ISSN 2176-5480

22nd International Congress of Mechanical Engineering (COBEM 2013)
November 3-7, 2013, Ribeirão Preto, SP, Brazil


# Kinematics programming for two cooperating robots performing tasks

**Cristiane Pescador Tonetto**
Universidade Federal do Espírito Santo
cris.tonetto@gmail.com

**Carlos Rodrigues Rocha**
Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul
carlos.rocha@riogrande.ifrs.edu.br

**Altamir Dias**
Universidade Federal de Santa Catarina
altamir@emc.ufsc.br

***Abstract.*** *This paper presents a novel approach for kinematics programming of a system composed by two robots, performing defined tasks. The application of multi-robot cooperating system is of major interest for industries, since it enhances the robotics system flexibility, making it possible to surpass many limitations imposed by the single robot while performing tasks. The common limitations for the application of a single robot are the small lift-weight capacity, workspace or joints limitations and inability to perform complex tasks, that need more degrees of freedom that are available from a single robot. A novel approach is made necessary in order to fully support the cooperation, since the conventional Denavit-Hartenberg method originally does not include multi-robot computation and the adaptations that are presented in literature are not general enough to cover all the possibilities of cooperative tasks. A well structured procedure is presented, using screw theory, Davies' method and Assur virtual chains. By following the presented methodology, the task is specified, and the workspace trajectories are computed. By using the tasks' information, as well as the robots geometric parameters, the joint trajectories are computed. These computational steps include all joints screw definition and the addition of virtual chains, establishing the relationships that make the two-robot system to behave like one coherent system. Finally, the Davies' method is applied over the system representation for the joint trajectories computation. A comparison between the classic Denavit-Hartenberg and the Davies method approaches is finally provided, emphasizing the limitations of the Denavit-Hartenberg approach and sampling the advantage of the full cooperative mode that can be reached by the novel approach.*

***Keywords:*** *kinematics programming, Screw's theory, Davies' method , Cooperative robots*

## 1. Introduction

The goal of this paper is to compare two main approaches for the robotics kinematics computation: the Denavit-Hartenberg convention and the screw theory, applied to cooperative multirobot systems. A comparison between the approaches was done by Rocha *et al.* (2011), in which the authors use the both ways of robotic analysis on the direct kinematics computation for a single Stanford manipulator. In this paper, the analysis methods are extended to a system composed by two robots, comparing the results, including the task planning and kinematics programming, and thinking to apply them to general multirobot cases.

A system composed by two or more robots is defined as a multirobot system, being classified in three main categories:

- Cooperation or Cooperative: the robot help or collaborate for the accomplishment of one or more tasks simultaneously, with some degree of interrelationship among them. The robots system should present some degree of dependency among them. In this paper, Cooperative Multirobot System will be abbreviated as CMS.
- Coordination or Coordinated: The robots share the same common workspace, but every robot accomplishes its task individually. The robots tasks don't have degree of dependency between them.
- Competition or Competitive: The robots compete for tools, workspace or access to the workpiece in the task environment. In contest between robots teams they may have to share tools in order to take advantage in the play.

In the multirobot systems, the robots, in general, share the intersection of compound workspaces. So, the tasks must be well defined, as well as the trajectories related to the end effectors and robots' joints. Such definitions are needed in order to prevent programming difficulties and also while performing the tasks, such as collisions, joints stress to execute the specified operations. Even so some questions emerge: How to compute the kinematics of robots composed system? Which approach is the most appropriate?

ISSN 2176-5480



The paper is divided in six parts: a short introduction, followed by a summary of the Denavit-Hartenberg convention and Screw theory. In the third section, the differences of the approaches are described, according to the kinematics computation of multirobot systems. After that, a simulation is presented, as an application of both approaches in an specific case of study, composed by two robot manipulators. Finally, in the fifth and sixth sections the conclusion and bibliography are presented.

## 2. Approaches for the robot kinematics computation

The Denavit-Hartenberg convention (D-H convention) is certainly the most known and traditional approach applied to the single robot system to compute the position kinematics data in open chains, and the Jacobian for differential kinematics and inverse kinematics. In this section a short presentation of the kinematics computation with D-H convention is made, but it is possible to find more details in robotic literature, here we suggest this reference: Siciliano *et al.* (2009).

### 2.1 Denavit-Hartenberg Convention

The D-H convention suggests a set of steps for defining the pose of the coordinate systems over each joint of the robot. The parameters and the relationship among coordinate systems are shown in Figure 1.

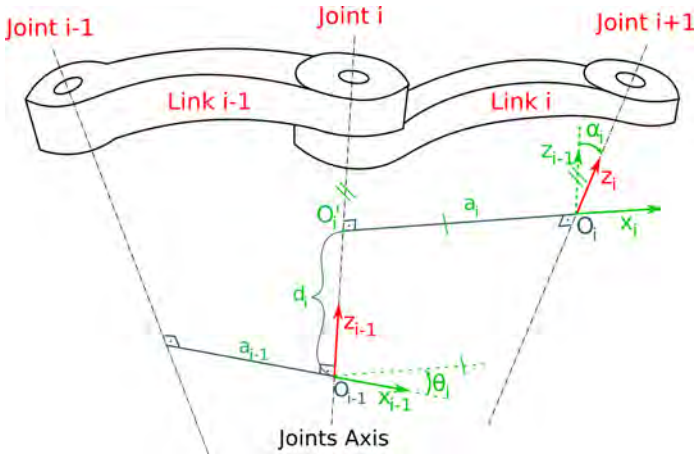


Figure 1. Denavit-Hartenberg Parameters.

1. Identify every link and joint of the robot. The links are enumerated from 0 (base) to $n$ (effector);
2. Define the $z_i$ axis of the $i$ coordinated system through $i + 1$ joint;
3. Locate the origin $O_i$ of the $i$ coordinated system in the intersection of the $z_i$ with the normal axis common to the axes $z_{i-1}$ and $z_i$;
4. Locate $O'_i$ in the intersection of the common normal with the $z_{i-1}$ axis;
5. Choose $x_i$ through the common normal of the axes $z_{i-1}$ and $z_i$;
6. Choose $y_i$ according to the right-hand rule.

The D-H parameters are:

- $a_i$ - the distance between the fixed coordinated systems to the robot links ($O_i$ and $O'_i$);
- $d_i$ - the direction of the coordinate axis from $O'_i$ through $z_{i-1}$;
- $\alpha_i$ - the angle between the $z_{i-1}$ and $z_i$ axes related to the $x_i$ axis;
- $\vartheta_i$ - the angle between the $x_{i-1}$ and $x_i$ axes related to the $z_{i-1}$ axis;

The distances $a_i$ and the angles $\alpha_i$ are always constants, depending only of the geometry of the consecutive joints of link $i$ . In particular, if the $i$ joint is revolute, $\vartheta_i$ is variable, while if $i$ joint is prismatic, $d_i$ is variable (Siciliano *et al.*, 2009; Spong *et al.*, 2006).

The transformation matrix for the coordinates system between the robot's joints $i$ and $i - 1$ is given by:

ISSN 2176-5480



$$A_i^{i-1}(q_i) = \begin{bmatrix} c_{\vartheta_i} & -s_{\vartheta_i}c_{\alpha_i} & c_{\vartheta_i}s_{\alpha_i} & a_i c_{\vartheta_i} \\ s_{\vartheta_i} & c_{\vartheta_i}c_{\alpha_i} & -c_{\vartheta_i}s_{\alpha_i} & a_i s_{\vartheta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

The transformation matrix for the coordinates system from the base to the end-effector is given by a net of the transformation matrices:

$$T_n^0(q) = A_1^0 A_2^1 \cdots A_n^{n-1} = \begin{bmatrix} R(q) & p(q) \\ 0 & 1 \end{bmatrix} \tag{2}$$

in which $R(q)$ is a submatrix with elements in the first to the third rows, and from the first to the third column of the $T_n^0$ matrix and represent a rotation between coordinates system, $p(q)$ is the submatrix containing elements from the first to the third rows of the fourth column of $T_n^0$ and represent the translation between coordinates system, and $q$ is the vector containing the joints variables.

The equation 3 allows to compute the Jacobian systematically, in the direct kinematics programming (Siciliano *et al.*, 2009). The direct kinematics relations is given by:

$$\begin{bmatrix} J_{Pi} \\ J_{Oi} \end{bmatrix} = \begin{cases} \begin{bmatrix} z_{i-1} \\ 0 \end{bmatrix} & \text{for a } \textit{prismatic} \text{ joint} \\ \begin{bmatrix} z_{i-1} \times (p - p_{i-1}) \\ z_{i-1} \end{bmatrix} & \text{for a } \textit{revolute} \text{ joint} \end{cases} \tag{3}$$

in which $z_{i-1}$ is the third column elements of the rotation matrix $R_{i-1}^0$, $p$ is given by the first three elements of the fourth column of the $T_n^0$ and $p_{i-1}$ is given by the first three elements of the fourth column of the $T_{n-1}^0$ matrix.

### 2.2 Screw Theory, Virtual Chains and Davies' Method

Screw $\$$ is a geometric element of 3D space defined by a directed line and by screw pitch (a scalar parameter $h$) (Hunt, 2000). One screw can be represented by a magnitude $\dot{\mathrm{q}}$ and its normalized axis $\hat{\$}$:

$$\$ = \hat{\$}\dot{q} \quad \text{where} \quad \hat{\$} = \begin{bmatrix} s_i \\ s_{oi} \times s_i + h s_i \end{bmatrix} \tag{4}$$

where $s_i$ is an unitary vector with the direction of the screw axis through which the translation and rotation is observed. The $s_{oi}$ vector defines the position of the $s_i$ screw vector related to a fixed coordinate system, $s_{oi} \times s_i$ is the cross product of $s_{oi}$ and $s_i$ vectors, and $h$ is the screw pitch. The Figure 2 shows the screw displacement from $P_1$ to $P_2$ around $s_i$ axis. The translation $t$ has $h$ magnitude for complete rotation $\theta = 2\pi$ around $s_i$ axis.

Now, the screw can represent pure translation or rotation of body motion. When the movement is a pure rotation, the screw step is null ($h = 0$), and the equation 4 leads to $\hat{\$}_{revolute}$. Moreover, when the movement is a translation, Equation 4 should gives the two moving separately as:

$$\hat{\$}_{revolute} = \begin{bmatrix} s_i \\ s_{oi} \times s_i \end{bmatrix}; \qquad \hat{\$}_{translation} = \begin{bmatrix} 0 \\ s_i \end{bmatrix} \tag{5}$$

So, the screw movement description may be used to define the differential displacement between two bodies related to a reference coordinate system (based on Chasles' theorem and on Mozzi's theorem). More details of the screw theory and its applications can be found in the following works: Hunt (2000), Davies (1981), Rocha *et al.* (2011) and Tonetto *et al.* (2012).

The task programming becomes easy by using the screw theory movement description associated with graph theory, where robotic kinematic chain can be represented by a graph. Graph theory properties should be now used to compute task programming. So, a robotic system can be viewed as a combination of real and virtual kinematic chains. The virtual chains, when added to a CMS, either help to analyse the displacements of a kinematic chain or even to impose desired movement to a kinematic chain, as described before. By definition, a virtual chain is a kinematic chain composed by virtual links and joints, that can be added to a real kinematic chain without changing the main behaviour of a real chain

ISSN 2176-5480



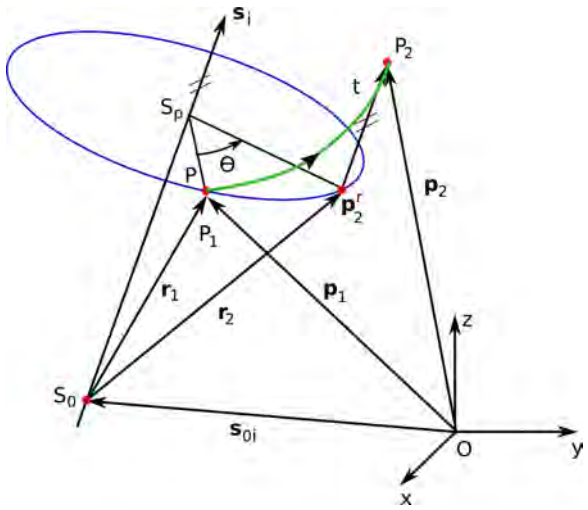


Figure 2. Screw parameters.

(Campos *et al.*, 2005). The kinematic chain should be used to describe the relationship among robots links, tasks and parts in the scenario of CMS's planning Rocha *et al.* (2011), and Tonetto *et al.* (2012).

Davies adapted the Kirchhoff circuit law' method to compute and to relate the joints' velocities magnitudes of a closed kinematic chain, known as Davies'method. The method states that the sum of the relative speeds between kinematic pairs throughout any closed kinematic chain is null (Davies, 1981):

$$\sum_{0}^{n} \$_i = \sum_{0}^{n} \hat{\$}_i \dot{q}_i = 0 \quad \Leftrightarrow \quad N\dot{q} = 0 \tag{6}$$

in which $n$ is the number of joints of the CMS and $N$ is a network matrix containing the normalized screws. To better understand the kinematic chain behaviour the system's joint is classified in a set of primary $N_p$ and secondary $N_s$ joints. Thus, the Equation 6 can be written as:

$$\left[N_s \vdots N_p\right] \begin{bmatrix} \dot{q}_s \\ \cdots \\ \dot{q}_p \end{bmatrix} = 0 \quad \Leftrightarrow \quad N_s\dot{q}_s = -N_p\dot{q}_p \tag{7}$$

If $N_s$ is invertible, the magnitude of the secondary joints $q_s$ can be computed by the following equation:

$$\dot{q}_s = -N_s^{-1}N_p\dot{q}_p \tag{8}$$

By substituting $N_s$, $N_p$ and $q_s$ on Equation 8, the secondary joints velocities magnitudes turn to be described as a function of the primary joints.

## 3. Denavit-Hartenberg and Davies' Method comparison

The robotic system programming can be split in three programming modules: the robotic structure, task and differential kinematic data and computation, as shown in the Figure 3. It can be applied to program just one robot task, as well as extended to a CMS, as shown the set of $m$ levels robots in the Figure 3. The task environment prepares data about the task and compute the trajectories to be used in the differential kinematics computation. The robotics structure includes robot parameters and kinematic model that would be used for computation. In this environment, two approaches are depicted: one follows the Denavit-Hartenberg convention, and the other the Davies Method screw based. In order to compare these two approaches, a system composed by two robots is defined. These robots perform their tasks cooperatively, as shown in the Figure 4.

The kinematics modelling from the Denavit-Hartenberg convention associates the movements of the robots joints to the spatial movements of its end-effector. The approach is based on a single robot and its interaction with the workspace is done exclusively by the end-effector (Figure 5 (a)). In a multirobot system, the Denavit-Hartenberg modelling implies in the individual modelling of each robot, and the "perform tasks" activity would have to be specified as an end-effector movement for each one of the robots. The original Denavit-Hartenberg methodology doesn't support multiple robots, and thus all the modelling and computation must remain individualized.

The screw approach defines the model of each robot individually, too. It uses the reference pose and the application of successive screw displacements. However, after the model definition, the screws are unified (merged) as a single model, representing the robotics system's response to changes in any of the joints that compose it (Figure 5 (b)).

ISSN 2176-5480



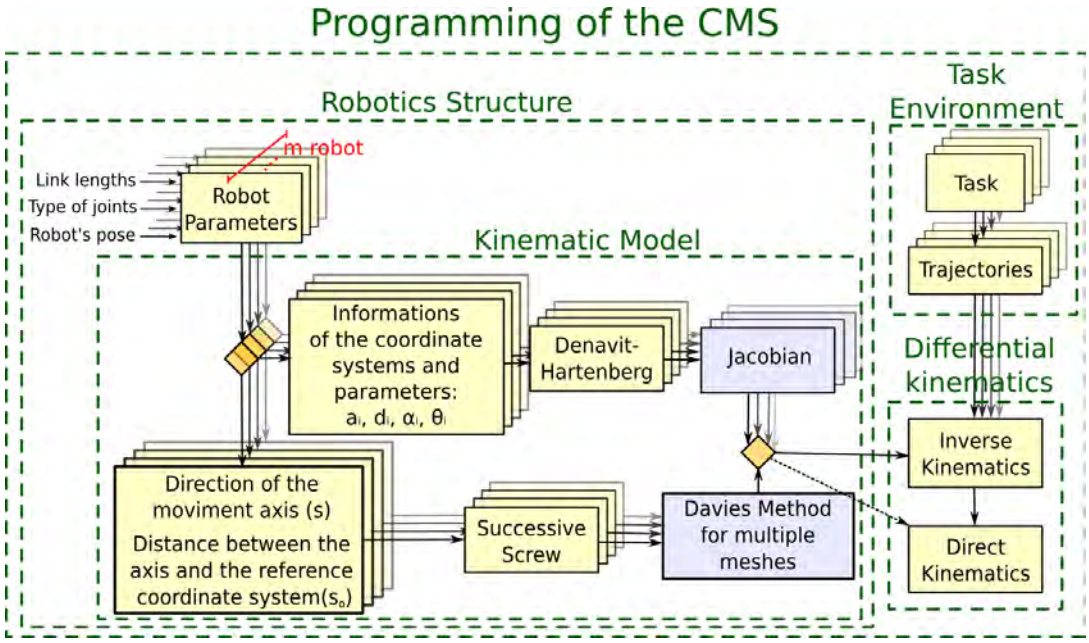


Figure 3. Methodologies applied in CMS kinematics computation.

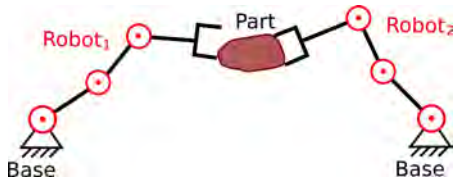


Figure 4. Cooperative Multirobot System.

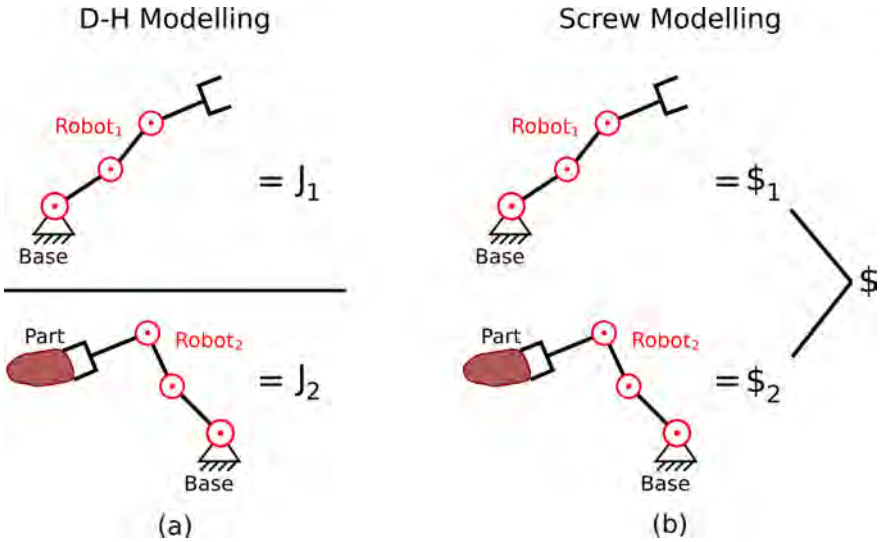


Figure 5. Kinematics modelling by the D-H method and Davies' method.

Since the Denavit-Hartenberg modelling relates the joints to the end-effector movements, every task specification must be provided as an end-effector trajectory. This approach implies that the trajectory planning is robot-oriented, defined as a part of its workspace, while it would be more intuitive and productive to provide task-oriented trajectories (Figure 6 (a)). Besides that, each robot needs it own end-effector trajectory defined, regardless of whether or not depend on trajectories of the other robots and the part as well. This relationship must be assumed, as if each robot would assume that the other will perform their tasks accordingly.

In end-effector trajectory planning, if the task is specified over the part, the effector's movement is defined by a composition of the part's movement and the end-effector trajectory over the part. In this process, the trajectory planning will pass through a trajectory merge, every time the task is defined over the part's or other robot's movement. Finally, the

ISSN 2176-5480



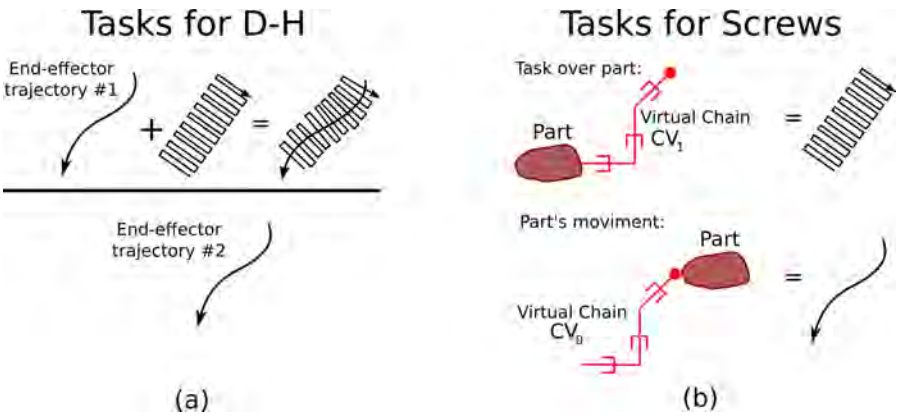


Figure 6. Trajectories computated by each one of the methods.

multirobot system trajectory planning problem is decomposed as multiple single robot problems, losing its cooperative robotic system characteristic, but resembling a coordination system.

The approach using screws provides a robust task-oriented trajectory planning method. Each trajectory must be defined according to the task to be performed - independently of the robot that would perform it, its interrelationships and end-effector movements. It is expected that the method should be capable of handling the interrelationships, which would further come in the kinematics resolution process. The task oriented planning is more natural, since it focus in the final object of the robotic job: the tasks complete. In the task oriented planning, there is two main categories: the displacement of the part related to the reference coordinate system; and the trajectories of the tasks to be performed over the part. Both categories are equally modelled with the Davies' Method, by using virtual chains (Figure 6 (b)). More generally, any task, even outside these two categories, can be modelled by using Assur virtual chains. Just like robots, the tasks modelled by virtual chains are also presented as screw displacements, so every component of the system (robots or tasks) is modelled homogeneously.

modeladas com cadeias virtuais também são representadas da forma helicoidal, de forma que tanto os robôs quanto as tarefas são modelados de forma homogênea.

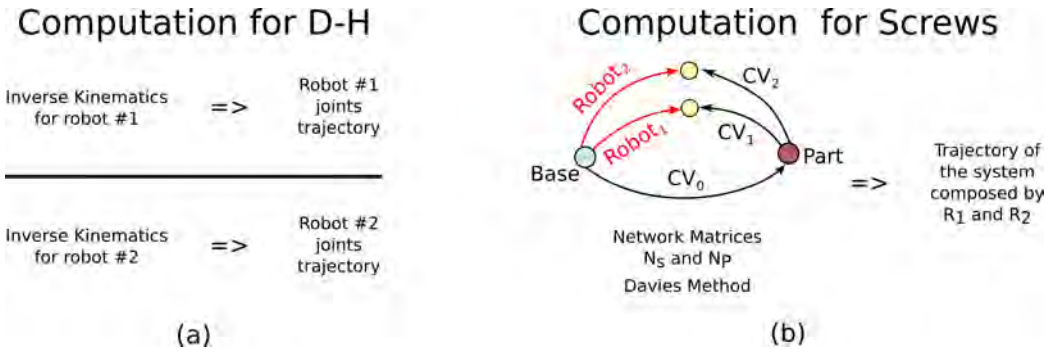


Figure 7. Joint trajectory computation.

Finally, in the joint trajectory computation (differential kinematics), the Denavit-Hartenberg still needs to hold the division assumption, splitting the $n$-robot robotic system in $n$ single robot problems (Figure 7 (a)). This way, the traditional tools apply, for example, the inverse kinematics by the Jacobian inverse, providing the joint movements that lead to the desired movements of the end-effectors, as previously defined in the trajectory planning (task specification by Denavit-Hartenberg). This way, the kinematics computation is not really cooperative, but coordinated, since every task is divided between the robots and one must assume the others performance, as if they would complete their sub-tasks as imposed.

In the screw based methodology, the relationships between robots and tasks are defined from the relationship graph, composed solely by screw elements. With the homogeneous representation of robots and tasks it is possible to unify the models (Figure 7 (b)). After the unification of the relationships of the graph in the network matrix, the Davies' method can be applied, computing the joints to be defined (robots joints) as a function of the already defined joints (the virtual chains or, more specifically, the tasks).

## 4. Simulation

A multirobot system example is composed by two robots: one for the part's displacement and the other for the part's painting job. The part couldn't possibly be painted by a single robot, given the dimension of the task, which is bigger than

ISSN 2176-5480



the workspace of the painter robot. The auxiliary robot displaces the part in order to make the task feasible. The Figure 8 shows the initial position of the robots and the tasks to be performed.

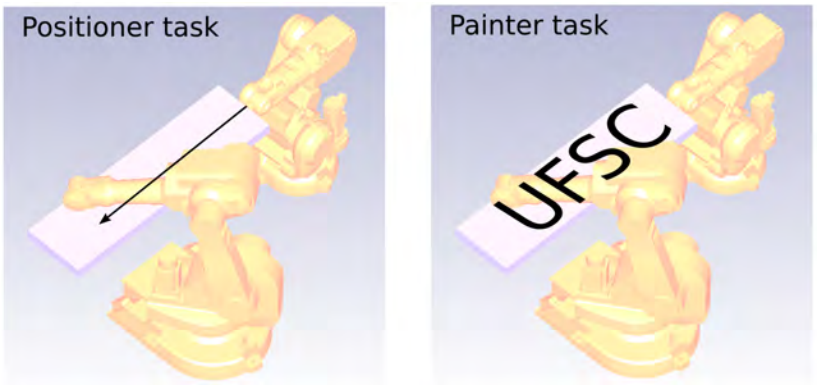


Figure 8. Tasks specification.

The ABB IRB-140 was chosen as positioning robot, while the paint job robot is the ABB IRB-1600. The paint job is defined as the painting of four letters: "UFSC" over the part. The positioning robot displaces the part while the other perform the task over the part. This displacement is defined in the task planning process, in order to move the part into the workspace of the painter robot, so that it would be able to perform the painting over all the part, which is too big to be painted without the movement. Three poses of the robots performing tasks can be seen in the Figure 9.

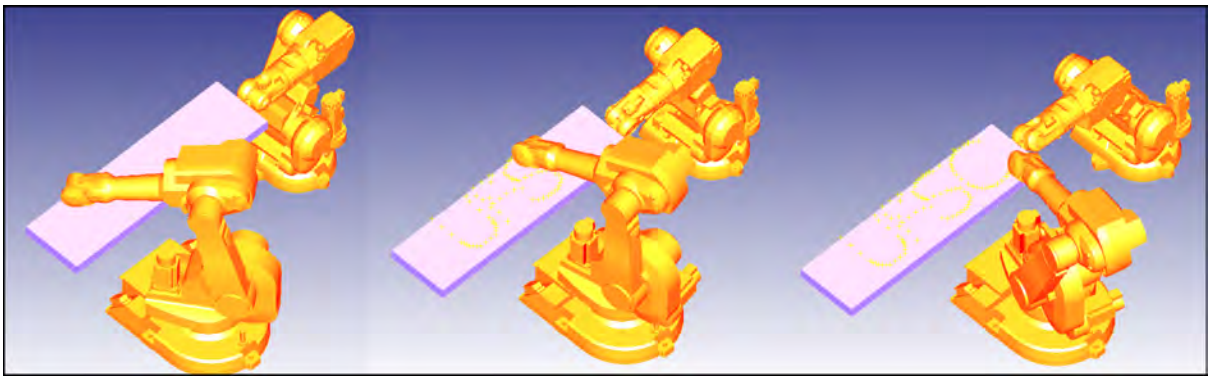

Figure 9. Robots performing tasks.

If there was no intervention of the positioning robot, the painter robot would be unable to perform the painting at once, since the task exceeds the size of its workspace. With all the tasks completely specified during the task planning, both Denavit-Hartenberg and screw methods can be applied. This application shows one example of task that could only be possibly programmed with the cooperation approach, and allows the comparison between the both methods.

The Denavit-Hartenberg method is composed by the classic steps for robots kinematics computation:

1. Modelling of each robot separately, with its Jacobian;
2. Planning of the trajectories of the end-effectors, from the tasks specifications;
3. Computing the inverse kinematics, for each robot individually (through the Inverse Jacobian), finding each robot's joints speeds.

Once more, it's worth noting that the Denavit-Hartenberg implies splitting the two-robot problem into two single-robot problems. This is only possible if the tasks trajectory planning could be decoupled, defining all the end-effectors trajectories before the next step takes place. Since the next step is the joints' speeds computation, this means that the tasks could only be decoupled if the trajectories do not depend on the other robot's speeds.

For this example, the trajectory planning step for Denavit-Hartenberg method requires the combination of the part's trajectory with the paint job trajectory for the definition of the end-effector movement for the painter robot. The part trajectory is defined as a straight line translation, positioning the portion to be painted close to the painter robot initial pose, while it performs the painting. This avoids the part to be outside the workspace of the robots. The composition of the tasks for the painter job is defined as the sum of the trajectories, since there is no rotations defined for the positioning robot.

The screw theory's method, by the other hand, is composed by the following steps:

ISSN 2176-5480



1. Modelling of the robotic system with screws, building the $ screw matrix.

2. Modelling of the tasks with virtual chains, also modelled with screws.

3. Unify, by using the relationship graph, the network matrix, with every element, robot or tasks, modelled by screws.

4. Apply the Davies' method for the joints speed computation, as a function of the tasks to be performed.

Since there is no separation into two individual problems, the planning step can be task oriented. The network matrix construction defines the relationships between tasks and their effects over the robot's joints.

Besides their differences, both methods results in the exactly same joints trajectories for the robots, which is expected, since the methods are numerically equivalent (Ribeiro *et al.*, 2008). The joints trajectories can be seen in the Figure 10, in a single plot, since the results are the same for both methods. The plot shows the position of each joint relative to the reference position, numbered from the base (#1) to the end-effector (#6). By this point of view, both Denavit-Hartenberg and screws can take advantages of the potentiality of the multirobot system. It is, however, required to perform the trajectory planning with task merge (summing) in order to enable the Denavit-Hartenberg method for two robots.

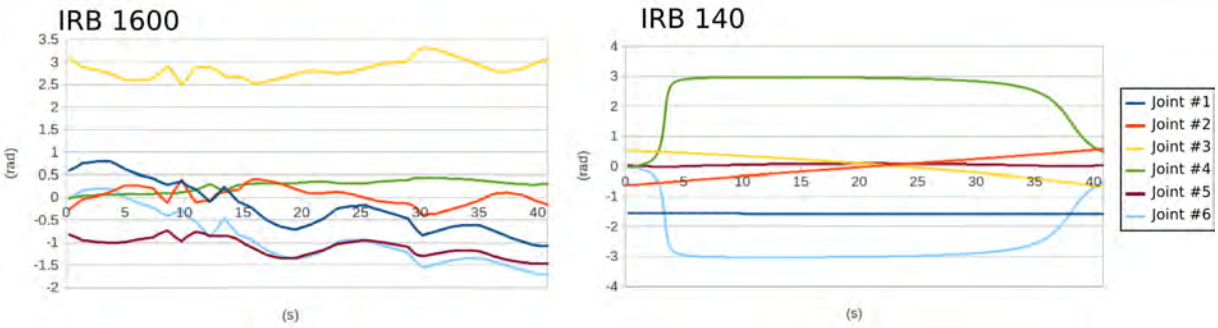


Figure 10. Computed joints position with straight line positioning and "UFSC" painting.

There are cases in which the Denavit-Hartenberg classical approach can not be applied. One example is the definition of the same paint job task, but with the following restriction: the positioning robot must pose the part into the workspace of the painter robot in the optimal way, without the straight line or any other predefined trajectory for the part. This implies that the robotic system must be capable of define which part displacement would be more adequate to the paint job. In this case, there is no possibility of application for Denavit-Hartenberg method, since the part trajectory (thus the end-effector trajectory) is not defined before the kinematics computation, it is integrated to the computation process.

The screw approach, by using virtual chains and the Davies' method, is capable of solving this problem directly. During the task planning, the task over the part is defined, and the part's trajectory can be assumed as secondary: it is defined a virtual chain for this movement, but no trajectory is defined *a priori*. After the network matrix computation, it is necessary to find a solution for the whole system and, since there is redundancy (one task to be performed by two robots), some optimization method can be applied to solve the system.

From the solution chosen by the optimization, a new kinematics resolution is found, and this solution is very different from the solution found by the predefined straight line part displacement, as can be seen in the Figure 11. The position of each joint on the plot is numbered relative to the reference position, from the base (#1) to the end-effector (#6). By predefining the part's trajectory the trajectory planner intends to find a good solution, but he/she has little control over the final performance of the system. By letting the resolution to take profit of the system's inherent redundancy, it is possible to find a solution with better performance.

This integrated resolution of movements, targeting the task completion is one of the main characteristics of Cooperative Multirobot Systems.

In the presented example the second scenario was created by changing the predefined straight line trajectory by the free trajectory to be defined by the computation. In this case, it is only needed to provide the specifications of the paint job. Since the part's trajectory is not defined, the classical Denavit-Hartenberg approach is not applicable. By using the screw theory and Davies' method, some very distinctive results were found. The straight line was exchanged by a complex spatial trajectory, which includes rotations of the part, minimizing the displacement and joints speeds in both robots.

In the predefined straight line trajectory, the joint speeds of the robots went as high as 56% of maximum joints speed, while in the trajectory computed by the cooperative system the joints speed went as high as 17% of the maximum speed. The performance gain is significant: the task was initially programmed to be performed in $40.8s$; with the straight line trajectory, this time can be reduced to $22.8s$, while with the computed trajectory by using the cooperative multirobot system strategy, the time can be reduced to $7.1s$.

ISSN 2176-5480



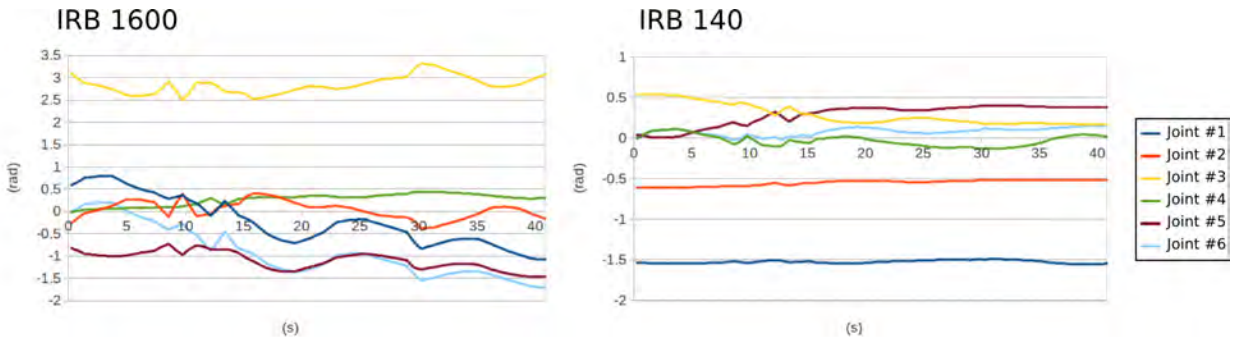


Figure 11. Joints position for the non predefined part trajectory.

In general, if some of the robots is redundant, the Denavit-Hartenberg approach would take the redundancy to help to solve the problem of kinematics computation for that robot, but yet, it needs a segmented solution for each robot. By the other hand, the screws approach and the Davies' method takes advantage of the redundancy for the whole system, enhancing the system's performance, not only the performance of the redundant robot. This is a very important advantage, since the source of the performance restriction can be in some of the non-redundant robots, and by ignoring the systemic improve of the redundancy, it wouldn't be possible for one robot to help another to perform the tasks.

## 5. Conclusion

This paper provides an comparison between the classical Denavit-Hartenberg method and the screw theory for the kinematics model definition in CMS, aiming the planning and programming of multirobot systems in order to perform tasks. The Denavit-Hartenberg is widely applied in the literature, being the standard choice for single robot task planning. It is possible to apply this method for multirobot systems, too, even thought its application requires some restrictive assumptions: the multiple robot system planning must be divided in multiple single robot planning; and the task specification must be adapted to end-effector trajectory planning, by means of a merge process. The screw approach simplifies the axis specifications, when compared to the Denavit-Hartenberg convention, and it presents several advantages, such as the task oriented planning, homogeneous representation of robots and tasks and integrated kinematics computation method, allowing a fully cooperative system resolution.

The Denavit-Hartenberg approach is more well-known, and is a systematical and structured methodology, but to fully take the advantages in cooperation system, it is needed to be better planned in order to compute the task planning, as shown before. More ideally, new approaches should be experimented, such as the Davies' method with virtual chains and screws.

The screw approach allows the use of optimization techniques and to perform tasks in a full cooperative way, taking profit from the redundancy of the cooperative system. There is performance gains, and methodological gains, since the task-oriented procedure is more intuitive and the virtual chains allows the modelling of several tasks in the same representation model given to the robots.

## 6. REFERENCES

Campos, A., Guenther, R. and Martins, D., 2005. "Differential kinematics of serial manipulators using virtual chains". *Brazilian Society of Mechanical Sciences and Engineering*, Vol. XXVII, No. 4, pp. 345 – 356. ABCM.

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. ISSN 0094-114X. doi:10.1016/0094-114X(81)90033-1.

Hunt, K.K., 2000. "Don't cross-thread the screw". In *A Symposium Commemorating the Legacy, Works, and Life of Sir Robert Stawell Ball Upon the 100th Anniversary of A Treatise on the Theory of Screws*. Ball 2000 Conference, University of Cambridge, Trinity College, pp. pp. 1–56.

Ribeiro, L., Guenther, R. and Martins, D., 2008. *Screw-based relative jacobian for manipulators cooperating in a task*, Vol. 3 of *ABCM Symposium Series Mechatronics*. ABCM – Associação Brasileira de Engenharia e Ciências Mecânicas, Brasil. ISBN 978-85-85769-38-3.

Rocha, C., Tonetto, C. and Dias, A., 2011. "A comparison between the denavit–hartenberg and the screw-based methods used in kinematic modeling of robot manipulators". *Robotics and Computer-Integrated Manufacturing*, Vol. 27, No. 4, pp. 723 – 728. ISSN 0736-5845. doi:10.1016/j.rcim.2010.12.009. URL `http://www.sciencedirect.com/science/article/pii/S073658451100010X`.

Siciliano, B., Sciavicco, L., Villani, L. and Oriolo, G., 2009. *Robotics: Modelling, Planning and Control*, Vol. XXIV of *Advanced Textbooks in Control and Signal Processing*. Springer, London. ISBN 978-1-84628-641-4.

ISSN 2176-5480



Spong, M.W., Hutchinson, S. and Vidyasagar, M., 2006. *Robot modeling and control*. John Wiley & Sons, New Jersey. ISBN 978-0-471-64990-8.

Tonetto, C., Rocha, C., Simas, H. and Dias, A., 2012. "Kinematics programming for cooperating robotic systems". In L. Camarinha-Matos, E. Shahamatnia and G. Nunes, eds., *Technological Innovation for Value Creation*, Springer Berlin Heidelberg, Vol. 372 of *IFIP Advances in Information and Communication Technology*, pp. 189–198. ISBN 978-3-642-28254-6. doi:10.1007/978-3-642-28255-3_21.

## 7. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.