Finally resulting on equation:

$$\begin{bmatrix} N_s \vdots N_p \end{bmatrix} \begin{bmatrix} \dot{q}_s \\ \cdots \\ \dot{q}_p \end{bmatrix} = 0 \tag{13}$$

Setting the matrices apart produces equation:

$$N_s \dot{q}_s = -N_p \dot{q}_p \tag{14}$$

And thus, equation 9 can be rewritten to:

$$\begin{bmatrix} \hat{\$}_{m+1} & \hat{\$}_{m+2} & \ldots & \hat{\$}_n \end{bmatrix} \begin{bmatrix} \dot{q}_{m+1} \\ \dot{q}_{m+2} \\ \vdots \\ \dot{q}_n \end{bmatrix} = - \begin{bmatrix} \hat{\$}_A & \hat{\$}_B \ldots & \hat{\$}_m \end{bmatrix} \begin{bmatrix} \dot{q}_A \\ \dot{q}_B \\ \vdots \\ \dot{q}_m \end{bmatrix} \tag{15}$$

where m is the number of primary joints.

2.3 Assur Virtual Chains

The Assur virtual chains were developed in order to help the application of Kirchhoff-Davies method. An Assur virtual chain is a serial kinematic chain with virtual links and joints (Bonilla, 2004). It must comply with certain properties, so that the virtual joints movements are represented by linear independent screws and a virtual chain cannot change the degree of freedom of a real kinematic chain.

Therefore, in order to apply it to a kinematic chain, it is assumed that a serial chain must have the same degree of freedom of the screw system in which the real kinematic chain is presented. As the kinematic chains are modified by the added virtual kinematic chains without changing the original chains mobility, it is possible to get information and introduce characteristics on its movement, making possible the application of closed chains algorithms in open chains.

The virtual kinematic chains can be classified according to the screw system order λ in which the real kinematic chain is included. If the real kinematic chain is planar, the screw system has $\lambda = 3$. The planar virtual chains can be PPR, two prismatic and one revolute joint or RPR, one prismatic and two revolute joints. If the real kinematic chain is spatial the screw system has $\lambda = 6$. The spatial virtual chains can be PPPS with three prismatic and one spherical joint, RPPS with one revolute, two prismatic and one spherical joint and RRPS with two revolute, one prismatic and one spherical. (Bonilla, 2004).

The definition of which virtual joints to choose varies according to the desired characteristics of movement, defined by the space over which the trajectory will be defined. In spatial virtual kinematic chains, the order of the screw system is $\lambda = 6$, and thus the virtual chain have a degree of freedom of six. Remembering, a spherical joints can be swapped by three orthogonal revolute joints.

3. REDUNDANT ROBOTS

Before defining redundant robots, some previous concepts must be defined, such as degrees of freedom for a kinematic chain, operational space and joint space:

Definition 1 *The **Mobility** or **degrees of freedom** of a kinematic chain refers to the number of independent parameters needed to completely specify the spatial configuration of a kinematic chain, relative to a link chosen as reference (Tsai, 1999).*

The specification of a task can be related to the operational or the joints space, thus:

Definition 2 *The **operational space** is the space on which the representation of the ends-effector pose allows the task description to be defined as a function of some independent parameters, such as, for example, the trajectory given as a function of time (Sciavicco and Siciliano, 2005).*

Definition 3 *The **joint space** is defined by the set of all possible values for the vector whose components are the variables related to the possible movements of robots joints. This variables include, for example, the angle of rotation for rotational joints or the displacement on the axis direction for prismatic joints (Sciavicco and Siciliano, 2005).*

A robotic manipulator is considered **redundant** when the robot's number of degrees of freedom is greater than the degrees of freedom required to execute a given task. For example, the Roboturb-UFSC manipulator is defined redundant, because it was designed with seven degrees of freedom in order to perform the welding task in hydroelectric turbine rotor blade (Roboturb, 2006).

In the next section the Kirchhoff-Davies methodology and the screw theory will be applied to compute the kinematics of two cooperating robots.

4. KINEMATIC MODEL FOR TWO REDUNDANT ROBOTS

The described theory above was used to simulate two robots Roboturb performing a task. The Roboturb project purpose was to develop a robot for the maintenance of hydroelectric turbine blades, and was fully developed on the Santa Catarina Federal University - Brazil (Roboturb, 2006). The Roboturb robot has seven degrees of freedom, redundant, of course, due the moving and reduced space where the task has to be executed. The Figure 3 depicts the Roboturb robot and a schematic frontal view, which highlights the screws for each joint as well.

The information used to define the kinematic model of cooperative robotics system are the parameters for the manipulators links, such as the links sizes and joint types. For the Roboturb robot, they are $a_1 = 0.150mm$, $a_2 = 0.300mm$, $a_3 = 0.300mm$ and $d_4 = 0.089mm$. The configuration and type of the joints is defined by s and s_o.

One load sharing task is defined for two Roboturb and the trajectory over which the part reference point must be displaced by the robots is a sine wave trajectory $s(t)$, given by the equation:

$$s(t) = \begin{bmatrix} 1 - 0.1\sin(t) & -0.3 + 0.01t & -0.05 \end{bmatrix} \tag{16}$$

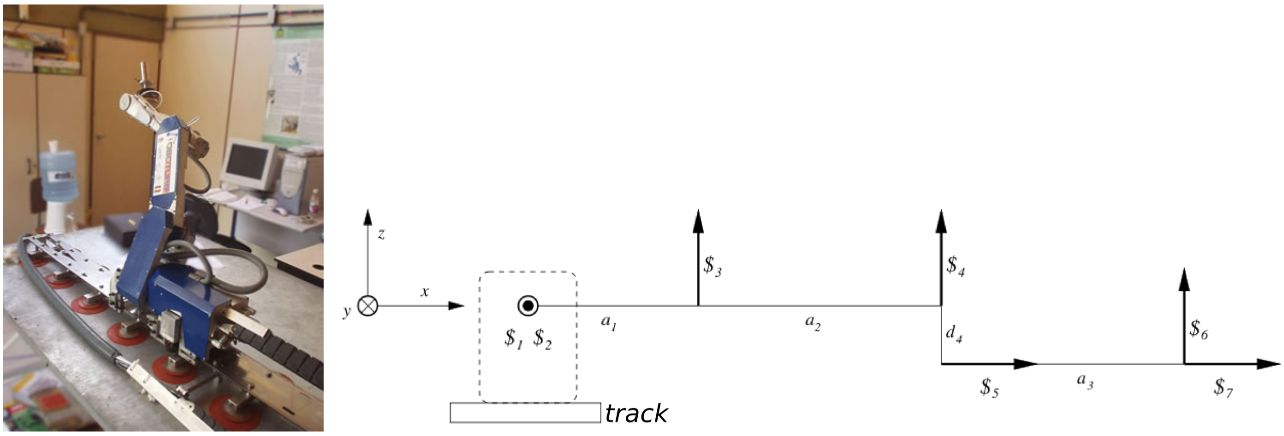


Figure 3. Roboturb (Left) (Roboturb, 2004) and the frontal view of the robot including its screws (Right) (Simas, 2008).

This way, the values of s and s_o are given on Table 1. These values are described for the manipulators reference position, since in this configuration it is easier to define the values of s and s_o.

Table 1. Values of s and s_o for the first manipulator reference position.

$Joints$	s	s_o
1	$(0,-1,0)$	$(0,0,0)$
2	$(0,-1,0)$	$(0,0,0)$
3	$(0,0,1)$	$(a_1,0,0)$
4	$(0,0,1)$	$(a_1+a_2,0,0)$
5	$(1,0,0)$	$(a_1+a_2,0,-d_4)$
6	$(0,0,1)$	$(a_1+a_2+a_3,0,-d_4)$
7	$(1,0,0)$	$(a_1+a_2+a_3,0,-d_4)$

The Table 2 lists the values of s and s_o for the joints of the virtual chain P_x, P_y, P_z, R_x, R_y and R_z that is added in order to close the kinematic chain composed by the serial manipulator.

Table 2. Values of s and s_o for the virtual chain joints.

$Joints$	s	s_o
P_{X1}	$(1,0,0)$	$---------$
P_{Y1}	$(0,1,0)$	$---------$
P_{Z1}	$(0,0,1)$	$---------$
R_{X1}	$(1,0,0)$	$(0,0,0)$
R_{Y1}	$(0,1,0)$	$(0,0,0)$
R_{Z1}	$(0,0,1)$	$(0,0,0)$

In a serial manipulator is necessary to know the movement of the end effector (located on the last link) related to the other joints movement, that is, how the movements of the joints are related to the end effectors movements. Considering that the screws are already known for a given robots configuration, it is needed to compute the screw for all possible configurations. It can be done by the successive screw technique developed by (Tsai, 1999). From the last screw to the first one, the screw $\$_7$ is displaced by all the previous screws, and so on. So, the homogeneous transformation matrix A_7 must be pre-multiplied by the previous A_6, and this product must be pre-multiplied by the previous one, and so on, until there is the multiplication by the A_1 matrix regarding the axis related to the robot base:

$$A_t = A_1 A_2 \cdots A_6 A_7 \quad (17)$$

The matrix A_i, with $i = 1, 2, \ldots 7$, is known as the Rodrigues matrix (Tsai, 1999) and is given by:

$$A_{\vec{s},\theta} = \begin{bmatrix} (s_x^2-1)(1-\cos\theta)+1 & s_x s_y(1-\cos\theta)-s_z\sin\theta & \cdots \\ s_x s_y(1-\cos\theta)+s_z\sin\theta & (s_y^2-1)(1-\cos\theta)+1 & \cdots \\ s_x s_z(1-\cos\theta)-s_y\sin\theta & s_y s_z(1-\cos\theta)-s_x\sin\theta & \cdots \\ 0 & 0 & \cdots \end{bmatrix.$$
$$\left. \begin{matrix} \cdots & s_x s_z(1-\cos\theta)+s_y\sin\theta & ts_x - s_{o_x}(a_{11}-1) - s_{o_y}a_{12} - s_{o_z}a_{13} \\ \cdots & s_y s_z(1-cos\theta)-s_x\sin\theta & ts_y - s_{o_x}a_{21} - s_{o_y}(a_{22}-1) - s_{o_z}a_{23} \\ \cdots & (s_x^2-1)(1-\cos\theta)+1 & ts_z - s_{o_x}a_{31} - s_{o_y}a_{32} - s_{o_z}(a_{33}-1) \\ \cdots & 0 & 1 \end{matrix} \right] \tag{18}$$

The manipulators end effectors pose will be moved from its reference pose to the desired pose by a sequence of successive screw displacements over the joint axis. Thus, the resulting displacement is found by the pre-multiplication of the homogeneous transformation matrix of the successive screws by the equation 17.

The Figure 4 depicts a circuit showing the coupling of the system composed by two robots and their virtual chains, added in order to close the chains. Closed chain is necessary to apply the Kirchhoff-Davies method. Each virtual chain is constituted by three revolute joints and three prismatic joints (3P3R). The circuit is used to get the B incidence matrix and the N network matrices that will be used in the inverse kinematics computation.

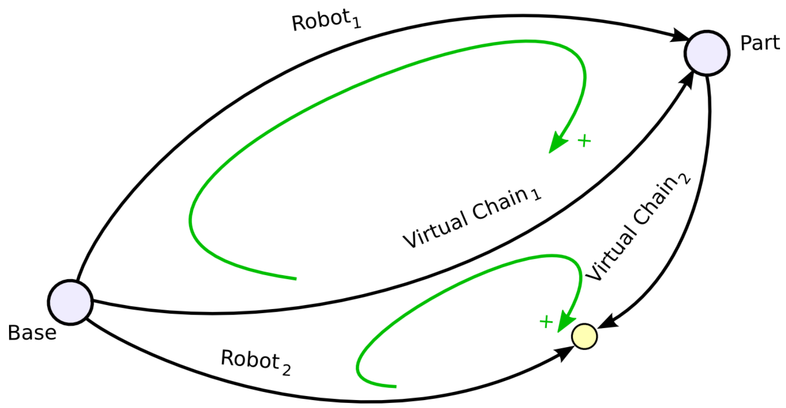


Figure 4. Graph of the cooperating manipulators.

Thus, the B incidence matrix is given by:

$$B = \begin{array}{c} \begin{matrix} \quad R_1 & \quad CV_1 & \quad R_2 & \quad CV_2 \end{matrix} \\ \begin{bmatrix} [1]_{1\times7} & [-1]_{1\times6} & 0 & 0 \\ 0 & [1]_{1\times6} & [-1]_{1\times7} & [-1]_{1\times6} \end{bmatrix} \end{array} \tag{19}$$

And the D matrix of normalized screws is given by:

$$D = \begin{bmatrix} [\hat{\$}_{R_1}]_{6\times7} & [\hat{\$}_{CV_1}]_{6\times6} & [\hat{\$}_{R_2}]_{6\times7} & [\hat{\$}_{CV_2}]_{6\times6} \end{bmatrix} \tag{20}$$

in which $[\hat{\$}_{CV_1}]_{6\times6}$ is the matrix including the screws of the virtual chain 1 and $[\hat{\$}_{R_1}]_{6\times7}$ is the matrix that includes the screws for the robot 1, as well as $[\hat{\$}_{CV_2}]_{6\times6}$ is the matrix for the virtual chain 2 and $[\hat{\$}_{R_2}]_{6\times7}$ is the matrix that includes the robot 2 screws.

The network matrix N can be found multiplying the D matrix by the diagonal matrix representation of the B_b vector, with $b = [1, 2]$, as described by the equation 21.

$$N = \begin{bmatrix} Ddiag\{B_1\} \\ Ddiag\{B_2\} \end{bmatrix} \tag{21}$$

The D matrix is the network matrix and $diag\{B_b\}$ is the diagonal matrix representation of the b line of the incidence matrix B and $b = [1, 2]$.

This way, the N network matrix is:

$$N = \begin{bmatrix} [\hat{\$}_{R_1}]_{6\times 7} & [-\hat{\$}_{CV_1}]_{6\times 6} & 0 & 0 \\ 0 & [\hat{\$}_{CV_1}]_{6\times 6} & [-\hat{\$}_{R_2}]_{6\times 7} & [-\hat{\$}_{CV_2}]_{6\times 6} \end{bmatrix} \tag{22}$$

By the Kirchhoff-Davies method, the network matrix N can be decomposed in N_p e N_s:

$$N_p = \begin{bmatrix} [-\hat{\$}_{CV_1}]_{6\times 6} & 0 \\ [\hat{\$}_{CV_1}]_{6\times 6} & [-\hat{\$}_{CV_2}]_{6\times 6} \end{bmatrix} \tag{23}$$

$$N_s = \begin{bmatrix} [\hat{\$}_{R_1}]_{6\times 7} & 0 \\ 0 & [-\hat{\$}_{R_2}]_{6\times 7} \end{bmatrix} \tag{24}$$

The value of $\dot{q}_p$ is given by the derivative of the robots trajectory, thus for this example the $\dot{q}_p$ vector is given by:

$$\dot{q}_p = \begin{bmatrix} \dot{q}_{CV1} & \dot{q}_{CV2} \end{bmatrix} \tag{25}$$

The N_s matrix is not square, since the robots have seven joints each, and the N_s matrix has 12 rows and 14 columns, being thus not possible to invert it in order to solve equation 14. In order to verify the Kirchhoff-Davies based methodology, a right-hand Jacobian pseudo-inverse (Sciavicco and Siciliano, 2005) was adapted, by using the N_s. This development leads to the equation 26. With this particular pseudo-inverse ones can develop method to optimize the norm of the joints speeds and, by this optimization constraint solve the kinematics for redundant robots.

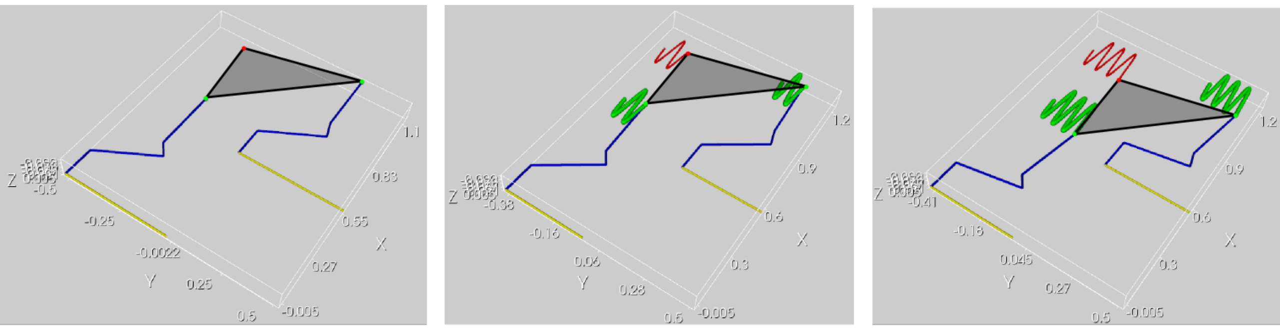
$$\dot{q}_s = -N_s^T (N_s N_s^T)^{-1} N_p \dot{q}_p \tag{26}$$

The task accomplished by the robots is merely illustrative, so that it is possible to verify that the robots end effectors were displaced correctly in order to accomplish the desired trajectory. The task simulates a load sharing task, in which the robots hold an object and move it over the operational space. In this example, the moving is a sine wave trajectory. The trajectory is computed based on the derivative of the trajectory that is given by the vector (notice that the original trajectory is translated according to the position of the robots' grip on the piece):

$$q_p = \begin{bmatrix} 1 - 0.1\sin(t) - 0.3 & 0.01t - 0.3 & -0.05 & 0 & 0 & 0 & 0.1 & 0.5 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{27}$$

$$\dot{q}_p = \begin{bmatrix} 0.1\cos(t) & 0.01 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{28}$$

In the Figure 5 shows the trajectory of each end effector and the configuration of the robots for some intermediate values. Each one of the robots end effectors holds a different location of the piece. This way, the robots trajectory must be such that the piece reference point execute the desired trajectory.



(a) (b) (c)

Figure 5. Two Roboturb operating cooperatively during a task execution: (a) Beginning of the task (b) Middle of the task (c) End of the task.

The left manipulator represents the Robot 1 in the graph, so only it received the trajectory on its virtual chains. The right manipulator (Robot 2) follows the movement holding the other piece corner, since its virtual chain, linked directly to the piece, remains constant (null derivative). Then the movement of the right and left manipulators is cooperative. The input is the trajectory in the $\dot{q_p}$ vector and the robots configuration data and the output is the trajectory of each joint of both robots.

The joint speeds are found and in order to compute the joints position it is still needed to integrate the speeds:

$$q_s(t) = \int_0^t \dot{q}_s(\tau)d\tau \tag{29}$$

The selected robots configuration shown in Figure 5 were computed for time $t = \pi k$, with $k = \{0, 1, 2\}$.

5. CONCLUSION

This paper presents the main concepts for the kinematics modeling of two redundant robots working cooperatively in order to accomplish a task. This kinematics modeling makes possible to apply a novel technique composed by the Kirchhoff-Davies method together with the screw theory and Assur virtual chains. These techniques are been deployed by the UFSC robotics group with good results (Roboturb, 2006).

By using the screw theory it is possible to model both prismatic and revolute joints in a generalized way, and use them easily to compute speeds and displacements. The Assur virtual chains made the use of Kirchhoff-Davies method flexible and allows the use of many different virtual chains, each one used to easily program the robots, by choosing virtual chains that better describe the task to be executed.

One advantage of the Kirchhoff-Davies method is the possibility to apply the technique to robots with any given number of joints without great difficulty. The development presented on this paper on a pseudo-inverse for the Kirchhoff-Davies methods network matrices allows even the use of it on redundant robots. Moreover, the technique can be extended in order to apply it to collision avoidance, too (Simas et al, 2007).

In the proposed methodology, the system redundancy is exploited, despite of the individual structures of each robot. This leads to a more integrated approach, which can solve robots individual redundancy as well as the cases where there are non-redundant robots that, working together, constitute a redundant system.

6. REFERENCES

Bonilla, A. A. C., 2004, "Cinemática diferencial de manipuladores empregando cadeias virtuais", PhD. Thesis (in portuguese), Universidade Federal de Santa Catarina.

Davies, T. H., 1981,"Kirchhoff's circulation law applied to multi-loop kinematic chains", Mechanism and Machine Theory, Vol.16, No.3, pp. 171 - 183.

Dourado A., 2005, "Cinemática de robôs cooperatives", Universidade Federal de Santa Catarina, Master research.

Lewis, C. L., 1996, "Trajectory generation for two robots cooperating to perform a task", International Conference on Robotics and Automation, Proceedings of the 1996 IEEE.

Owen, W. S., Croft E. A., and Benhabib B. C., 2003, "Minimally compliant trajectory resolution for robotic machining", The 11th International Conference on Advanced Robotics, Proceedings of ICAR 2003, pp. 702-707, Portugal, 2003.

Robotics-UFSC Laboratory, http://www.robotica.ufsc.br/, Acessed in 02/05/2009.

Roboturb, 2006, http://www.robotica.ufsc.br/tiki-index.php?page=ROBOTURB, Acessed in 13/04/2009.

Roboturb, 2005, "Robô brasileiro recupera turbinas de usinas hidrelétricas", Inovação Tecnológica, http://www.inovacaotecnologica.com.br/noticias/noticia.php?artigo=010180041130, Acessed in 03/05/2009.

Ribeiro L., Guenther R. and Martins D., 2007, "Screw-Based Relative Jacobian for Manipulators Cooperating in a Task", 19th International Congress of Mechanical Engineering, Brasilia DF, Brasil.

Sciavicco, L and Siciliano B., 2005, "Modelling and control of robot manipulators", Springer, Second edition.

Simas H., da Cruz D. F. M., Guenther R. and Martins D., 2007, "A collision avoidance method using assur virtual chains", 19th International Congress of Mechanical Engineering, Brasilia DF, Brasil.

Simas H., 2008, "Planejamento de trajetórias e evitamento de colisão em tarefas de manipuladores redundantes operando em ambientes confinados", PhD. Thesis (in portuguese), Universidade Federal de Santa Catarina.

Tsai, Lung-Wen, 1999, "Robot Analysis: The Mechanics of Serial and Parallel Manipulators", Jonh Wiley & Sons, INC.

7. Responsibility notice