INTECH
open science | open minds

OPEN ACCESS
ARTICLE

International Journal of Advanced Robotic Systems

Kinematic Modelling and Simulation of a 2-R Robot Using SolidWorks and Verification by MATLAB/Simulink

Regular Paper

Mahmoud Gouasmi[1,*], Mohammed Ouali[1], Brahim Fernini[1] and M'hamed Meghatria[1]

1 Algeria Structural Mechanics Research Laboratory, Mechanical Engineering Department, Blida University, Algeria
* Corresponding author E-mail: ygouasmi@hotmail.com

Received 11 Apr 2012; Accepted 26 May 2012

DOI: 10.5772/50203

Abstract The simulation of robot systems is becoming very popular, especially with the lowering of the cost of computers, and it can be used for layout evaluation, feasibility studies, presentations with animation and off-line programming.

The trajectory planning of redundant manipulators is a very active area since many tasks require special characteristics to be satisfied. The importance of redundant manipulators has increased over the last two decades because of the possibility of avoiding singularities as well as obstacles within the course of motion. The angle that the last link of a 2 DOF manipulator makes with the x-axis is required in order to find the solution for the inverse kinematics problem. This angle could be optimized with respect to a given specified key factor (time, velocity, torques) while the end-effector performs a chosen trajectory (i.e., avoiding an obstacle) in the task space.

Modeling and simulation of robots could be achieved using either of the following models: the geometrical model (positions, postures), the kinematic model and the dynamic model.

To do so, the modelization of a 2-R robot type is implemented. Our main tasks are comparing two robot postures with the same trajectory (path) and for the same length of time, and establishing a computing code to obtain the kinematic and dynamic parameters.

SolidWorks and MATLAB/Simulink softwares are used to check the theory and the robot motion simulation.

This could be easily generalized to a 3-R robot and possibly therefore to any serial robot (Scara, Puma, etc.).

The verification of the obtained results by both softwares allows us to qualitatively evaluate and underline the validityof the chosen model and obtain the right conclusions. The results of the simulations are discussed and an agreement between the two softwares is certainly obtained.

Keywords Multibody Systems, 2-R Robot Modelization, Kinematic Model, Dynamic Behaviour, Manipulability Simulation

1. Introduction

Robotics is a special engineering science which deals with robot design, modelling, controlling and utilization. Nowadays robots accompany people in everyday life and have taken over some of their daily procedures. The range of robot utilization is very wide, from toys through to office and industrial robots, through to very sophisticated ones needed for space exploration.

A large family of robot manufacturing equipment exists, supplying the motion required in manufacturing processes such as arc welding, spray painting, assembly, pick and place, cutting, polishing, milling, drilling, etc. Of this class of equipment, an increasingly popular type is the industrial robot. Different manipulator configurations are available such as rectangular, cylindrical, spherical, revolute and horizontal jointed.

Kinematically redundant manipulators have many advantages over non-redundant robots. A redundant manipulator is defined as having an infinite number of solutions to the joint variables of the manipulator for a given task. It can avoid obstacles and singularities, and is an excellent candidate for optimization techniques. Many optimization techniques have been applied to select the optimal trajectory of a manipulator using different criteria such as minimum time, minimum kinetic energy and obstacle avoidance.

A manipulation task is usually given in terms of a desired end-effector trajectory. Since the manipulator is controlled by joint servos, a mapping from the task space to the joint space is required. Trajectory planning converts a description of a desired motion to a trajectory defining the time sequence of intermediate configurations of the arm between the origin and the final destination. An effective approach to the motion control problem for robotic manipulators is the so-called kinematic control.

This is based on an inverse kinematic transformation which feeds the reference values corresponding to an assigned end-effector trajectory to the joint servos.

A vertical revolute configuration, a 2-R robot with two degrees of freedom is generally well-suited for small parts insertion tasks for assembly lines like electronic components insertion [2]. Although the final aim is real robotics, it is often very useful to perform simulations prior to investigations with real robots [1]. This is because simulations are easier to setup, less expensive, faster and more convenient to use. Building up new robot models and setting up experiments only takes a few hours. A simulated robotics setup is less expensive than real robots and real-world setups, thus allowing better design exploration. Simulations often run faster than real robots while all the parameters are easily displayed on screen [3].

The possibility to perform real-time simulations becomes particularly important in the later stages of the design process. The final design can be verified before one embarks on the costly and time consuming process of building a prototype [4].

The need for accurate and computationally efficient manipulator dynamics has been extensively emphasized in recent years. The modelling and simulation of robot systems by using various program softwares will facilitate the process of designing, constructing and inspecting the robots in the real world. A simulation is important for robot programmers in allowing them to evaluate and predict the behaviour of a robot, and in addition to verify and optimize the path planning of the process [5]. Moreover, this will save time and money, and play an important role in the evaluation of manufacturing automation [6]. Being able to simulate opens up a wide range of options, helping to solve many problems creatively. One can investigate, design, visualize and test an object before making it a reality [7].

In this work, a two axis 2-R robot system for a "pick and place" operation will be designed and developed using the SolidWorks program and MATLAB/Simulink simultaneously as shown in Figure 1 and Figure 2.

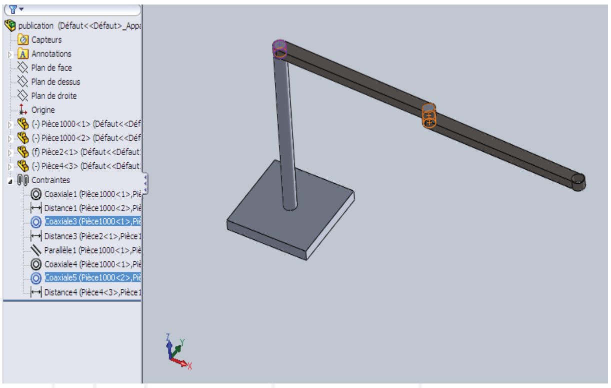

publication (Défaut<<Défaut>_Appa
Capteurs
Annotations
Plan de face
Plan de dessus
Plan de droite
Origine
(-) Pièce1000<1> (Défaut<<Déf
(-) Pièce1000<2> (Défaut<<Déf
(f) Pièce2<1> (Défaut<<Défaut
(-) Pièce4<3> (Défaut<<Défaut
Contraintes
Coaxiale1 (Pièce1000<1>,Piè
Distance1 (Pièce1000<2>,Piè
Coaxiale3 (Pièce1000<1>,Piè
Distance3 (Pièce2<1>,Pièce1
Parallèle1 (Pièce1000<1>,Piè
Coaxiale4 (Pièce1000<1>,Piè
Coaxiale5 (Pièce1000<2>,Piè
Distance4 (Pièce4<3>,Pièce1
Z
Y
X

Figure 1. A 2-R robot using SolidWorks

The structure will be built depending on the principles of solid bodies' modelling with SD technology [8, 9].

The individual joints' trajectories, velocities and accelerations can be obtained using inverse kinematics equations.

To emphasize the obtained results in the SolidWorks program, simulations using MATLAB/Simulink software will be carried out. The results of both sofwares will be presented and discussed. In the paper, the kinematic equations for a 2-R robot as well as the robot dynamics for each joint were developed using the D-H formulation.

The paper is organized as follows: in chapter 2 forward (direct) kinematics, inverse kinematics as well as the dynamics of the 2R robot are presented. In chapter 3 we present the simulation applications, implementing both softwares, and finally in the concluding chapter 4 some important results are shown.

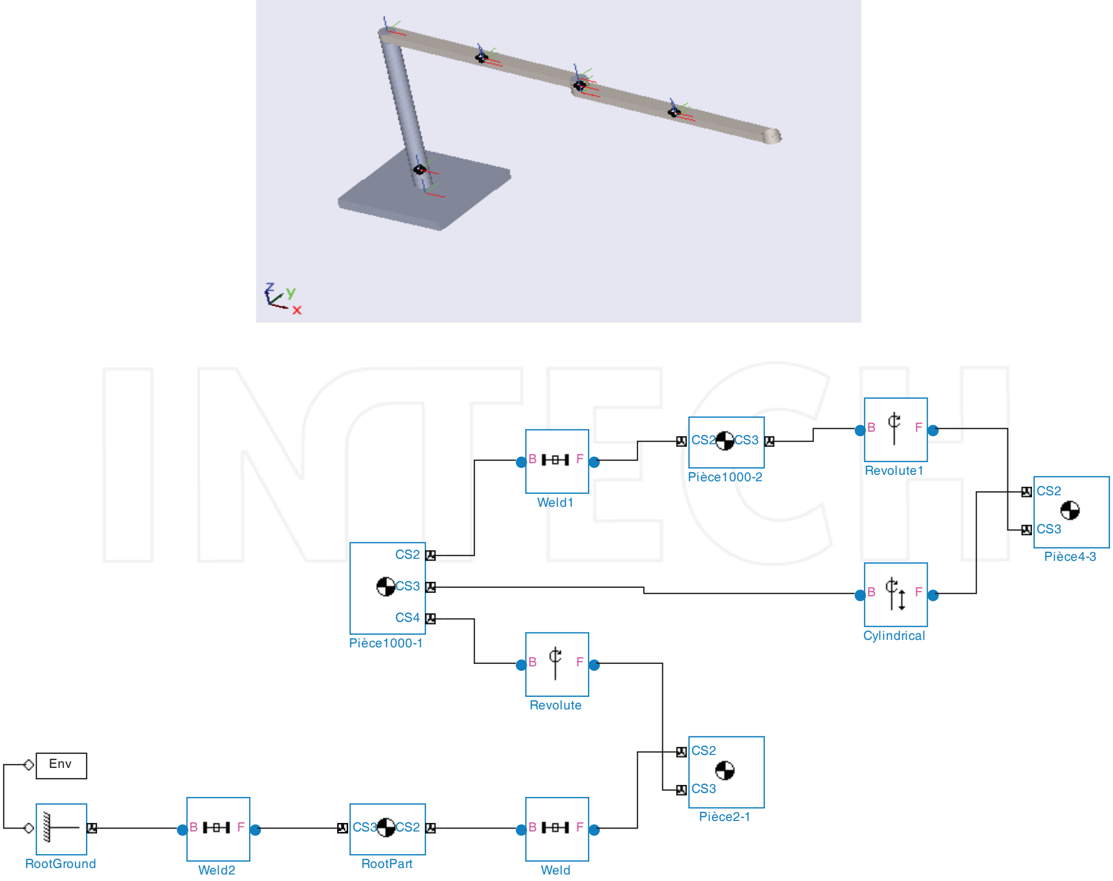


Figure 2. The same 2-R robot modelled in MATLAB/Simulink

2. Robot kinematics and dynamics

2.1 Forward kinematics

Consider the two DOF planar redundant manipulator as shown in Figure 3 where the joints' axes are assigned based on the Denavit-Hartenberg representation (Table 1). The manipulator moves in a vertical plane so the gravity force is included in the numerical simulation.

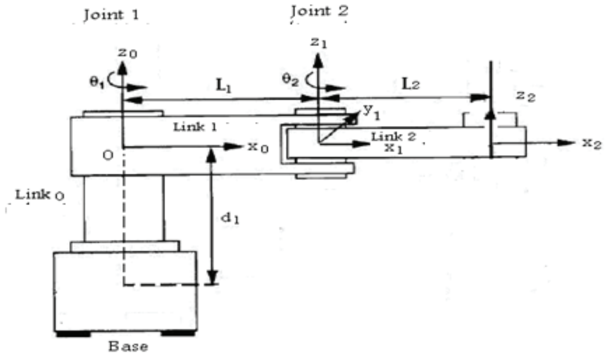


Figure 3. D-H parameters of the two-joint 2-R robot

Link	a_i	α_i	d_i	θ_i
1	l_1	0	0	θ_1^*
2	l_2	0	0	θ_2^*

Table 1. D-H parameters of the 2-R robot

The (4x4) rigid transformation matrices are:

For the first link:

$$T_1^0 = \begin{bmatrix} c_1 & -s_1 & 0 & l_1c_1 \\ s_1 & c_1 & 0 & l_1s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

and the second link:

$$T_2^1 = \begin{bmatrix} c_2 & -s_2 & 0 & l_2c_2 \\ s_2 & c_2 & 0 & l_2s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

We can then get the overall manipulator transformation matrix:

$$T_2^0 = \begin{bmatrix} c_{12} & -s_{12} & 0 & l_1c_1 + l_2c_{12} \\ s_{12} & c_{12} & 0 & l_1s_1 + l_2s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

2.2 Inverse kinematics of the robot

The desired location of the 2-R robot is given by the (4x4) matrix:

$$T_H^R = \begin{pmatrix} n_X & o_X & a_X & p_X \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4}$$

For the joint variable θ_2 :
The final equation representing the robot is:[10]:

$$T_2^0 = T_H^R \tag{5}$$

We get:

$$p_X = l_1c_1 + l_2c_{12} \tag{6}$$

$$p_y = l_1s_1 + l_2s_{12} \tag{7}$$

We then have:

$$c_2 = \frac{1}{2l_1l_2}({p_X}^2 + {p_y}^2 - l_1^2 - l_2^2) \tag{8}$$

$$s_2 = \pm\sqrt{1 - {c_2}^2} \tag{9}$$

$$\theta_2 = \pm a\tan\frac{s_2}{c_2} \tag{10}$$

The joint variable θ_1 :

Rearranging equation (6) and equation (7) yields:

$$p_X = (l_1 + l_2c_2)c_1 - l_2s_2s_1 \tag{11}$$

$$p_y = l_2s_2c_1 + (l_1 + l_2c_2)s_1 \tag{12}$$

Solving equations (11) and (12) by Cramer's rule:

$$\Delta = \begin{bmatrix} l_1 + l_2c_2 & -l_2s_2 \\ l_2s_2 & l_1 + l_2c_2 \end{bmatrix} = (l_1 + l_2c_2)^2 + (l_2s_2)^2 \tag{13}$$

and we have:

$$p_X^2 + p_y^2 = (l_1 + l_2c_2)^2 + (l_2s_2)^2 \tag{14}$$

$$\Delta s_1 = \begin{bmatrix} l_1 + l_2c_2 & p_x \\ l_2s_2 & p_y \end{bmatrix} \tag{15}$$

$$\Delta c_1 = \begin{bmatrix} p_x & -l_2s_2 \\ p_y & l_1 + l_2c_2 \end{bmatrix} \tag{16}$$

$$s_1 = \frac{\Delta s_1}{\Delta} = \frac{(l_1 + l_2c_2)p_y - l_2s_2p_x}{p_X^2 + p_y^2} \tag{17}$$

$$c_1 = \frac{\Delta c_1}{\Delta} = \frac{(l_1 + l_2c_2)p_x + l_2s_2p_y}{p_X^2 + p_y^2} \tag{18}$$

and finally we have:

$$\theta_1 = a\tan\frac{s_1}{c_1} = a\tan\frac{(l_1 + l_2c_2)p_y \mp l_2s_2p_x}{(l_1 + l_2c_2)p_x \pm l_2s_2p_y} \tag{19}$$

The sign (±) indicates the two postures, "elbow up" and "elbow down", of the manipulator.

The angles for elbow up are:

$$\theta_2 = -a\tan\frac{s_2}{c_2} \tag{20}$$

$$\theta_1 = a\tan\frac{+p_xl_2s_2 + p_y(l_1 + l_2c_2)}{p_x(l_1 + l_2c_2) - (p_yl_2s_2)} \tag{21}$$

and the angles for elbow down are:

$$\theta_2 = +a\tan\frac{s_2}{c_2} \tag{22}$$

$$\theta_1 = a\tan\frac{-p_xl_2s_2 + p_y(l_1 + l_2c_2)}{p_x(l_1 + l_2c_2) + (p_yl_2s_2)} \tag{23}$$

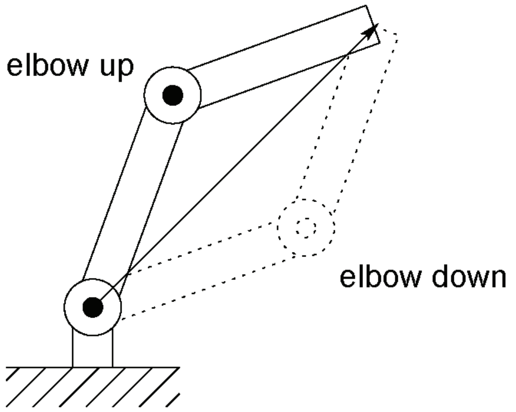


Figure 4. The two postures of the 2-R robot

2.3 Robot dynamics

2.3.1 Kinetic energy

$$K_1 = \frac{1}{2} m_1 l_1^2 \dot{\theta}_1^2 \qquad (24)$$

$$K_2 = \frac{1}{2} m_2 l_1^2 \dot{\theta}_1^2 + \frac{1}{2} m_2 l_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + m_2 l_1 l_2 (\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2) \cos\theta_2 \qquad (25)$$

2.3.2 Potential energy

$$V_1 = m_1 g l_1 \sin\theta_1 \qquad (26)$$

$$V_2 = m_2 g y_2 = m_2 g [l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2)] \qquad (27)$$

2.3.3 Joints torques

Using the Newton-Euler recursive method we can calculate the necessary torques for each joint:

$$\begin{aligned} \tau_1 &= m_2 l_2^2 (\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 l_1 l_2 c_2 (2\ddot{\theta}_1 + \ddot{\theta}_2) \\ &+ (m_1 + m_2) l_1^2 \ddot{\theta}_1 - m_2 l_1 l_2 s_2 \dot{\theta}_2^2 - 2 m_2 l_1 l_2 s_2 \dot{\theta}_1 \dot{\theta}_2 + \\ &+ m_2 l_2 g c_{12} + (m_1 + m_2) l_1 g c_1 \end{aligned} \qquad (28)$$

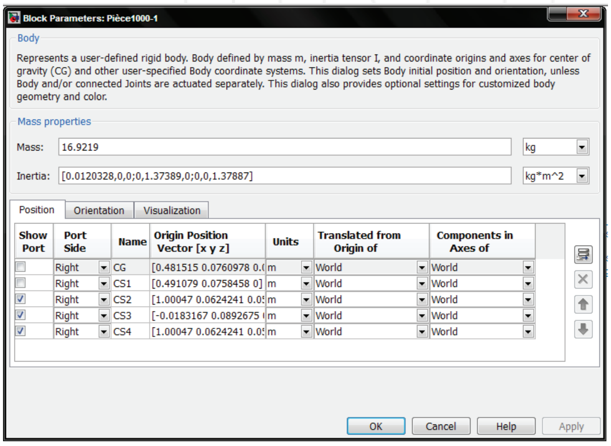
$$\tau_2 = m_2 l_1 l_2 c_2 \ddot{\theta}_1 + m_2 l_1 l_2 s_2 \dot{\theta}_1^2 + m_2 l_2 g c_{12} + m_2 l_2^2 (\ddot{\theta}_1 + \ddot{\theta}_2) \qquad (29)$$

3. Simulation results

3.1 Application

This application will bear the energetic comparison during the movement of the 2-R robot with $m_1 = m_2 = 16.92 Kg$ and $l_1 = l_2 = 1m$, for the two postures.

We can easily follow and observe the movements of the two postures “elbow down” and “elbow up” for the same nature of the trajectory (path) , reaching the same desired position during the same interval of time of 1s as shown in the following diagram:



Block Parameters: Pièce1000-1

Body

Represents a user-defined rigid body. Body defined by mass m, inertia tensor I, and coordinate origins and axes for center of gravity (CG) and other user-specified Body coordinate systems. This dialog sets Body initial position and orientation, unless Body and/or connected Joints are actuated separately. This dialog also provides optional settings for customized body geometry and color.

Mass properties

Mass: 16.9219 kg

Inertia: [0.0120328,0,0;0,1.37389,0;0,0,1.37887] kg*m^2

Position | Orientation | Visualization

Show Port	Port Side	Name	Origin Position Vector [x y z]	Units	Translated from Origin of	Components in Axes of
□	Right	CG	[0.481515 0.0760978 0.(	m	World	World
□	Right	CS1	[0.491079 0.0758458 0]	m	World	World
☑	Right	CS2	[1.00047 0.0624241 0.0!	m	World	World
☑	Right	CS3	[-0.0183167 0.0892675 (	m	World	World
☑	Right	CS4	[1.00047 0.0624241 0.0!	m	World	World

Figure 5. Link:mass parameters

The initial position (at t = 0) from the homogeneous transformation matrix (3) where $\theta_1 = 0°$ and $\theta_2 = 0°$ as shown in the following figures:

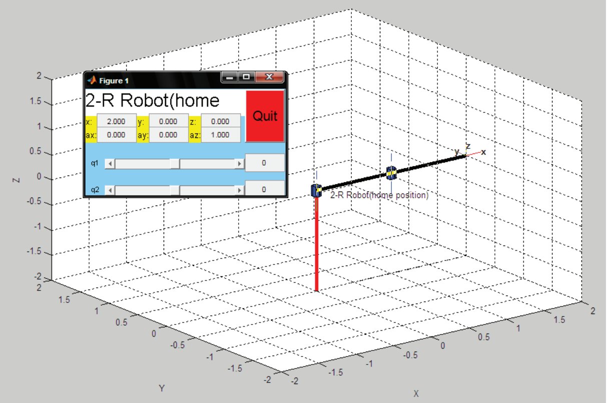
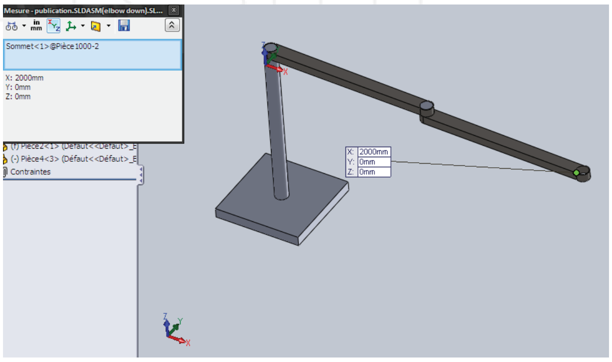


Figure 6. The home position of the 2-R robot (two postures): MATLAB/Simulink



Mesure - publication.SLDASM(elbow down).SL...
Sommet<1>@Pièce1000-2
X: 2000mm
Y: 0mm
Z: 0mm
(f) Pièce2<1> (Défaut<<Défaut>_E
(-) Pièce4<3> (Défaut<<Défaut>_E
Contraintes
X: 2000mm
Y: 0mm
Z: 0mm

Figure 7. The home position of the 2-R robot (two postures): SolidWorks

Starting from a given initial position, i.e.,

$$p_X = 2 \quad ; \quad p_y = 0$$

To a desired position, i.e,

$$p_X = 0.86 \quad ; \quad p_y = 1.5$$

The equations of movement for the borrowed path for “elbow down” to reach the desired positions are given by the two following chosen polynomials [11]:

$$\theta_1 = (10t^4 + 10t^3 + 10t^2)° \qquad (30)$$

$$\theta_2 = (50t^4 + 10t^3)° \qquad (31)$$

While the equations of movement for the followed trajectory for “elbow up” to reach the desired position will be given by the relations **[11]:**

$$\theta_1 = (60t^4 + 20t^3 + 10t^2)° \qquad (32)$$

$$\theta_2 = (-50t^4 - 10t^3)° \qquad (33)$$

The choice of expressions of θ_1 and θ_2 must satisfy the following two conditions [11]:

- In the meantime (interval time) must verify the initial and the desired position.
- At every moment the two postures (elbow down and up) must have the same coordinates P_x and P_y.

NB: Another example of trajectory generation is given in the Appendix at the end of this paper.

3.1.1 The resulting path

We may check the path:

Applying the relations (20), (21), (22) and (23) in (6) and (7) we can obtain the following table:

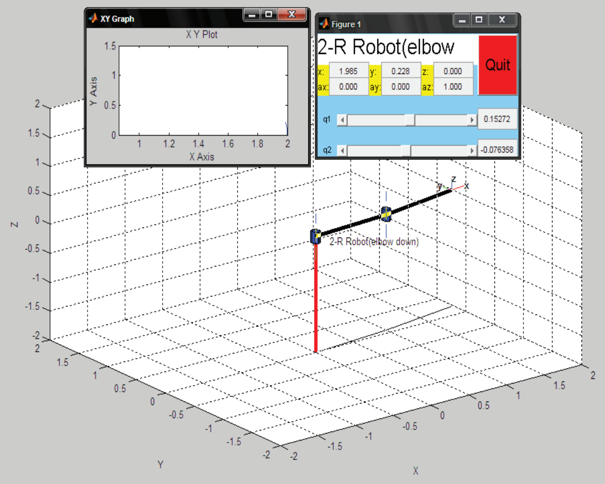
time (sec)	Elbow down				Elbow up			
	Theta1 (deg)	Theta2 (deg)	p_X (m)	p_y (m)	Theta1 (deg)	Theta2 (deg)	p_X (m)	p_y (m)
0	0	0	2	0	0	0	2	0
0.1	0.11	0.015	1.99	0.004	0.126	-0.015	1.99	0.004
0.2	0.49	0.16	1.99	0.02	0.65	-0.16	1.99	0.02
0.3	1.25	0.67	2	0.05	1.92	-0.67	2	0.05
0.4	2.49	1.92	1.996	0.12	4.41	-1.92	1.996	0.12
0.5	4.37	4.37	1.98	0.21	8.75	-4.37	1.98	0.21
0.6	7.05	8.64	1.95	0.39	15.69	-8.67	1.95	0.39
0.7	10.73	15.43	1.88	0.62	26.16	- 15.43	1.88	0.62
0.8	15.61	25.6	1.71	0.927	41.21	-25.6	1.71	0.927
0.9	20	40.09	1.44	1.20	59.04	-40.09	1.44	1.20
1	30	60	0.86	1.5	90	-60	0.86	1.5

Table 2. The position for the two postures during 1s

3.1.2 Checking the theoretical results of Table 2 using SolidWorks and MATLAB/Simulink

At t=0.5s

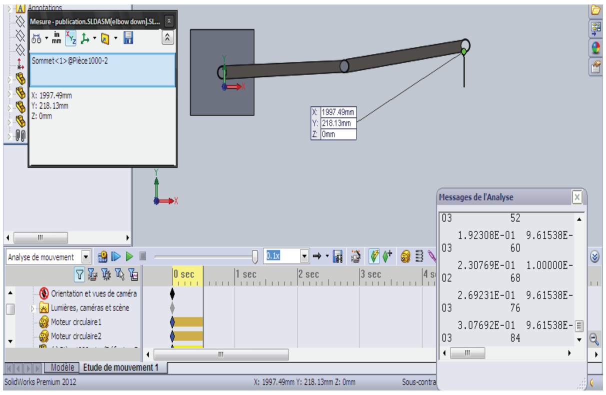
Elbow down;



2-R Robot(elbow

x: 1.985 y: 0.228 z: 0.000
ax: 0.000 ay: 0.000 az: 1.000
q1 0.15272
q2 -0.076358

Figure 9. Elbow down (MATLAB/Simulink)



Mesure - publication.SLDASM(elbow down).SL...

Sommet<1>@Pièce1000-2

X: 1997.49mm
Y: 218.13mm
Z: 0mm

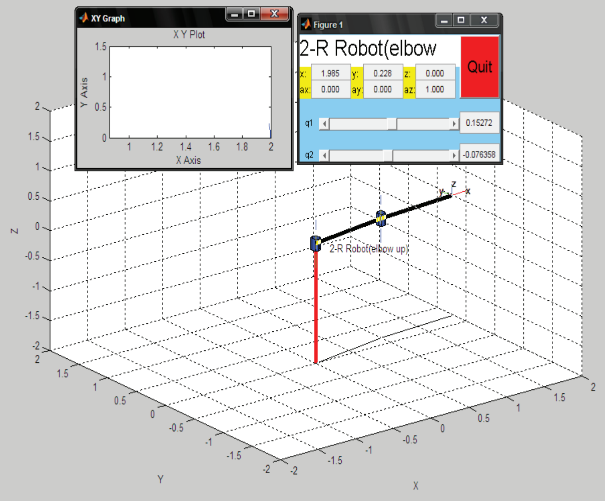
Messages de l'Analyse

```
03              52
    1.92308E-01  9.61538E-
03              60
    2.30769E-01  1.00000E-
02              68
    2.69231E-01  9.61538E-
03              76
    3.07692E-01  9.61538E-
03              84
```

SolidWorks Premium 2012   X: 1997.49mm Y: 218.13mm Z: 0mm

Figure 10. Elbow down (SolidWorks)

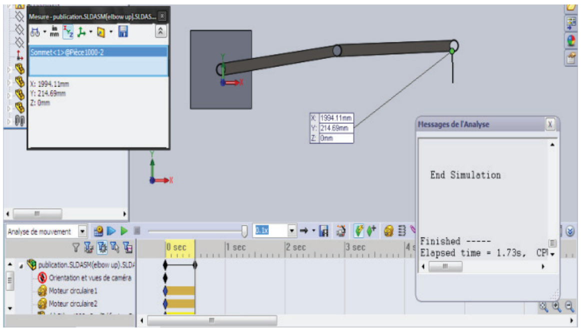
Elbow up:



2-R Robot(elbow

x: 1.985 y: 0.228 z: 0.000
ax: 0.000 ay: 0.000 az: 1.000
q1 0.15272
q2 -0.076358

Figure 11. Elbow up (MATLAB/Simulink)



Mesure - publication.SLDASM(elbow up).SLDAS...

Sommet<1>@Pièce1000-2

X: 1994.11mm
Y: 214.69mm
Z: 0mm

Messages de l'Analyse

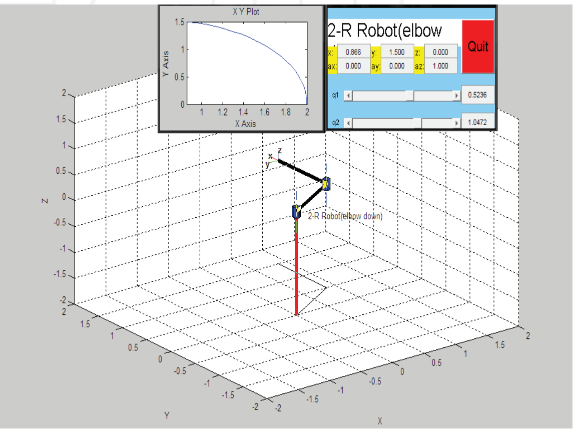
```
End Simulation

Finished -----
Elapsed time = 1.73s,  CP
```

Figure 12. Elbow up (SolidWorks)

and at t=1s

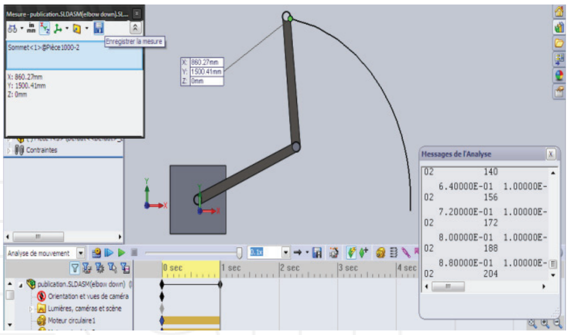
Elbow down;



2-R Robot(elbow

x: 0.866 y: 1.500 z: 0.000
ax: 0.000 ay: 0.000 az: 1.000
q1 0.5236
q2 1.0472

Figure 13. Elbow down (MATLAB/Simulink)



Mesure - publication.SLDASM(elbow down).SL...

Sommet<1>@Pièce1000-2

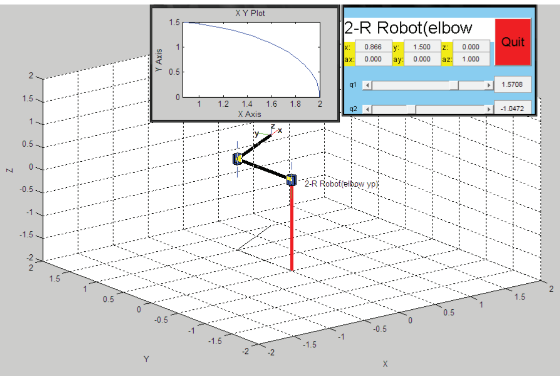
X: 860.27mm
Y: 1500.41mm
Z: 0mm

Messages de l'Analyse

```
02              140
    6.40000E-01  1.00000E-
02              156
    7.20000E-01  1.00000E-
02              172
    8.00000E-01  1.00000E-
02              188
    8.80000E-01  1.00000E-
02              204
```

Figure 14. Elbow down (SolidWorks)

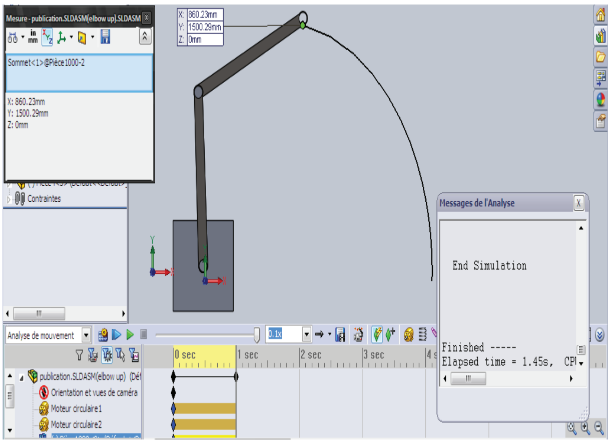
Elbow up;



2-R Robot(elbow

x: 0.866 y: 1.500 z: 0.000
ax: 0.000 ay: 0.000 az: 1.000
q1 1.5708
q2 -1.0472

Figure 15. Elbow up (MATLAB/Simulink)



Sommet<1>@Pièce1000-2

X: 860.23mm
Y: 1500.29mm
Z: 0mm

X: 860.23mm
Y: 1500.29mm
Z: 0mm

```
End Simulation

Finished -----
Elapsed time = 1.45s,  CPl
```

Figure 16. Elbow up (SolidWorks)

The joint variables expressed in MATLAB/Simulink are in radians.

3.1.3 Individual change of position of the two links for the two postures during 1s.

We can take advantage of this analysis using SolidWorks to study the behaviour of the robot for the two postures to establish an energetic balance comparative to these postures.

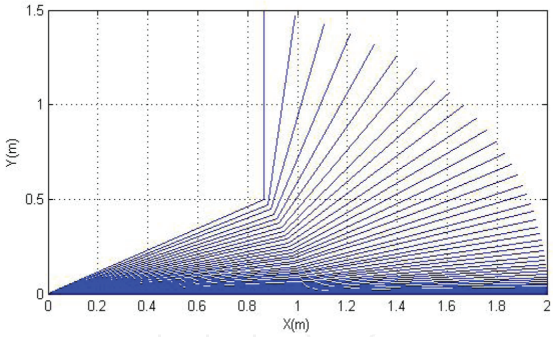


Figure 17. Elbow down

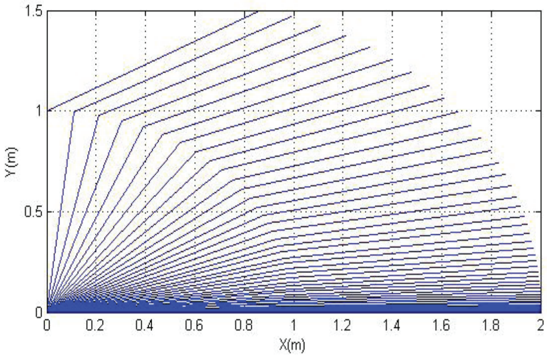


Figure 18. Elbow up

NOTE 1: The results obtained, whether using SolidWorks or MATLAB/Simulink, are exactly the same. This similarity of results confirms the reliability of the kinematic model.

3.1.4 The displacement of centre of gravity for the two postures

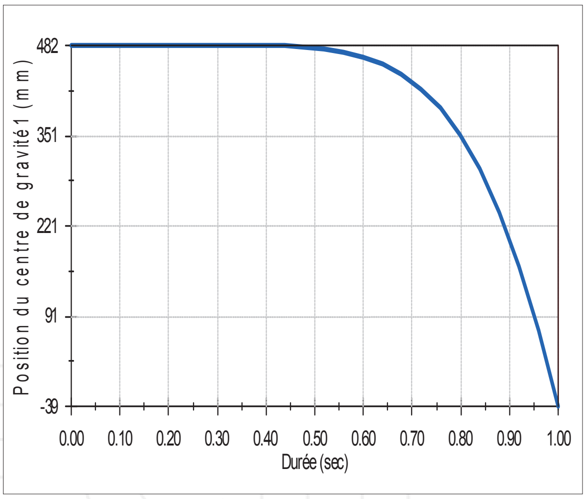


Figure 21. Link 1(x) elbow up

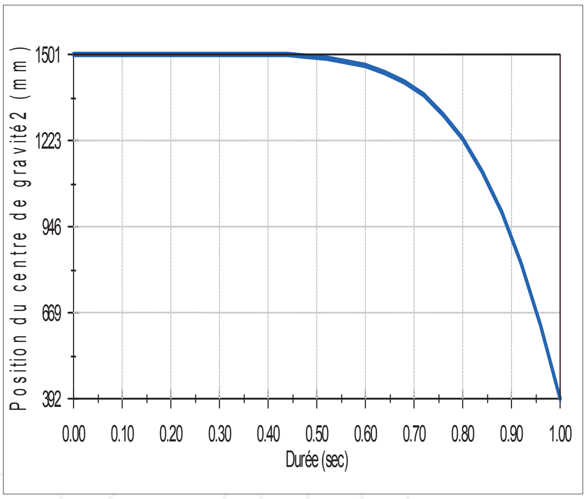


Figure 22. link 2(x) elbow up

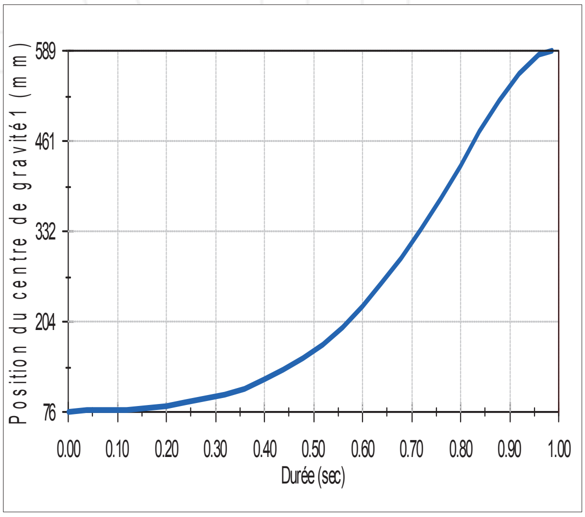


Figure 23. Link 1(y) elbow up

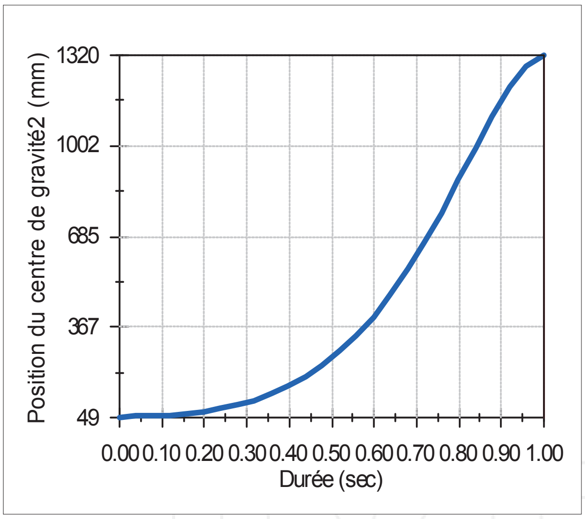


Figure 24. link 2(y) elbow up

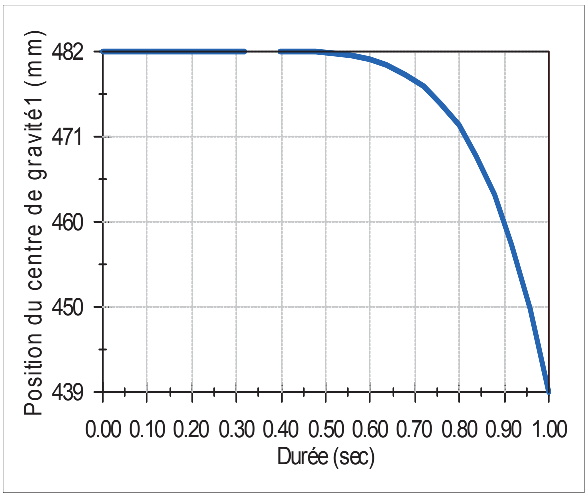


Figure 25. Link 1(x) elbow down

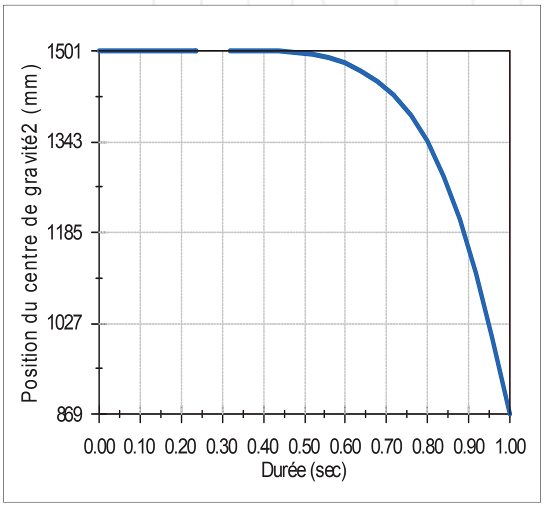


Figure 26. link 2(x) elbow down

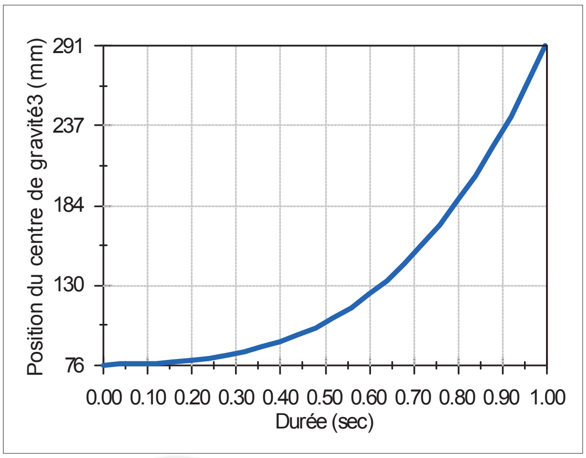


Figure 27. Link 1(y) elbow down

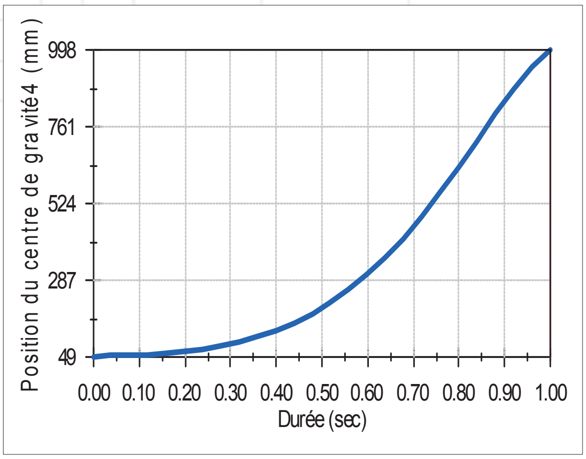


Figure 28. Link 2(y) elbow down

NOTE 2: According to the results, we notice that the displacement and velocity of the links 1 and 2 of elbow up are higher than the displacement and velocity of the links 1 and 2 of elbow down, this difference can induce a difference in the consumed energy for the two postures, for the same trajectory and the same desired position.

3.1.5 Simulation of dynamic model

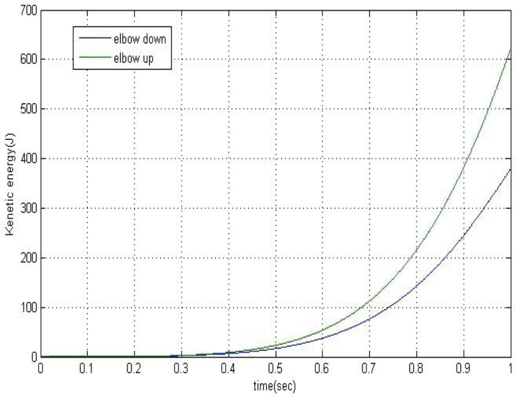


Figure 29. Kinetic energy

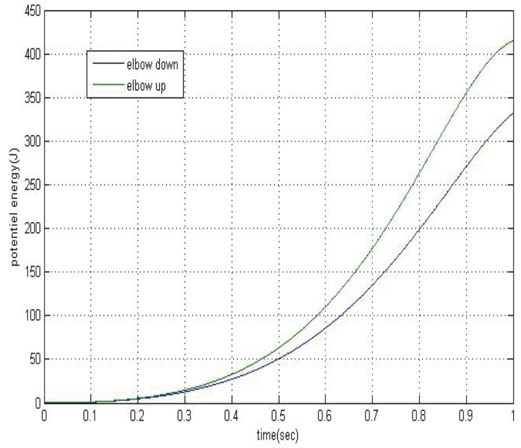


Figure 30. Potential energy

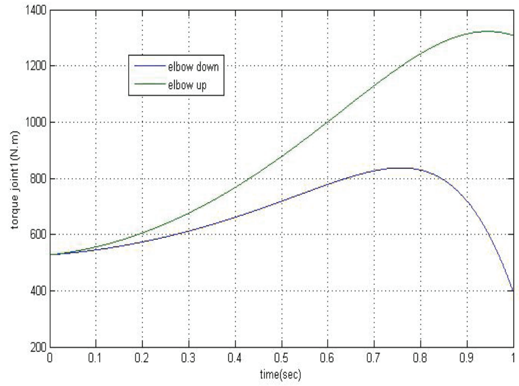


Figure 31. Joint torque (1)

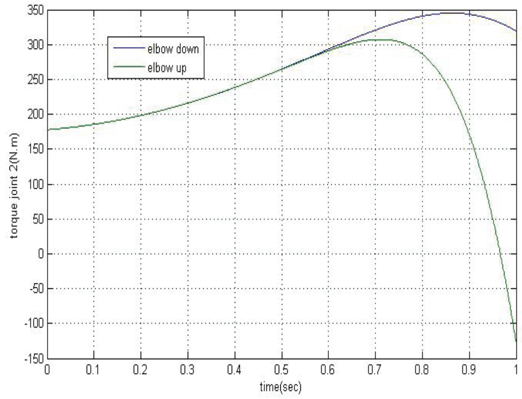


Figure 32. Joint torque (2)

NOTE 3: Comparison between the two postures:

We notice from the simulation results that:

-The variation of the total kinetic energy calculated according to posture elbow up is higher than the variation of the total kinetic energy calculated for posture elbow down with: $\Delta K = 243.02J$

-The variation of the total potential energy calculated according to posture elbow up is higher than the variation of the total potential energy calculated according to posture elbow down with: $\Delta V = 82.95J$.

We notice also that the torque of joint 1 calculated according to posture elbow up at t=1s is $\tau_1 = 446.33Nm$ with a max $\tau_1 = 900Nm$,on the other hand the torque of joint 1 calculated for posture elbow down is $\tau_1 = 395.18Nm$ with a max $\tau_1 = 820Nm$

Otherwise the torque of joint 2 calculated according to posture elbow up at t=1s is $\tau_2 = -127Nm$ with a max $\tau_2 = 300Nm$,while the torque of joint 2 calculated according to posture elbow down is $\tau_2 = 319.65Nm$ with a max $\tau_2 = 330Nm$

NOTE 4: The robot consumes much more energy when it is requested to work according to elbow up, for the same trajectory and the same desired position according to elbow down; [10] [11][12]

4. Conclusion

In this paper we used both softwares SolidWorks and MATLAB/Simulink to verify the theory (Table 2) and the simulation of the robot motion.

Checking the results obtained by the two softwares, SolidWorks and MATLAB/Simulink, permitted us to qualitatively develop and highlight the relevance of the studied model.

This can be easily justified when bearing in mind that the effect of different combinations of the two angles will play a major role in determining the value of the potential energy of the second link, and the potential energy of a link will play the greatest role in determining the torque value of that link.

In our case, we can conclude (depending on the obtained results by the comparative analysis of the two postures) that the dissipated energies during the movement of the robot according to the first posture (elbow down) are lower than those which would be dissipated if the robot performs the movement according to the second posture (elbow up), and this is for the same trajectory and the same duration according to the presentations listed above. See notes 1,2,3 and 4.

Appendix

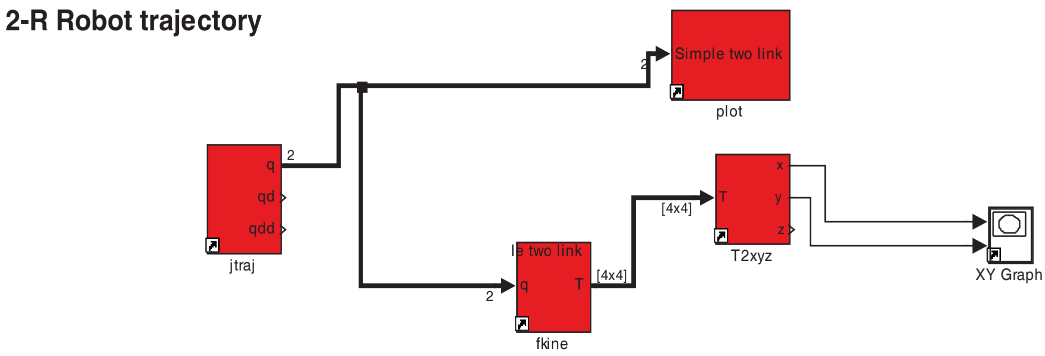


Figure 33. Trajectory generation of 2-R robot with MATLAB/Simulink

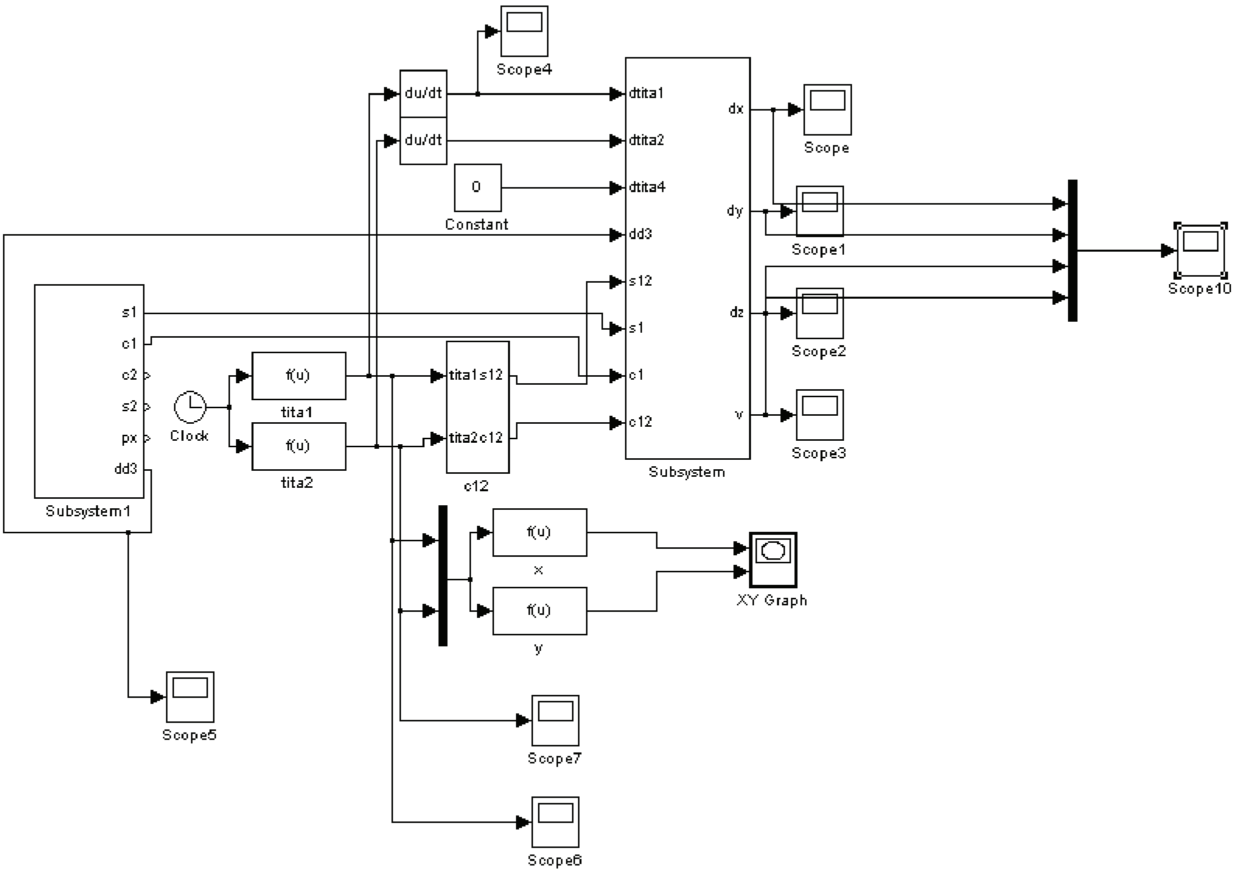
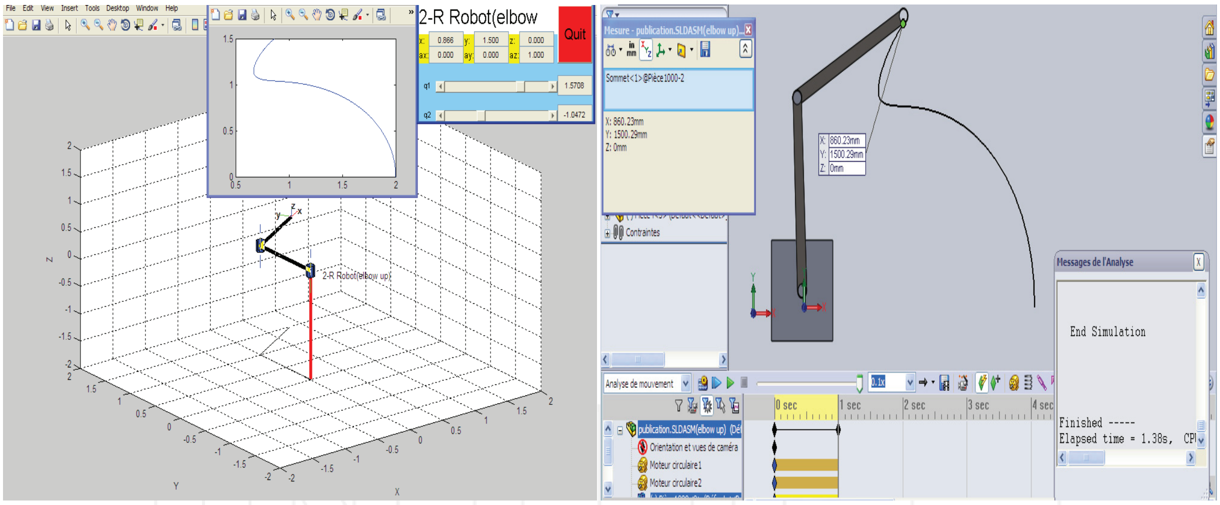


Figure 34. Simulation block to calculate the trajectory

Some trajectories obtained in the displacements of the two postures for the same desired position during 1s:

Elbow up: [11]	Elbow down: [11]
$\theta_1 = (180t - 20t^2 - 70t^5)^0$	$\theta_1 = (90t^{10} - 60t^2)^0$
$\theta_2 = (-180t + 20t^2 + 100t^5)^0$	$\theta_2 = (-90t^{10} + 150t^2)^0$
$l_1 = l_2 = 1m$	$l_1 = l_2 = 1m$

Simulation of the motion

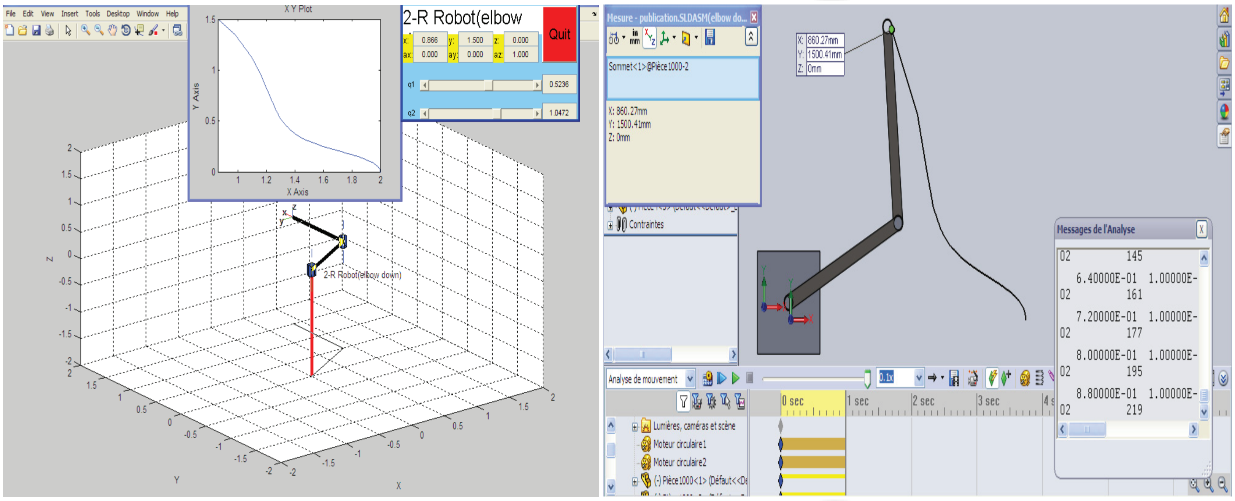


```
End Simulation

Finished -----
Elapsed time = 1.38s,  CP
```

(a) (b)

Figure 35. (a) Elbow up (MATLAB/Simulink); (b) Elbow up (SolidWorks)



```
02              145

   6.40000E-01  1.00000E-
02              161

   7.20000E-01  1.00000E-
02              177

   8.00000E-01  1.00000E-
02              195

   8.80000E-01  1.00000E-
02              219
```

(a) (b)

Figure 36. (a) Elbow down (MATLAB/Simulink); (b) Elbow down (SolidWorks)

5. References

[1] P. Sorenti, 1997. "Efficient Robotic Welding in Shipyards – Virtual Reality Simulation Holds the Key", Industrial Robot, 24(4), pp.278-281

[2] M.T. Das and L.C. Dulger, 2005. "Mathematical modelling, simulation and experimental verification of a scara robot", Simulation Modelling Practice and Theory 13, pp.257–271

[3] O. Michel, 2004. "Professional Mobile Robot Simulation, International Journal of Advanced Robotic Systems", Vol.1, No.1, pp.39-42 ISSN 1729-8806.

[4] A. Kazi, G. Merk, M. Otter and H. Fan, 2002. "Design optimization of industrial robots using the Modelica multi-physics modeling language", Proceedings of the 33rd ISR (International Symposium on Robotics) October 7 – 11.

[5] F. Ionescu, F. Chojnowski and G. Constantin, 2002. "Virtual Reality in Mechanical Engineering, Modelling and Simulation with Solid Dynamics", ARA-Journal, vol. 2002. No. 27

[6] F. Ionescu and D. Stefanoiu, 2002. "HYPAS - A Modular Structured Model Design, Simulation and Control Programming Environment", Proceed. of the IASTED Intern. Confer. On Artificial and Computational Intelligence, Sept. 25 - 27, Tokyo, Japan, Acta Press, ISBN: 0 - 88986 -358 - X, ISSN: 1482 - 7913, pp. 324 – 329

[7] L. Zlajpah, 2008. "Simulation in robotics", ScienceDirect, Mathematics and Computers in Simulation 79, pp. 879-897

[8] F. Ionescu, D. Stefanoiu and C.I Vlad, 2002. "Modular Structured Model Design, Simulation and Control of Hydraulic and Pneumatic Drive Systems by Using HYPAS", ARA-Journal, Volume

2000-2002, No. 25-26, ISBN 3-00-011583-8, pp. 168 - 177.

[9] F. Ionescu, 2007. “Modeling and Simulation in Mechatronics”, IFAS inter.confer.MCPL2007, Sep. pp.26-29, Sibiu, Romania.

[10] John Wiley & Sons, Inc. Robot modeling and control. New York / Chichester / Weinheim / Brisbane / Singapore / Toronto page 97

[11] R. N. Jazar, 2009,Theory of applied robotics (kinematics, dynamics and control) second edition Springer,new York (path planning) ISBN 978-1-4419-1749-2 e-ISBN 978-1-4419-1750-8 DOI 10.1007/978-1-4419-1750-8 Springer New York Dordrecht Heidelberg London Springer Science+Business Media, LLC 2006, 2010

[12] B. Siciliano, L. Sciavicco, G. Villani, G. Oriolo: “Robotics: Modelling, Planning and Control”, Springer, 2009 (3rd Edition)