ROBOT MANIPULATOR DESIGN ASSIGNMENT

Souto T.L.

MASSEY UNIVERSITY
TE KUNENGA KI PŪREHUROA

In this paper is reported the design of a robotic manipulator with a fixed platform in a flat surface, It's able to be integrated with a robotic gripper also designed. The objective of this robotic system is to pick an object from a shelf or from the wall and place it onto a horizontal surface. Several tools were used to accomplish the objective of this project. For the calculations of forward and inverse kinematics Python programming language and the Pycharm IDE(Integrated Development Environment) were used, for modelling the robotic system SolidWorks, to simulate Mathworks Simulink and OpenModelica were chosen.

***Keywords*—robotic systems, forward kinematics, inverse kinematics**

This paper is a Individual design Project and It is part of the second assignment of the course Master of engineering - Mechatronics at Massey university, Auckland.

An electronic version of this report can be found by following this link, and the programming documentation can be found by following this link.

Thiago Lima Souto is a student register under the number 19044686 at Massey university. Questions, comments or communications can be addressed via email thiago.souto@yahoo.com.br

Contents

I. Introduction

Industrial robot systems as well as computer-aided design and manufacturing (CAD and CAM) are leading the industrial automation. [1]

The mechanical manipulator is the most important form of the industrial robot and the localization of objects in the three-dimensional space is one of the most important aspect of the mechanical manipulator. Links, parts, tools, other objects on the manipulator environment and the motion of these objects are the subjects of study of Kinematics, as well as all the geometrical and time-based properties of the motion, with no regards to the forces applied that causes it.

The two basic problems in the study of mechanical manipulation are forward and inverse kinematics, the first computes the position and orientation of the end-effector on the manipulator and the second calculates all possible sets of joint angles that could be used a given position and orientation.

Nowadays, CAD and CAM advanced software's are of easy access and used to design, simulate and calculate all that is necessary for modern robot design.

The main objective of this assessment is to design a robotic manipulator with a fixed platform and flat surface, that is able to be integrated with a robot gripper for picking an object vertical wall/shelf and placing it onto a horizontal surface.

To accomplish this objective a robot system is proposed after this introduction followed by the manipulator and other components design. The forward and inverse kinematics of the robot system are studied with manual calculations as well as computed calculations. A Model with correct dimensions and a simulation of the proposed robot are made using Solidworks and OpenModelica. Finally, the results are discussed and the report is concluded.

II. Robot System Initial Proposal

Aiming on the objective of designing a robot system, basically, capable of picking an object from a shelf and placing it onto a horizontal surface, the system proposed is a 6DOF (degrees of freedom) robot manipulator, consisting of a fixed base, a rotating base, two solid links and a toll, as shown in the Figure 1.

To control the joints four brushless AC motors are used. The motors have attached a magnetic absolute encoder and a integrated controller. At the end of the system there is a simple gripper making the robotic manipulator able to pickup an object and place it onto a horizontal surface.

III. Manipulator Design

A. Robot Arms

Link 1 and Link 2 constitute the "arms" of the manipulator, they are designed to maximize the joint angle reach for more flexibility. Also in the rotation base there are two cuts two maximize even further the arms reach as can be seen on Figure 2. In this picture, with the Wireframe view with hidden lines visible, the fourth motor located inside the base can be seen, It is hidden on Figure 12.

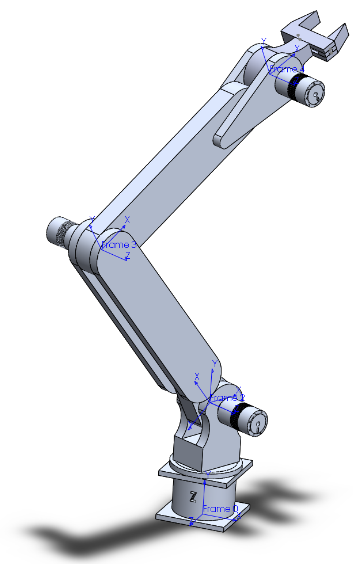

Fig. 1. Proposed robot manipulator system.

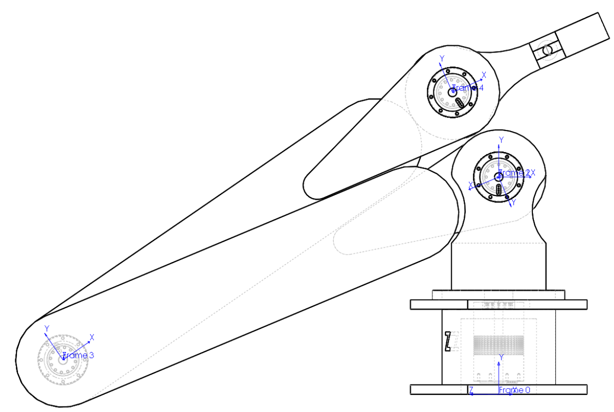

Fig. 2. Arms flexibility on Wireframe view.

B. Robot Gripper

A gripper capable to hold a 80 mm square object is connected to frame $\{W\}$. In one of the gripper's claws there is a micro-motor to enable the gripper to hold. One of the claws of the gripper is attached to a threaded cylinder and then attached to a micro-motor, which will rotate the threaded cylinder and make the claw move. Figure 3 shows the details of the gripper.

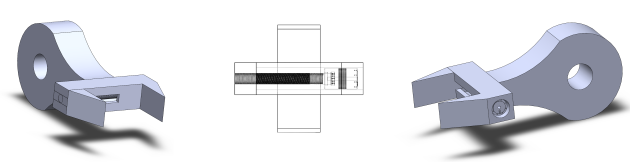

Fig. 3. Gripper details.

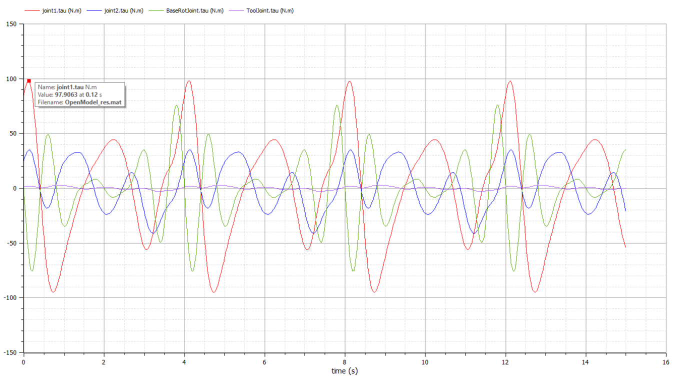

Fig. 4. Torque plot for various angles of the 4 joints, Max. 90.7566 N.m.

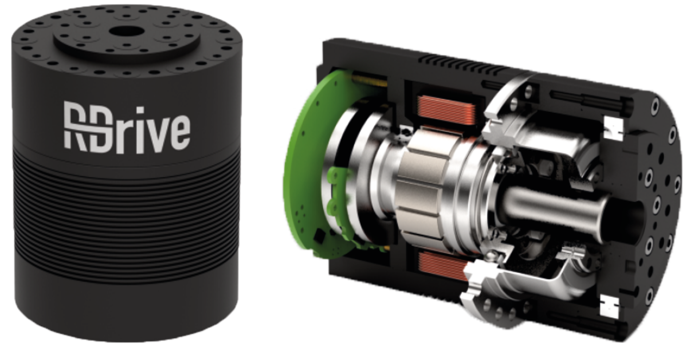

Fig. 5. RDrive motor.

C. Entire Model

Create a model of the robot manipulator with the correct dimensions. Using any method, software and hardware you are familiar with to simulate the movement (pick and place) of the robot manipulator. (3 marks)

IV. Other Components

Also, send a sheet of paper or PDF with complete contact information for all authors. Include full mailing addresses, telephone numbers, fax numbers, and e-mail addresses.

A. Motors

There are various types of motors and key factors need to be taken into account when selecting one for a particular application [3], in this case to control the joints of a robotic manipulator. The main factors are:

1) Inertia matching

The robotic system have to be capable to achieve a required torque to give a load a moment of particular inertia and to achieve a desired angular acceleration. The moment of inertia was calculated using the Iterative Newton-Euler Dynamics Algorithm [7], and this is solved in two parts, first the links velocities and accelerations are iteratively computed across the links applying the Newton-Euler equations to each link, then the forces and torques of interaction and joint actuator torques are computed recursively from the last link to the first.

This calculation were made using Python (Apenddix A) and confirmed by simulating the system using a simulation software, OpenModelica. The values for τ of each joint during the time of the simulation, $15s$, are shown on Figure 4, and, as can be noted, the maximum torque required was $97.9063N.m$. The simulation was made with the load attached to the system.

2) Torque requirements

High torque means a mechanism is able to handle heavier loads. The motor used for the modelling is capable of $157N.m$ and should be able to handle the $97.9063N.m$ with a $59.0937N.m$ margin.

3) Power requirements

As well as the torque requirements this project don't require that the motors run at maximum velocity, therefore overheating will not be a problem, and this is one of the main aspects of power requirements for a motor. The total power required is the sum of the power needed to overcome friction and that needed to accelerate the load [4].

After analysing and taking into consideration the aspects discussed above the RDrive 85 motor with rated torque of $108N.m$ and peak torque of $157N.m$ and $450W$ of Power was chosen to the task. [11] For the gripper a 20 mm diameter motor was used.

B. Sensors

To know the angular position of the joints absolute encoders shall be the choice because they give the actual angular position, a unique identification of an angle. The incremental encoders would detect the changes but in relation to some Datum. [3]

So with the absolute encoders we can track θ_1, θ_2, θ_3 and θ_4 and rearrange the links accordingly with the joints angles.

Also a loading cell can be used on the toll to sense the amount of pressure to set a trigger to avoid damage on the object.

C. Controllers

The final printed size of an author photograph is exactly 1 inch wide by 1 1/4 inches long (6 picas $\times$ 7 1/2 picas). Please ensure that the author photographs you submit are proportioned similarly. If the author's photograph does not appear at the end of the paper, then please size it so that it is proportional to the standard size of 1 9/16 inches wide by 2 inches long (9 1/2 picas $\times$ 12 picas). JPEG files are only accepted for author photos.

D. Control Methodology

IEEE accepts color graphics in the following formats: EPS, PS, TIFF, Word, PowerPoint, Excel, and PDF. The resolution of a RGB color TIFF file should be 400 dpi.

When sending color graphics, please supply a high quality hard copy or PDF proof of each image. If we cannot achieve a satisfactory color match using the electronic version of your files, we will have your hard copy scanned. Any of the files types you provide will be converted to RGB color EPS files.

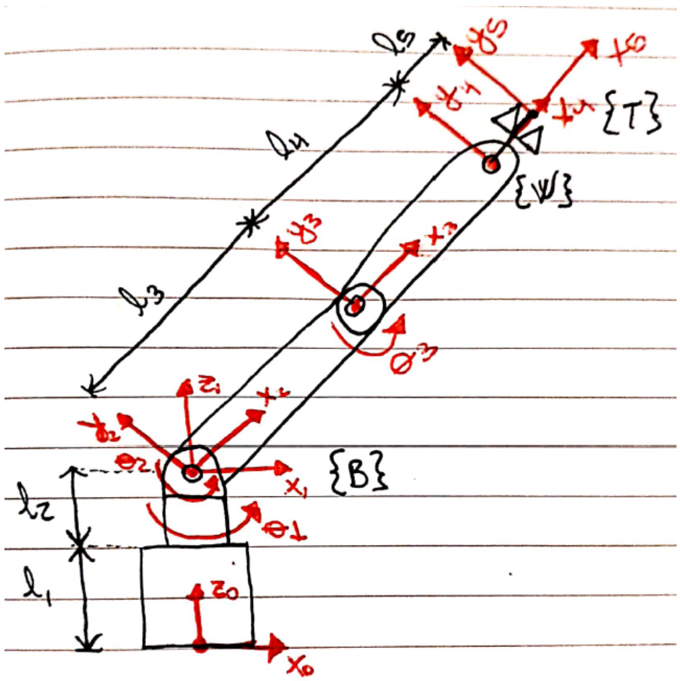

Fig. 6. $\hat{Z}, \hat{Y}, \hat{X} axis, \theta_1, \theta_2, \theta_3, l1, l2, l3$ and $l4$.

V. Forward Kinematic

To calculate the forward kinematics equation, a Python Class called "FowardKinematics" was created, this Class has two main methods involved on the calculations, the rotation and the translation for the $\hat{Z}$ and $\hat{X}$ axis. The parameters for these methods are extracted from the Denavit–Hartenberg parameters at (1), the coordinate systems and also the basic frames $\{B\}$, $\{W\}$ and $\{T\}$, Base, Wrist and Tools respectively are identified on the Figure 6. The size of the links are $l_1 = 230$, $l_2 = 500$, $l_3 = 500$, $l_4 = 180$.

The forward kinematics calculations were confirmed by a python programming code that can be found in the Appendix A.

The rotation method receives an argument $self$ and θ_i for the rotation on the $\hat{Z}$ axis and α_{i-1} for $\hat{X}$. The $self$ argument is what makes this a method and not just a plain function, this is filled in automatically, when we call this method on the object. So we'll just provide one argument, and the fact that it's being called on the method will provide the first argument, self. It will then build a $sympy$ symbolic matrix and passes the $self$ argument to the method to be put in place, if no arguments are passed default values will be put in place as specified in the key word arguments ($* \; kwargs$) on the $\_\_init\_\_$ function. A Matrix is then returned after calling the $.evalf()$ function to evaluate.

Like in the rotation method the translation method receives a argument d_i and a_{i-1} to return a matrix that translates in $\hat{Z}$ and $\hat{X}$ axis respectively. This class is also detailed in the Appendix A.

The objective of the forward kinematics is to provide a kinematics equation relating the end-effector orientation and position. This is done by finding the

Finally, the forward kinematics equation is presented on Equation 8.

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	l_1	θ_1
2	90°	0	0	θ_2
3	0	l_2	0	θ_3
4	0	l_3	0	θ_4
5	0	l_4	0	0

(1)

Denavit–Hartenberg parameters

$$ {}^0_1T = \begin{bmatrix} cos\theta_1 & -sin\theta_1 & 0 & 0 \\ sin\theta_1 & cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2} $$

$$ {}^1_2T = \begin{bmatrix} cos\theta_2 & -sin\theta_2 & 0 & 0 \\ -0.448sin\theta_2 & -0.448cos\theta_2 & -0.893 & 0 \\ 0.893sin\theta_2 & 0.893cos\theta_2 & -0.448 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3} $$

$$ {}^2_3T = \begin{bmatrix} cos\theta_3 & -sin\theta_3 & 0 & l_2 \\ sin\theta_3 & cos\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4} $$

$$ {}^3_4T = \begin{bmatrix} cos\theta_4 & -sin\theta_4 & 0 & l_3 \\ sin\theta_4 & cos\theta_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5} $$

$$ {}^4_5T = \begin{bmatrix} 1 & 0 & 0 & l_4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6} $$

$$ {}^0_5T = {}^0_1 T {}^1_2T {}^2_3T {}^3_4T {}^4_5T => {}^0 \cancel{{}_1T {}^1_2T {}^2_3T {}^3_4T {}^4} {}_5T = {}^0_5 T \tag{7} $$

VI. Inverse Kinematics

To find the joint displacements leading the centre of the end-effector from a vertical position to a horizontal position with correct orientation is necessary to obtain the inverse kinematics equations. So let Equation 9 be:

$$ {}^0_5T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & P_x \\ r_{21} & r_{22} & r_{23} & P_y \\ r_{31} & r_{32} & r_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9} $$

for θ_1, θ_2, θ_3 and θ_4, we have:

$$ {}^0_5T = {}^0_1 T(\theta_1) {}^1_2T(\theta_2) {}^2_3T(\theta_3) {}^3_4T(\theta_4) {}^4_5T \tag{10} $$

Equate r_{13} and r_{23} in relation to Equation 8 we have:
For r_{13}:

$$ 0.893sin(\theta_1) = r_{13} \tag{11} $$

For r_{23}:

$$ -0.893cos(\theta_1) = r_{23} \tag{12} $$

$$
{}^0_5T =
\begin{bmatrix}
\begin{matrix}0.448sin(\theta_1)sin(\theta_2+\theta_3+\theta_4)\\+cos(\theta_1)cos(\theta_2+\theta_3+\theta_4)\end{matrix} & \begin{matrix}0.448sin(\theta_1)cos(\theta_2+\theta_3+\theta_4)\\-sin(\theta_2+\theta_3+\theta_4)cos(\theta_1)\end{matrix} & 0.893sin(\theta_1) & \begin{matrix}224.036sin(\theta_1)sin(\theta_2)\\+80.653sin(\theta_1)\\sin(\theta_2+\theta_3+\theta_4)\\+500cos(\theta_1)cos(\theta_2)\\+180cos(\theta_1)\\cos(\theta_2+\theta_3+\theta_4)+\\362.018cos(-\theta_1+\theta_2+\theta_3)\\+137.981cos(\theta_1+\theta_2+\theta_3)\end{matrix} \\
\begin{matrix}sin(\theta_1)cos(\theta_2+\theta_3+\theta_4)-\\0.448sin(\theta_2+\theta_3+\theta_4)cos(\theta_1)\end{matrix} & \begin{matrix}-sin(\theta_1)sin(\theta_2+\theta_3+\theta_4)-\\0.448cos(\theta_1)cos(\theta_2+\theta_3+\theta_4)\end{matrix} & -0.893cos(\theta_1) & \begin{matrix}500sin(\theta_1)cos(\theta_2)+\\180sin(\theta_1)cos(\theta_2+\theta_3+\theta_4)\\-224.036sin(\theta_2)cos(\theta_1)-\\362.018sin(-\theta_1+\theta_2+\theta_3)\\+137.981sin(\theta_1+\theta_2+\theta_3)\\-80.653sin(\theta_2+\theta_3+\theta_4)\\cos(\theta_1)\end{matrix} \\
0.893sin(\theta_1+\theta_2) & 0.893cos(\theta_1+\theta_2) & -0.448 & \begin{matrix}446.9sin(\theta_2)+\\446.9sin(\theta_2+\theta_3)\\160.919sin(\theta_2+\theta_3+\theta_4)\\+230\end{matrix} \\
0 & 0 & 0 & 1
\end{bmatrix}
\tag{8}
$$

We can make the trigonometric substitutions:

$$
\begin{aligned} r_{13} &= \rho sin\theta_1, \\ r_{23} &= -\rho cos\theta_1, \end{aligned} \tag{13}
$$

$$
\rho = \sqrt{r_{13}^2 - r_{23}^2} \tag{14}
$$

Therefore, as can be seen at Figure 7

$$
\theta_1 = Atan2(r_{13}, -r_{23}) \tag{15}
$$

If both $r_{13} = 0$ and $r_{23} = 0$ the goal is unattainable.
Equate r_{14} and r_{24} in relation to Equation 8 we have:
For r_{14}:

$$
P_x = r_{14} \tag{16}
$$

For r_{24}:

$$
P_y = r_{24} \tag{17}
$$

The systematic to find the equations for θ_n will be first isolating the dependent transpose on the left side by multiplying both sides of Equation 10 by the dependent transpose inverse ${}^0_1T(\theta_1)^{-1}$, as in Equation 21.
From this relation is possible to infer that:

VII. System Simulation

OpenModelica is currently the most complete open-source Modelica and FMI based modelling, simulation, optimization, and model-based development environment. Its long-term development is supported by a non-profit organization – the Open Source Modelica Consortium (OSMC). [2]

This system was chosen, mainly, because of the open-source aspect, since Mathworks Simulink requires a paid plugin to

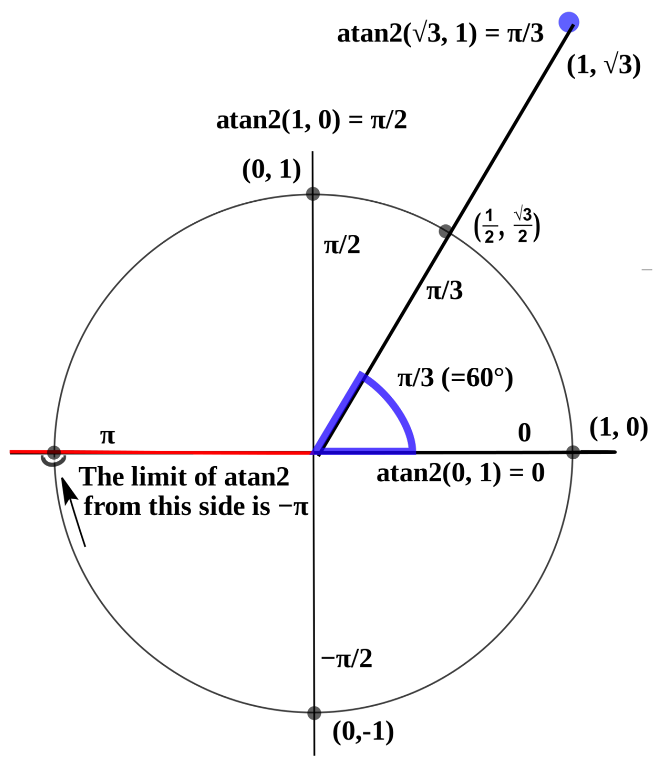

Fig. 7. Atan2.

$$
{}^{1}_{5}T = \begin{bmatrix} cos(\theta_1+\theta_2+\theta_3) & -sin(\theta_2+\theta_3+\theta_4) & 0 & \begin{matrix} l_2cos(\theta_2)+ \\ l_3cos(\theta_2+\theta_3)+ \\ l_4cos(\theta_2+\theta_3+\theta_4) \end{matrix} \\ 0 & 0 & -1 & \begin{matrix} (l_2sin(\theta_2)+ \\ l_3sin(\theta_2+\theta_3) \end{matrix} \\ sin(\theta_2+\theta_3+\theta_4) & cos(\theta_2+\theta_3+\theta_4) & 0 & \begin{matrix} (l_2sin(\theta_2)+ \\ l_3sin(\theta_2+\theta_3)+ \\ l_4sin(\theta_2+\theta_3+\theta_4) \end{matrix} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{20}
$$

$$
{}^{0}_{5}T^{0}_{1}T(\theta_1)^{-1} = ^{1}_{2} T(\theta_2)^{2}_{3}T(\theta_3)^{3}_{4}T(\theta_4)^{4}_{5}T \tag{18}
$$

Where,

$$
{}^{1}_{2}T(\theta_2)^{2}_{3}T(\theta_3)^{3}_{4}T(\theta_4)^{4}_{5}T = ^{1}_{5} T
$$

Therefore,

$$
\begin{bmatrix} cos\theta_1 & sin\theta_1 & 0 & 0 \\ -sin\theta_1 & cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & P_x \\ r_{21} & r_{22} & r_{23} & P_y \\ r_{31} & r_{32} & r_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = ^{1}_{5} T \tag{19}
$$

$$
{}^{0}_{5}T^{0}_{1}T(\theta_1)^{-1} = ^{1}_{2} T(\theta_2)^{2}_{3}T(\theta_3)^{3}_{4}T(\theta_4)^{4}_{5}T \tag{21}
$$

Where,

$$
{}^{1}_{2}T(\theta_2)^{2}_{3}T(\theta_3)^{3}_{4}T(\theta_4)^{4}_{5}T = ^{1}_{5} T
$$

Therefore,

$$
\begin{bmatrix} cos\theta_1 & sin\theta_1 & 0 & 0 \\ -sin\theta_1 & cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & P_x \\ r_{21} & r_{22} & r_{23} & P_y \\ r_{31} & r_{32} & r_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = ^{1}_{5} T \tag{22}
$$

connect the Solidworks model. Also due to the fact that Its a complete system for simulation modelling, versatile and capable of very complex tasks, much more complex than the current project.

For the modelling simulation parameters, information from the CAD simulator (SolidWorks) regarding to mass, center of mass and moments of inertia, were confronted with the Python simulation and was consistent as shown on Figure 8 and at the Python programming documentation that can be found on Appendix A. More information about the model is also provided but not included necessary to the link properties at OpenModelica, like density, volume, surface area, among others. The parameters necessary for the model were the length, mass and center of mass, as well as the inertia tensors, as shown on Figure 9.

To simulate the manipulator the component $Joints.Revolute$ was used for the joints, and $Parts.BodyShapes$ was used for the links, base and

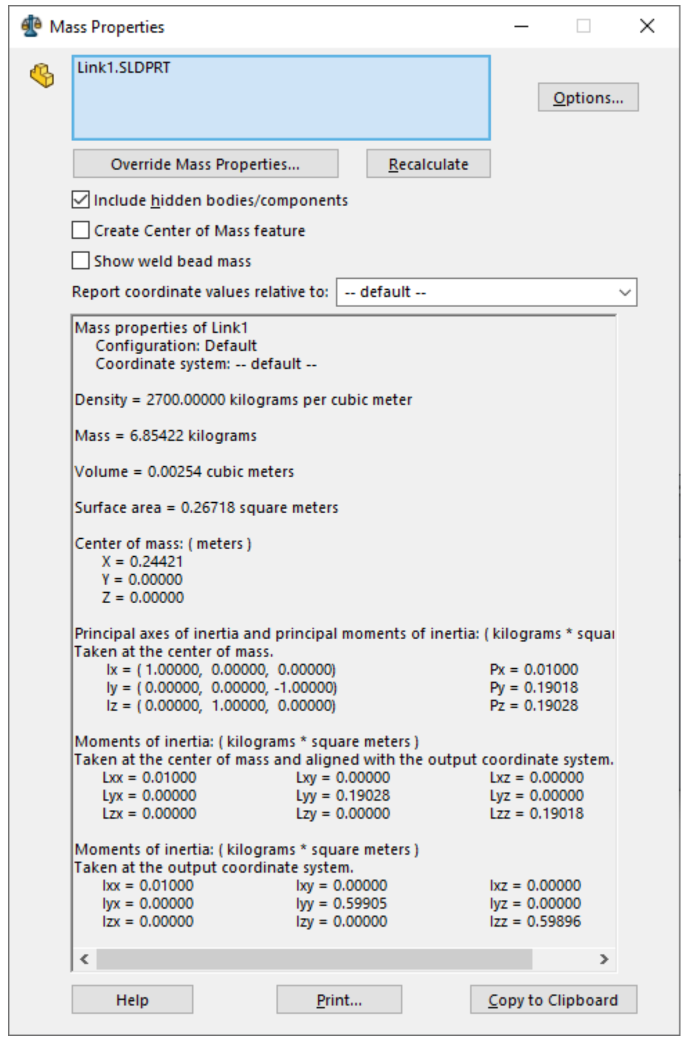

Fig. 8. Mass, center of mass and moments of inertia used on the simulation from link1 - SolidWorks.

tool, and also some auxiliary component blocks to simulate controllers, world conditions and a fixed base (ground). On the $Joints.Revolute$ component for the joints the option $useAxisFlange$, allow the control of the rotation and this option was used as shown on the $joint1$ parameters on Figure 10. A unit conversion block has to be used to convert from degrees to radians, there is a math block for that. With this control system the position can provide for $joint1$ by setting

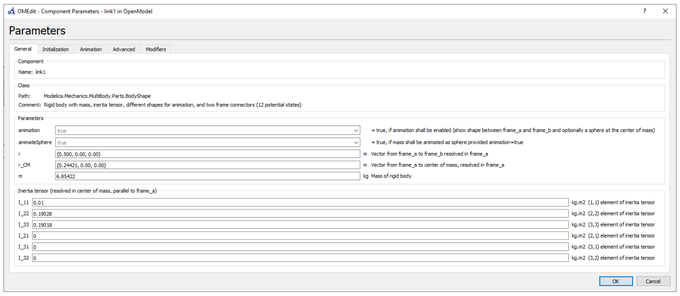

Fig. 9. OpenModelica, link1 parameters.

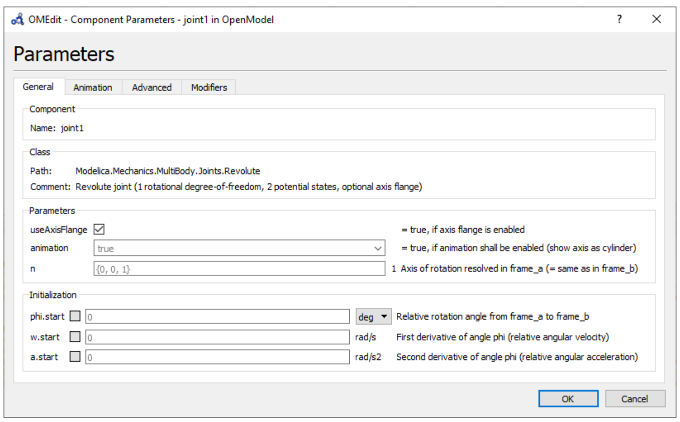

Fig. 10. OpenModelica, Joint1 parameters.

a value to the $Gain$ block. This will take the gain input and will provide it as a signal that the joint can use. This 6 blocks represent the first joint link of the system as represented in Figure 11.

The CAD modeling software chosen was SolidWorks from Dassault Systems, It is a solid modeling computer-aided design software with 2.3 million active users at over 234,800 companies in 80 countries. [5] SolidWorks goal is building 3D CAD software that was easy-to-use, affordable, and available on Windows. [6] The main reasons to use SolidWorks in this project were the easy of use, the calculations that can be used as OpenModelica parameters and the exporting features that allows easy integration between the two software.

An import aspect of exporting from Solidworks to Open-

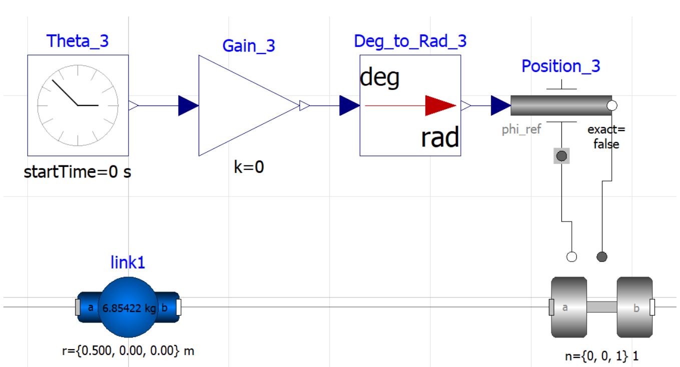

Fig. 11. Joint1, Link1 and controller.

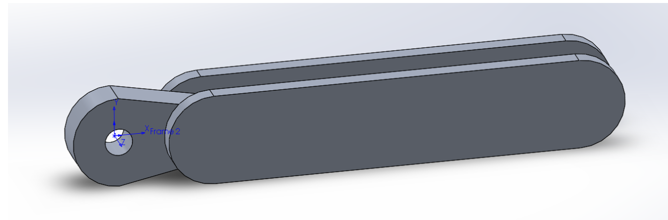

Fig. 12. Coordinate system on the frame $\{4\}$ position for link 1.

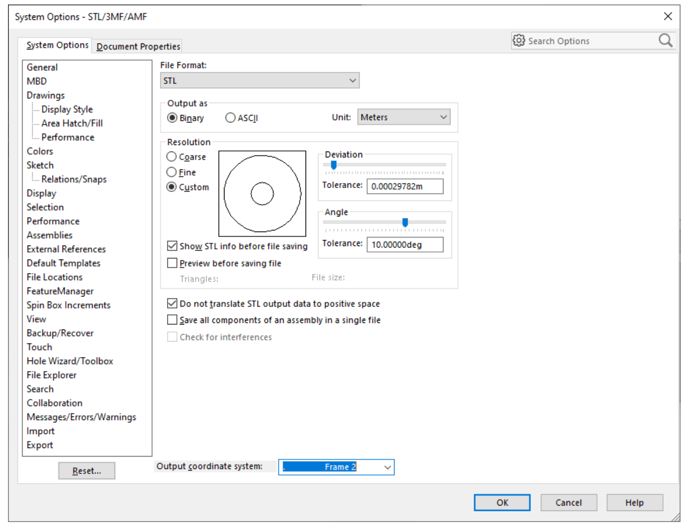

Fig. 13. Exporting parameters.

Modelica is the compatibility, the exported file can be a $.STL$ file. Although the measurements have to be in meters. In the STL exporting window there is the option "Do not translate STL output data to positive space", this option makes exported parts maintain their original position in global space, relative to the origin [4].

There are many ways to export from one software to the other, on this project approach, at the export STL window, the coordinate system is been output, as can be seen in the import parameters for link 1 on Figure 13, by exporting the coordinate system placed on the frame position, Figure 12, the vector from frame $\{A\}$ to the shape origin, resolved in $\{A\}$ is equal to 0, because the frame coordinate system exported is located at the origin.

We can use the center of mass directly from SoildWorks as well as use the exact distance between the frames at the r parameter as shown in the Figure 9 and Figure 14.

The calculations for the moments of inertia were made in Python, refer to Appendix A, and confirmed in the mass evaluation at SolidWorks, also, in OpenModelica, as can be seen see on Figure 15 the values for the torque for joint 1 and 2 using $\theta_{1-4} = 0°$.

At Figure 16 the simulation modelling at OpenModelica is shown, the parameters used at the $Gain$ block results in an animation, Figure 17, that goes from all the angles been 0 to the values set.

Another simulation was made in SolidWorks to show that the proposed design is capable of picking an object from a

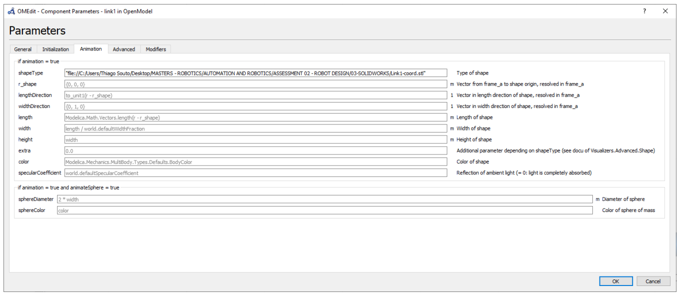

Fig. 14. Importing parameters.

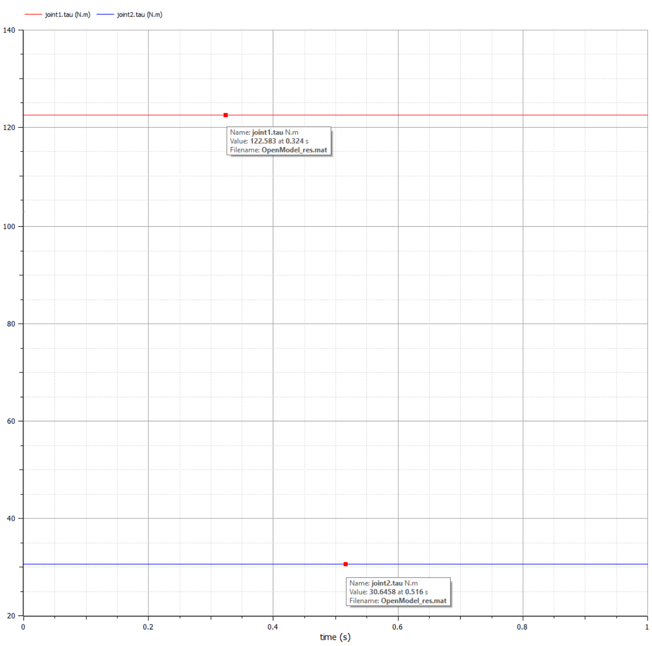

Fig. 15. Torques required for $\theta_{1-4} = 0°$.

vertical wall/shelf and placing it onto a horizontal surface.

VIII. Discussion and Conclusions

The IEEE Graphics Checker Tool enables users to check graphic files. The tool will check journal article graphic files against a set of rules for compliance with IEEE requirements. These requirements are designed to ensure sufficient image quality so they will look acceptable in print. After receiving a graphic or a set of graphics, the tool will check the files against a set of rules. A report will then be e-mailed listing each graphic and whether it met or failed to meet the requirements. If the file fails, a description of why and instructions on how to correct the problem will be sent. The IEEE Graphics Checker Tool is available at http://graphicsqc.ieee.org/

For more Information, contact the IEEE Graphics H-E-L-P Desk by e-mail at mailto:graphics@ieee.org. You will then receive an e-mail response and sometimes a request for a sample graphic for us to check.

A. Copyright Form

An IEEE copyright form should accompany your final submission. You can get a .pdf, .html, or .doc version at http://www.ieee.org/copyright. Authors are responsible for obtaining any security clearances.

TABLE I
Units for Magnetic Properties

Symbol	Quantity	Conversion from Gaussian and CGS EMU to SI [a]
Φ	magnetic flux	1 Mx $\rightarrow 10^{-8}$ Wb $= 10^{-8}$ V·s
B	magnetic flux density, magnetic induction	1 G $\rightarrow 10^{-4}$ T $= 10^{-4}$ Wb/m^2
H	magnetic field strength	1 Oe $\rightarrow 10^3/(4\pi)$ A/m
m	magnetic moment	1 erg/G = 1 emu $\rightarrow 10^{-3}$ A·m^2 $= 10^{-3}$ J/T
M	magnetization	1 erg/(G·cm^3) = 1 emu/cm^3 $\rightarrow 10^3$ A/m
$4\pi M$	magnetization	1 G $\rightarrow 10^3/(4\pi)$ A/m
σ	specific magnetization	1 erg/(G·g) = 1 emu/g $\rightarrow$ 1 A·m^2/kg
j	magnetic dipole moment	1 erg/G = 1 emu $\rightarrow 4\pi \times 10^{-10}$ Wb·m
J	magnetic polarization	1 erg/(G·cm^3) = 1 emu/cm^3 $\rightarrow 4\pi \times 10^{-4}$ T
χ, κ	susceptibility	$1 \rightarrow 4\pi$
χ_ρ	mass susceptibility	1 cm^3/g $\rightarrow 4\pi \times 10^{-3}$ m^3/kg
μ	permeability	$1 \rightarrow 4\pi \times 10^{-7}$ H/m $= 4\pi \times 10^{-7}$ Wb/(A·m)
μ_r	relative permeability	$\mu \rightarrow \mu_r$
w, W	energy density	1 erg/cm^3 $\rightarrow 10^{-1}$ J/m^3
N, D	demagnetizing factor	$1 \rightarrow 1/(4\pi)$

Vertical lines are optional in tables. Statements that serve as captions for the entire table do not need footnote letters.
[a]Gaussian units are the same as cg emu for magnetostatics; Mx = maxwell, G = gauss, Oe = oersted; Wb = weber, V = volt, s = second, T = tesla, m = meter, A = ampere, J = joule, kg = kilogram, H = henry.

IX. Units

Use either SI (MKS) or CGS as primary units. (SI units are strongly encouraged.) English units may be used as secondary units (in parentheses). **This applies to papers in data storage.** For example, write "15 Gb/cm^2 (100 Gb/in^2)." An exception is when English units are used as identifiers in trade, such as "$3\frac{1}{2}$-in disk drive." Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity in an equation.

The SI unit for magnetic field strength H is A/m. However, if you wish to use units of T, either refer to magnetic flux density B or magnetic field strength symbolized as $\mu_0 H$. Use the center dot to separate compound units, e.g., "A·m^2."

X. Helpful Hints

A. Figures and Tables

Because IEEE will do the final formatting of your paper, you do not need to position figures and tables at the top and bottom of each column. Large figures and tables may span both columns. Place figure captions below the figures; place table titles above the tables. If your figure has two parts, include the labels "(a)" and "(b)" as part of the artwork. Please verify that the figures and tables you mention in the text actually exist. **Please do not include captions as part of the figures. Do not put captions in "text boxes" linked to the figures. Do not put borders around the outside of your figures.** Use

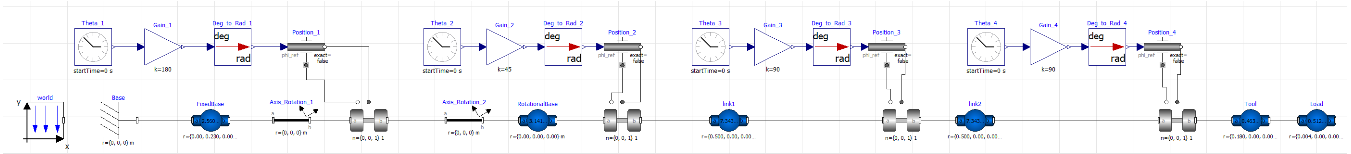

Fig. 16. Simulation model, Base, rotational link and 2 more links.

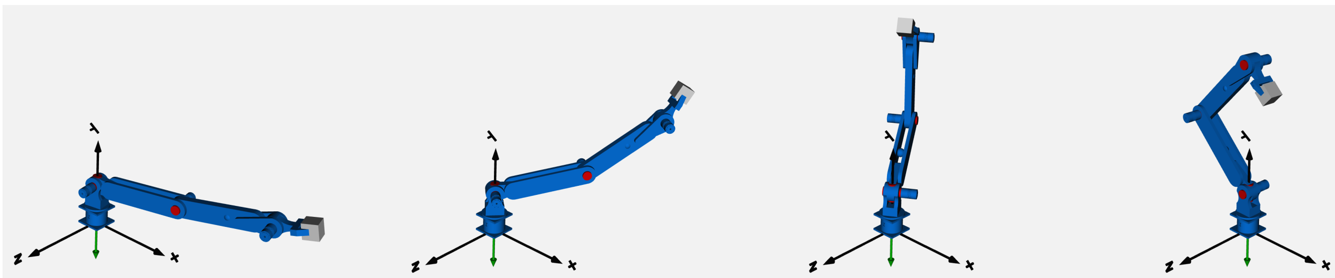

Fig. 17. Simulation animation.

the abbreviation “Fig.” even at the beginning of a sentence. Do not abbreviate “Table.” Tables are numbered with Roman numerals.

Figure axis labels are often a source of confusion. Use words rather than symbols. As an example, write the quantity “Magnetization,” or “Magnetization M,” not just “M.” Put units in parentheses. Do not label axes only with units. As in Fig. 1, for example, write “Magnetization (A/m)” or “Magnetization (A$\cdot$m^{-1}),” not just “A/m.” Do not label axes with a ratio of quantities and units. For example, write “Temperature (K),” not “Temperature/K.”

Multipliers can be especially confusing. Write “Magnetization (kA/m)” or “Magnetization (10^3 A/m).” Do not write “Magnetization (A/m) $\times$ 1000” because the reader would not know whether the top axis label in Fig. 1 meant 16000 A/m or 0.016 A/m. Figure labels should be legible, approximately 8 to 12 point type.

B. LaTeX-Specific Advice

Please use “soft” (e.g., `\eqref{Eq}`) cross references instead of “hard” references (e.g., `(1)`). That will make it possible to combine sections, add equations, or change the order of figures or citations without having to go through the file line by line.

Please don’t use the `{eqnarray}` equation environment. Use `{align}` or `{IEEEeqnarray}` instead. The `{eqnarray}` environment leaves unsightly spaces around relation symbols.

Please note that the `{subequations}` environment in LaTeX will increment the main equation counter even when there are no equation numbers displayed. If you forget that, you might write an article in which the equation numbers skip from (17) to (20), causing the copy editors to wonder if you’ve discovered a new method of counting.

BibTeX does not work by magic. It doesn’t get the bibliographic data from thin air but from .bib files. If you use BibTeX to produce a bibliography you must send the .bib files.

LaTeX can’t read your mind. If you assign the same label to a subsubsection and a table, you might find that Table I has been cross referenced as Table IV-B3.

LaTeX does not have precognitive abilities. If you put a `\label` command before the command that updates the counter it’s supposed to be using, the label will pick up the last counter to be cross referenced instead. In particular, a `\label` command should not go before the caption of a figure or a table.

Do not use `\nonumber` or `\notag` inside the `{array}` environment. It will not stop equation numbers inside `{array}` (there won’t be any anyway) and it might stop a wanted equation number in the surrounding equation.

C. References

Number citations consecutively in square brackets [1]. The sentence punctuation follows the brackets [2]. Multiple references [2], [3] are each numbered with separate brackets [1]–[3]. When citing a section in a book, please give the relevant page numbers [2]. In sentences, refer simply to the reference number, as in [3]. Do not use “Ref. [3]” or “reference [3]” except at the beginning of a sentence: “Reference [3] shows . . . .” Please do not use automatic endnotes in *Word*, rather, type the reference list at the end of the paper using the “References” style.

Number footnotes separately in superscripts (Insert | Footnote).[1] Place the actual footnote at the bottom of the column in which it is cited; do not put footnotes in the reference list (endnotes). Use letters for table footnotes (see Table I).

Please note that the references at the end of this document are in the preferred referencing style. Give all authors’ names; do not use “*et al.*” unless there are six authors or more. Use a space after authors’ initials. Papers that have not been published should be cited as “unpublished” [4]. Papers that

[1]It is recommended that footnotes be avoided (except for the unnumbered footnote with the receipt date on the first page). Instead, try to integrate the footnote information into the text.

have been accepted for publication, but not yet specified for an issue should be cited as "to be published" [5]. Papers that have been submitted for publication should be cited as "submitted for publication" [6]. Please give affiliations and addresses for private communications [7].

Capitalize only the first word in a paper title, except for proper nouns and element symbols. For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [11].

D. Abbreviations and Acronyms

Define abbreviations and acronyms the first time they are used in the text, even after they have already been defined in the abstract. Abbreviations such as IEEE, SI, ac, and dc do not have to be defined. Abbreviations that incorporate periods should not have spaces: write "C.N.R.S.," not "C. N. R. S." Do not use abbreviations in the title unless they are unavoidable (for example, "IEEE" in the title of this article).

E. Equations

Number equations consecutively with equation numbers in parentheses flush with the right margin, as in (23). First use the equation editor to create the equation. Then select the "Equation" markup style. Press the tab key and write the equation number in parentheses. To make your equations more compact, you may use the solidus (/), the exp function, or appropriate exponents. Use parentheses to avoid ambiguities in denominators. Punctuate equations when they are part of a sentence, as in

$$\int_0^{r_2} F(r,\phi)\, dr\, d\phi = [\sigma r_2/(2\mu_0)] \cdot \int_0^{\infty} \exp(-\lambda|z_j - z_i|)\lambda^{-1} J_1(\lambda r_2) J_0(\lambda r_i)\, d\lambda. \quad (23)$$

Be sure that the symbols in your equation have been defined before the equation appears or immediately following. Italicize symbols (T might refer to temperature, but T is the unit tesla). Refer to "(23)," not "Eq. (23)" or "equation (23)," except at the beginning of a sentence: "Equation (23) is . . . ."

F. Other Recommendations

Use one space after periods and colons. Hyphenate complex modifiers: "zero-field-cooled magnetization." Avoid dangling participles, such as, "Using (23), the potential was calculated." [It is not clear who or what used (23).] Write instead, "The potential was calculated by using (23)," or "Using (23), we calculated the potential."

Use a zero before decimal points: "0.25," not ".25." Use "cm^3," not "cc." Indicate sample dimensions as "0.1 cm $\times$ 0.2 cm," not "0.1 $\times$ 0.2 cm^2." The abbreviation for "seconds" is "s," not "sec." Do not mix complete spellings and abbreviations of units: use "Wb/m^2" or "webers per square meter," not "webers/m^2." When expressing a range of values, write "7 to 9" or "7-9," not "7$\sim$9."

A parenthetical statement at the end of a sentence is punctuated outside of the closing parenthesis (like this). (A parenthetical sentence is punctuated within the parentheses.) In American English, periods and commas are within quotation marks, like "this period." Other punctuation is "outside"! Avoid contractions; for example, write "do not" instead of "don't." The serial comma is preferred: "A, B, and C" instead of "A, B and C."

If you wish, you may write in the first person singular or plural and use the active voice ("I observed that . . ." or "We observed that . . ." instead of "It was observed that . . ."). Remember to check spelling. If your native language is not English, please get a native English-speaking colleague to carefully proofread your paper.

XI. Some Common Mistakes

The word "data" is plural, not singular. The subscript for the permeability of vacuum μ_0 is zero, not a lowercase letter "o." The term for residual magnetization is "remanence"; the adjective is "remanent"; do not write "remnance" or "remnant." Use the word "micrometer" instead of "micron." A graph within a graph is an "inset," not an "insert." The word "alternatively" is preferred to the word "alternately" (unless you really mean something that alternates). Use the word "whereas" instead of "while" (unless you are referring to simultaneous events). Do not use the word "essentially" to mean "approximately" or "effectively." Do not use the word "issue" as a euphemism for "problem." When compositions are not specified, separate chemical symbols by en-dashes; for example, "NiMn" indicates the intermetallic compound $\mathrm{Ni_{0.5}Mn_{0.5}}$ whereas "Ni–Mn" indicates an alloy of some composition $\mathrm{Ni}_x\mathrm{Mn}_{1-x}$.

Be aware of the different meanings of the homophones "affect" (usually a verb) and "effect" (usually a noun), "complement" and "compliment," "discreet" and "discrete," "principal" (e.g., "principal investigator") and "principle" (e.g., "principle of measurement"). Do not confuse "imply" and "infer."

Prefixes such as "non," "sub," "micro," "multi," and "ultra" are not independent words; they should be joined to the words they modify, usually without a hyphen. There is no period after the "et" in the Latin abbreviation "*et al.*" (it is also italicized). The abbreviation "i.e.," means "that is," and the abbreviation "e.g.," means "for example" (these abbreviations are not italicized).

An excellent style manual and source of information for science writers is [12]. A general IEEE style guide and an *Information for Authors* are both available at http://www.ieee.org/web/publications/authors/transjnl/index.html

XII. Editorial Policy

Submission of a manuscript is not required for participation in a conference. Do not submit a reworked version of a paper you have submitted or published elsewhere. Do not publish "preliminary" data or results. The submitting author is responsible for obtaining agreement of all coauthors and any consent required from sponsors before submitting a paper. IEEE TRANSACTIONS and JOURNALS strongly discourage courtesy authorship. It is the obligation of the authors to cite relevant prior work.

The Transactions and Journals Department does not publish conference records or proceedings. The TRANSACTIONS does publish papers related to conferences that have been recommended for publication on the basis of peer review. As a matter of convenience and service to the technical community, these topical papers are collected and published in one issue of the TRANSACTIONS.

At least two reviews are required for every paper submitted. For conference-related papers, the decision to accept or reject a paper is made by the conference editors and publications committee; the recommendations of the referees are advisory only. Undecipherable English is a valid reason for rejection. Authors of rejected papers may revise and resubmit them to the TRANSACTIONS as regular papers, whereupon they will be reviewed by two new referees.

XIII. Publication Principles

The contents of IEEE TRANSACTIONS and JOURNALS are peer-reviewed and archival. The TRANSACTIONS publishes scholarly articles of archival value as well as tutorial expositions and critical reviews of classical subjects and topics of current interest.

Authors should consider the following points:

1) Technical papers submitted for publication must advance the state of knowledge and must cite relevant prior work.
2) The length of a submitted paper should be commensurate with the importance, or appropriate to the complexity, of the work. For example, an obvious extension of previously published work might not be appropriate for publication or might be adequately treated in just a few pages.
3) Authors must convince both peer reviewers and the editors of the scientific and technical merit of a paper; the standards of proof are higher when extraordinary or unexpected results are reported.
4) Because replication is required for scientific progress, papers submitted for publication must provide sufficient information to allow readers to perform similar experiments or calculations and use the reported results. Although not everything need be disclosed, a paper must contain new, useable, and fully described information. For example, a specimen's chemical composition need not be reported if the main purpose of a paper is to introduce a new measurement technique. Authors should expect to be challenged by reviewers if the results are not supported by adequate data and critical details.
5) Papers that describe ongoing work or announce the latest technical achievement, which are suitable for presentation at a professional conference, may not be appropriate for publication in a TRANSACTIONS or JOURNAL.

XIV. Conclusion

Please include a brief summary of the possible clinical implications of your work in the conclusion section. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. Consider elaborating on the translational importance of the work or suggest applications and extensions.

REFERENCES

[1] J. J. Craig, *Introduction To Robotics: Mechanics And Control*, 3rd ed., Ed. New York: Pearson Education, 2009.

[2] P. Fritzson, et al. *The OpenModelica Integrated Modeling, Simulation and Optimization Environment* (Conference paper), PROCEEDINGS OF THE 1ST AMERICAN MODELICA CONFERENCE.Cambridge, USA: Massechusetts, 2018.

[3] W. Bolton, *Mechatronics - Electronic control systems in mechanical and electrical engineering*. United Kingdom: Pearson Education Limited, 2019.

[4] (SOLIDWORKS Online Help) Dassault Systems. (2020, May). Available: https://help.solidworks.com/.

[5] (The SOLIDWORKS blog) Dassault Systems. (2020, May). Available: https://blogs.solidworks.com/solidworksblog/2016/10/growing-solidworks-nation.html.

[6] (SolidWoks - Wikipedia)(2020, May). Available: https://en.wikipedia.org/wiki/SolidWorks.

[7] K.N. Frazer, School of Food and Advanced Technology, Massey University, Slides from Lecture 12, March 2019.

[8] *RDrive Datasheet* ROZUM Robotics LLC., Mountain House, CA - USA, 2020.

[9] (Atan2 - Wikipedia)(2020, May). Available: https://en.wikipedia.org/wiki/Atan2

[10] J. Wang, "Fundamentals of erbium-doped fiber amplifiers arrays (Periodical style—Submitted for publication)," *IEEE J. Quantum Electron.*, submitted for publication.

[11] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interfaces (Translation Journals style)," *IEEE Transl. J. Magn.Jpn.*, vol. 2, Aug. 1987, pp. 740–741 [*Dig. 9^{th} Annu. Conf. Magnetics* Japan, 1982, p. 301].

[12] M. Young, *The Technical Writers Handbook.* Mill Valley, CA: University Science, 1989.

[13] J. U. Duncombe, "Infrared navigation—Part I: An assessment of feasibility (Periodical style)," *IEEE Trans. Electron Devices*, vol. ED-11, pp. 34–39, Jan. 1959.

[14] S. Chen, B. Mulgrew, and P. M. Grant, "A clustering technique for digital communications channel equalization using radial basis function networks," *IEEE Trans. Neural Networks*, vol. 4, pp. 570–578, Jul. 1993.

[15] R. W. Lucky, "Automatic equalization for digital communication," *Bell Syst. Tech. J.*, vol. 44, no. 4, pp. 547–588, Apr. 1965.

[16] S. P. Bingulac, "On the compatibility of adaptive controllers (Published Conference Proceedings style)," in *Proc. 4th Annu. Allerton Conf. Circuits and Systems Theory*, New York, 1994, pp. 8–16.

[17] G. R. Faulhaber, "Design of service systems with priority reservation," in *Conf. Rec. 1995 IEEE Int. Conf. Communications,* pp. 3–8.

[18] W. D. Doyle, "Magnetization reversal in films with biaxial anisotropy," in *1987 Proc. INTERMAG Conf.*, pp. 2.2-1–2.2-6.

[19] G. W. Juette and L. E. Zeffanella, "Radio noise currents n short sections on bundle conductors (Presented Conference Paper style)," presented at the IEEE Summer power Meeting, Dallas, TX, Jun. 22–27, 1990, Paper 90 SM 690-0 PWRS.

[20] J. G. Kreifeldt, "An analysis of surface-detected EMG as an amplitude-modulated noise," presented at the 1989 Int. Conf. Medicine and Biological Engineering, Chicago, IL.

[21] J. Williams, "Narrow-band analyzer (Thesis or Dissertation style)," Ph.D. dissertation, Dept. Elect. Eng., Harvard Univ., Cambridge, MA, 1993.

[22] N. Kawasaki, "Parametric study of thermal and chemical nonequilibrium nozzle flow," M.S. thesis, Dept. Electron. Eng., Osaka Univ., Osaka, Japan, 1993.

[23] J. P. Wilkinson, "Nonlinear resonant circuit devices (Patent style)," U.S. Patent 3 624 12, July 16, 1990.

[24] *IEEE Criteria for Class IE Electric Systems* (Standards style), IEEE Standard 308, 1969.

[25] *Letter Symbols for Quantities*, ANSI Standard Y10.5-1968.

[26] R. E. Haskell and C. T. Case, "Transient signal propagation in lossless isotropic plasmas (Report style)," USAF Cambridge Res. Lab., Cambridge, MA Rep. ARCRL-66-234 (II), 1994, vol. 2.

[27] E. E. Reber, R. L. Michell, and C. J. Carter, "Oxygen absorption in the Earth's atmosphere," Aerospace Corp., Los Angeles, CA, Tech. Rep. TR-0200 (420-46)-3, Nov. 1988.

[28] (Handbook style) *Transmission Systems for Communications,* 3rd ed., Western Electric Co., Winston-Salem, NC, 1985, pp. 44–60.

[29] *Motorola Semiconductor Data Manual,* Motorola Semiconductor Products Inc., Phoenix, AZ, 1989.

[30] (Basic Book/Monograph Online Sources) J. K. Author. (year, month, day). *Title* (edition) [Type of medium]. Volume (issue). Available: http://www.(URL)

[31] J. Jones. (1991, May 10). Networks (2nd ed.) [Online]. Available: http://www.atm.com

[32] R. J. Vidmar. (1992, August). On the use of atmospheric plasmas as electromagnetic reflectors. *IEEE Trans. Plasma Sci.* [Online]. *21(3).* pp. 876–880. Available: http://www.halcyon.com/pub/journals/21ps03-vidmar

Matrices Manipulation Documentation

Release 1.0

Thiago Souto

May 09, 2020

CONTENTS:

CHAPTER

ONE

FORWARDKINEMATICS MODULE

class Forward_Kinematics.**ForwardKinematics**(***kwargs*)

Bases: object

Definition: This class generates Homogeneous transform matrices, although it uses a symbolic approach that can be used to multiply any matrix and obtain the translation or rotation.

sympy.cos and sympy.sin: cos and sin for sympy

sympy.simplify: SymPy has dozens of functions to perform various kinds of simplification. simplify() attempts to apply all of these functions in an intelligent way to arrive at the simplest form of an expression.

Returns: It returns Rotation and translation matrices.

Obs: **kwargs (keyword arguments) are used to facilitate the identification of the parameters, so initiate the object

rot_x(*alpha*)

Definition: Receives an alpha angle and returns the rotation matrix for the given angle at the *X* axis.

Parameters **alpha** (*string*) – Rotation Angle around the X axis

Returns: The Rotational Matrix at the X axis by an *given* angle

rot_z(*theta*)

Definition: Receives an theta angle and returns the rotation matrix for the given angle at the *Z* axis.

Parameters **theta** (*string*) – Rotation Angle around the Z axis

Returns: The Rotational Matrix at the Z axis by an *given* angle

trans_x(*a*)

Definition: Translates the matrix a given amount *a* on the *X* axis by Defining a 4x4 identity matrix with *a* as the (1,4) element.

Parameters **a** (*string*) – Distance translated on the X-axis

Returns: The Translation Matrix on the *X* axis by a given distance

trans_z(*d*)

Definition: Translate the matrix a given amount *d* on the *Z* axis. by Defining a matrix T 4x4 identity matrix with *d* (3,4) element position.

Parameters **d** (*string*) – Distance translated on the Z-axis

Returns: The Translation Matrix on the *Z* axis by a given distance

Forward_Kinematics.**main**()

Assessment 02 Robotic manipulator design - Forward Kinematics.

```
import numpy as np
import sympy as sympy


class ForwardKinematics:

    """
    Definition: This class generates Homogeneous transform matrices, although it uses␣
↪a symbolic approach
    that can be used to multiply any matrix and obtain the translation or rotation.

    sympy.cos and sympy.sin: cos and sin for sympy

    sympy.simplify: SymPy has dozens of functions to perform various kinds of␣
↪simplification.
    simplify() attempts to apply all of these functions
    in an intelligent way to arrive at the simplest form of an expression.

    Returns: It returns Rotation and translation matrices.

    Obs: **kwargs (keyword arguments) are used to facilitate the identification of␣
↪the parameters, so initiate the
    object
    """
    np.set_printoptions(precision=3, suppress=True)

    sympy.init_printing(use_unicode=True, num_columns=400)

    def __init__(self, **kwargs):
        """
        Initializes the Object.
        """
        self._x_angle = kwargs['x_angle'] if 'x_angle' in kwargs else 'alpha_i-1'
        self._x_dist = kwargs['x_dist'] if 'x_dist' in kwargs else 'a_i-1'
        self._y_angle = kwargs['y_angle'] if 'y_angle' in kwargs else '0'
        self._y_dist = kwargs['y_dist'] if 'y_dist' in kwargs else '0'
        self._z_angle = kwargs['z_angle'] if 'z_angle' in kwargs else 'theta_i'
        self._z_dist = kwargs['z_dist'] if 'z_dist' in kwargs else 'd_i'

    def trans_x(self, a):
        """
        Definition: Translates the matrix a given amount `a` on the *X* axis by␣
↪Defining a 4x4 identity
        matrix with `a` as the (1,4) element.

        :type a: string
        :param a: Distance translated on the X-axis

        Returns: The Translation Matrix on the *X* axis by a given distance
        """
        self._x_dist = a

        t_x = sympy.Matrix([[1, 0, 0, self._x_dist],
                            [0, 1, 0, 0],
                            [0, 0, 1, 0],
                            [0, 0, 0, 1]])
```

(continues on next page)

(continued from previous page)

```
        t_x = t_x.evalf()

        return t_x

    def trans_z(self, d):
        """
        Definition: Translate the matrix a given amount `d` on the *Z* axis. by␣
↪Defining a matrix T 4x4 identity
        matrix with *d* (3,4) element position.

        :type d: string
        :param d: Distance translated on the Z-axis

        Returns: The Translation Matrix on the *Z* axis by a given distance
        """
        self._z_dist = d

        t_z = sympy.Matrix([[1, 0, 0, 0],
                            [0, 1, 0, 0],
                            [0, 0, 1, self._z_dist],
                            [0, 0, 0, 1]])

        t_z = t_z.evalf()

        return t_z

    def rot_x(self, alpha):
        """
        Definition: Receives an alpha angle and returns the rotation matrix for the␣
↪given angle at the *X* axis.

        :type alpha: string
        :param alpha: Rotation Angle around the X axis

        Returns: The Rotational Matrix at the X axis by an *given* angle
        """
        self._x_angle = alpha

        r_x = sympy.Matrix([[1, 0, 0, 0],
                              [0, sympy.cos(self._x_angle), -sympy.sin(self._x_angle),
↪ 0],
                              [0, sympy.sin(self._x_angle), sympy.cos(self._x_angle),␣
↪0],
                              [0, 0, 0, 1]])

        r_x = r_x.evalf()

        return r_x

    def rot_z(self, theta):
        """
        Definition: Receives an theta angle and returns the rotation matrix for the␣
↪given angle at the *Z* axis.

        :type theta: string
        :param theta: Rotation Angle around the Z axis
```

(continues on next page)

(continued from previous page)

```
        Returns: The Rotational Matrix at the Z axis by an *given* angle
        """
        self._z_angle = theta

        r_z = sympy.Matrix([[sympy.cos(self._z_angle), -sympy.sin(self._z_angle), 0,␣
↪0],
                            [sympy.sin(self._z_angle), sympy.cos(self._z_angle), 0,␣
↪0],
                            [0, 0, 1, 0],
                            [0, 0, 0, 1]])

        r_z = r_z.evalf()

        return r_z


def main():
    """
    Assessment 02 Robotic manipulator design - Forward Kinematics.
    """
    a1 = ForwardKinematics()        # Rx(a_i-1)
    a2 = ForwardKinematics()        # Dx(a_i-1)
    a3 = ForwardKinematics()        # Dz(d_i)
    a4 = ForwardKinematics()        # Rz(theta_i)

    print()
    print('Matrix t_0_1:')
    t_0_1 = (a1.rot_x('0')) * (a2.trans_x('0')) * (a3.trans_z('l1')) * (a4.rot_z(
↪'theta_1'))
    print(sympy.pretty(t_0_1))

    print()
    print('Matrix t_1_2:')
    t_1_2 = (a1.rot_x('90.00')) * (a2.trans_x('0')) * (a3.trans_z('0')) * (a4.rot_z(
↪'theta_2'))
    print(sympy.pretty(t_1_2))

    print()
    print('Matrix t_2_3:')
    t_2_3 = (a1.rot_x('0')) * (a2.trans_x('l2')) * (a3.trans_z('0')) * (a4.rot_z(
↪'theta_3'))
    print(sympy.pretty(t_2_3))

    print()
    print('Matrix t_3_4:')
    t_3_4 = (a1.rot_x('0')) * (a2.trans_x('l3')) * (a3.trans_z('0')) * (a4.rot_z(
↪'theta_4'))
    print(sympy.pretty(t_3_4))

    print()
    print('Matrix t_4_5:')
    t_4_5 = (a1.rot_x('0')) * (a2.trans_x('l4')) * (a3.trans_z('0')) * (a4.rot_z('0'))
    print(sympy.pretty(t_4_5))

    t_0_5 = sympy.simplify(t_0_1 * t_1_2 * t_2_3 * t_3_4 * t_4_5)
    print('Matrix T_0_5:')
    print(sympy.pretty(t_0_5))
```

(continues on next page)

(continued from previous page)

```
    t_0_5_subs = t_0_5.subs([('l1', 230), ('l2', 500), ('l3', 500), ('l4', 180)])

    print('Matrix T_0_5: with substitutions Round')
    print(sympy.pretty(t_0_5_subs))


if __name__ == '__main__':
    main()
```

CHAPTER

TWO

INDICES AND TABLES

At the website you can navigate through the menus below:

- genindex
- modindex
- search

2.1 Running the documentation with Sphinx

To run the documentation for this project run the following commands, at the project folder:

Install Spinxs:

python -m pip install sphinx

Install the "Read the Docs" theme:

pip install sphinx-rtd-theme

make clean

make html

2.2 GitHub Repository

Find all the files at the GitHub repository here.

PYTHON MODULE INDEX

f