SIMULATION OF A ROBOTIC ARM FORWARD KINEMATICS IN CAD ENVIRONMENT

Konstantin Metodiev

Space Research and Technology Institute, Bulgarian Academy of Sciences
Acad. Georgy Bonchev Str., Bl. 1, 1113 Sofia,
e-mail: komet@space.bas.bg

Abstract. In the current case study, a CAD model of 5-DOF jointed robotic arm is designed and presented. The model has been developed in Autodesk® Inventor environment. In addition to part and assembly files, it includes an interface form and an iLogic rule for the user to control any joint deflection. End-effector transformation matrix is computed afterwards and displayed by a 3rd party tool DebugView after pressing a button. An alternative approach is to compute transformation matrix by means of Denavit-Hartenberg parameters. A thorough discussion as to how the proposed approach is beneficial to the project is given in corresponding section.

Keywords: Articulated Robot, Autodesk Inventor, Robotic Arm, iLogic

1. INTRODUCTION

Forward kinematics for a robotic arm comprises calculating position and orientation of the end effector (the tool or gripper at the arm end) based on given joint angles and link lengths. The process is essential for determining the end effector attitude. It is necessary to compute transformation matrices for each joint and multiply them afterwards. One way to do so is to calculate each matrix by means of Denavit-Hartenberg parameters (consisting of two linear and two angular quantities) describing connections between two consecutive joints. An alternative way is to make the most out of Inventor API and calculate the matrix by means of an iLogic rule.

Inventor is a powerful 3D CAD (Computer Aided Design) software developed by Autodesk®. It is widely used for mechanical design, simulation, visualization, and documentation. 3D models of mechanical components and assemblies (electronic mock-ups) might be created by means of parametric, direct, freeform, and rules-based design. With iLogic, repetitive tasks are automated and design is customized quickly.

DOI: 10.7546/EngSci.LIX.24.03.04

iLogic is a powerful toolset within Autodesk Inventor that enables the developer to automate specific tasks, processes, or functions in design workflow by means of rules. The rules are tiny scripts coded in Visual Basic for Applications and stored in the assembly file. The Inventor API methods and properties are accessible from within the coding environment. iLogic rules reduce manual effort and increase design accuracy, [1].

The project goal is to demonstrate opportunities that Autodesk Inventor affords of designing and testing a robotic arm. The robot forward kinematics parameters are considered in particular. The end-effector transformation matrix is calculated using either the Inventor API or Denavit-Hartenberg parameters.

2. MATERIALS AND METHODS

A 5-DOF robotic arm used as a test plant is shown in Fig. 1. The robotic arm parts and assembly have been developed in Autodesk Inventor IDE. A pair of User – defined Coordinate Systems (UCS) is attached to each joint so that an angle constraint might be applied to the axes. UCS is available in both assemblies and parts. In current study case, latter option has been chosen: UCSs are attached to each joint within the part file. The UCSs used in current project are visible in Fig. 1.

Fig. 1. A robotic arm with UCSs attached to each joint (render: Autodesk Inventor)

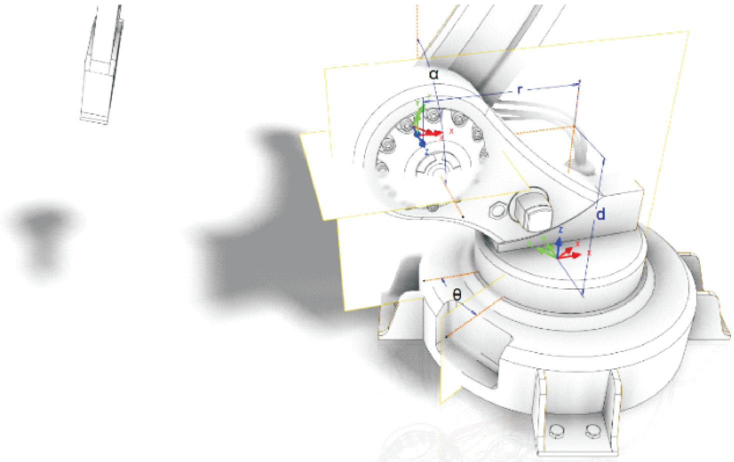

Fig. 2. Denavit-Hartenberg homogeneous parameters, [2]

In Fig. 2, Denavit-Hartenberg parameters are shown between first and second rotating joints. These parameters are following:

- d is a distance between X axes (along Z axis)
- r is a distance between Z axes (along X axis, i.e., length of the common normal itself)
- θrotates along the previous Z axis to align with the second X axis
- αrotates about the common normal angle between Z axes

For any joint, Z axis always coincides with the rotation/translation vector. Except for the first joint, X axis always points along the common normal (between Z axes). For the first joint, X axis is a free choice. Y axis completes a rectangular reference frame according to the right-hand rule.

Having computed the aforementioned parameters, transformation matrix for the joint under consideration is computed according to expression, [3]:

$$\left[\begin{array}{c:c} \mathbf{R} & \mathbf{T} \\ \hdashline 0 & 1 \end{array}\right] = \left[\begin{array}{ccc:c} \cos\theta & -\sin\theta\cos\alpha & \sin\theta\sin\alpha & r\cos\theta \\ \sin\theta & \cos\theta\cos\alpha & -\cos\theta\sin\alpha & r\sin\theta \\ 0 & \sin\alpha & \cos\alpha & d \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right] \tag{1}$$

The upper left cell **R** contains information about joint attitude in space after rotation whilst the upper right vector **T** contains information about joint translation. The last row [0, 0, 0, 1] never changes so as to augment the matrix to homogeneous one.

In Autodesk Inventor API, a ComponentOccurrence object represents an instance of a component (such as a part or a sub-assembly) within an assembly file. The object has a transformation property that either returns or sets the transformation matrix of the occurrence. In appendix section, the iLogic rule, a component is selected by name (end effector wrist1:1) in order to obtain the transformation matrix in relation to the assembly origin. Evidently, it solely takes a few lines of code to obtain and dump the matrix in DebugView window. No additional computations are required whatsoever.

3. RESULTS

In Fig. 3, a screenshot of Autodesk Inventor development environment shows the robotic arm itself, a control form, and DebugView 3^{rd} party tool. Joints are deflected by moving corresponding slider. As soon as desired attitude has been reached, a transformation matrix describing the end-effector position and orientation might be computed by pressing a button. The matrix is immediately put on DebugView window. The iLogic rule for computing transformation matrix elements might be found in the Appendix section, [4].

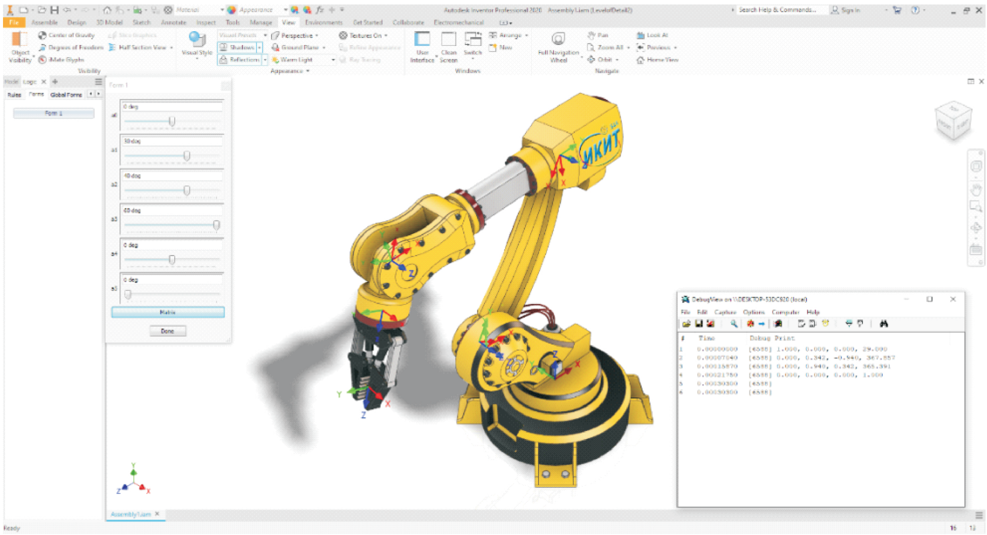

Fig. 3. Control form and transformation matrix of the end-effector attitude

In Fig. 4, transformation matrix elements computed in GNU Octave according to formulae (1) are compared to results obtained by Inventor API.

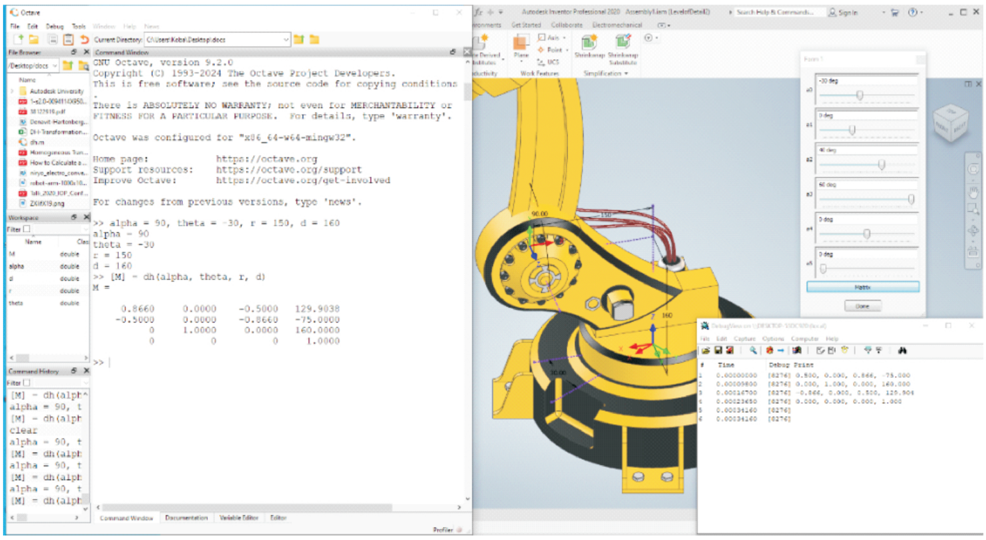

Fig. 4. Results comparison

4. DISCUSSION

Autodesk Inventor provides an excellent opportunity for developer to come up with and put to test new ideas of robot design. It takes some skills in drawing and code developing in VBA to accomplish the project, though. Nevertheless, the ultimate design stage allows for fixing the tiniest terrors, improving arrangement scheme, and even testing an actual control system in the loop by sending / receiving control data through a serial port. All these advantages eventually save time and money prior to commencing the production stage.

The elements of the transformation matrix have been calculated using the two approaches described above. Absolute values of matrix elements coincide.

Project source files might be downloaded from GrabCAD, link [5].

DISCLAIMER

Software bundles used to develop the current project are following:

- Autodesk Inventor, [6] with trial license
- GNU Octave 9.2.0, [7] with free license
- DebugView 4.90, [8] with free license

APPENDIX

iLogic rule for computing transformation matrix, [4]

```
Sub Main()

    MessageBox.Show("Press OK to start", "RoboCop")
    getMatrix()

End Sub

Public Sub DumpMatrix(oMatrix As Matrix)

    Dim i As Integer

    For i = 1 To 4
        Debug.Print (Format(oMatrix.Cell(i, 1), "0.000")
& ", " & _
                    Format(oMatrix.Cell(i, 2), "0.000")
& ", " & _
                    Format(oMatrix.Cell(i, 3), "0.000")
& ", " & _
                    Format(oMatrix.Cell(i, 4),
"0.000"))
    Next

    Debug.Print(vbCrLf)

End Sub

Public Sub getMatrix()

    Dim oAsmDoc As AssemblyDocument
    oAsmDoc = ThisApplication.ActiveDocument

    Dim oOcc As ComponentOccurrence
    oOcc =
oAsmDoc.ComponentDefinition.Occurrences.ItemByName("wris
t1:1")

    DumpMatrix(oOcc.Transformation)

End Sub
```

REFERENCES

[1] B. Ekins, iLogic and the Inventor API, Autodesk University (2016), online: https://static.au-uw2-prd.autodesk.com/handout_17917_SD17917-Ekins-AU2016.pdf

[2] Denavit-Hartenberg Reference Frame Layout, YouTube, uploaded by TekkotsuRobotics, Oct 22, 2009, online: https://youtu.be/rA9tm0gTln8.

[3] A. Owen-Hill, How to Calculate a Robot's Forward Kinematics in 5 Easy Steps, online: https://blog.robotiq.com/how-to-calculate-a-robots-forward-kinematics-in-5-easy-steps.

[4] B. Ekins, How Deep is the Rabbit Hole? Examing the Matrix and other Inventor® Math and Geometry Objects, Autodesk University DE205-2 (2008), online: https://modthemachine.typepad.com/files/mathgeometry.pdf

[5] K. Metodiev, Robotic arm, online: https://grabcad.com/library/robotic-arm-401.

[6] Autodesk, Try Inventor free for 30 days, online: https://www.autodesk.com/products/inventor/free-trial.

[7] GNU Octave, https://octave.org/.

[8] MicroSoft, DebugView v4.90, online: https://learn.microsoft.com/en-us/sysinternals/downloads/debugview.

Received August 04, 2024