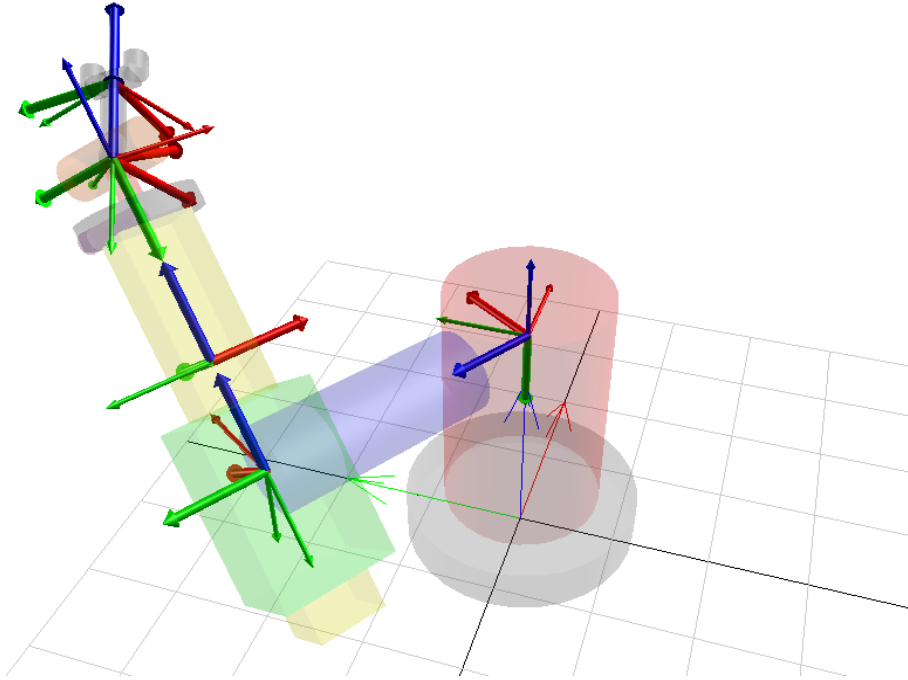


# Robot Manipulators

## Forward Kinematics of Serial Manipulators



**Fig. 1: Stanford Arm**

The focus of this module and the goal of forward kinematics (or direct kinematics) is obtaining the position and orientation of the end-effector of a robot manipulator, with respect to a reference frame, based on the orientations and configuration of the segments that comprise the manipulator. This module introduces the Denavit-Hartenberg (DH) convention and parameters that are used to describe these orientations and configurations, and provide a standard approach to forward kinematics.

### ► **Links and Joints**

### ▼ **Denavit-Hartenberg Convention**

This is a convention used to attach a coordinate system to each link of a manipulator. The coordinate systems are attached according to the following rules :

1. The origin of coordinate system  $i$  is located at the point of intersection of the axis of joint  $i + 1$  and the common normal between the axes of joints  $i$  and  $i + 1$ .
2. The  $z_i$ -axis is aligned with the axis of the  $(i + 1)^{th}$  joint. The positive direction of this axis can be chosen arbitrarily.
3. The  $x_i$ -axis is aligned with the common normal of the  $i^{th}$  and  $(i + 1)^{th}$  joint axes and points from the  $i^{th}$  to the  $(i + 1)^{th}$  joint.
4. The  $y_i$ -axis is determined using the right-hand rule.

The coordinate systems of the ground and end effector links do not follow these rules. The coordinate system of the ground link can be chosen based on convenience as long as the  $z_0$ -axis is aligned with the axis of joint 1. The coordinate system of the end effector, also called the hand coordinate system, can also be chosen based on convenience as long as its  $x$ -axis is normal to the last joint axis.

## Denavit-Hartenberg Parameters

Any serial manipulator can be described kinematically by specifying four parameters for each link.

Regardless of the physical construction of the actual link connecting two joints, their relative location can be described using two parameters:

1.  $a_i$  : length of the common normal between the axes of joint  $i$  and joint  $i + 1$ . This is also known as the *length* of the link.
2.  $\alpha_i$  : angle measured from the axis of joint  $i$  to axis of joint  $i + 1$ . This is also known as the *twist* of the link.

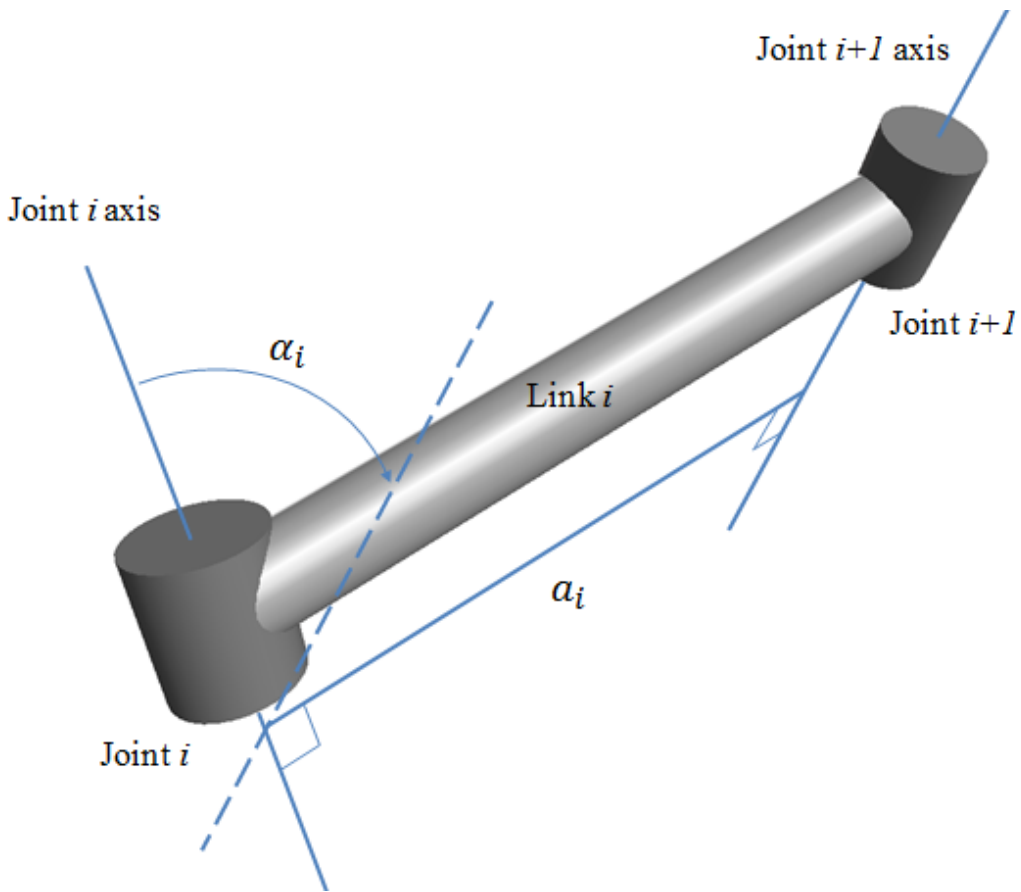


Fig. 3: Length and twist

Similarly, two more parameters can describe the locations of the links relative to each other:

3.  $d_i$ : offset distance between the common normal of the axes of joint  $i - 1$  and joint  $i$  and the common normal of the axes of joint  $i$  and joint  $i + 1$ . This is also known as the *offset* of the link.
4.  $\theta_i$ : angle between the common normal of the axes of joint  $i - 1$  and joint  $i$  and the common normal of the axes of joint  $i$  and joint  $i + 1$ . This is also known as the *angle* of the joint.

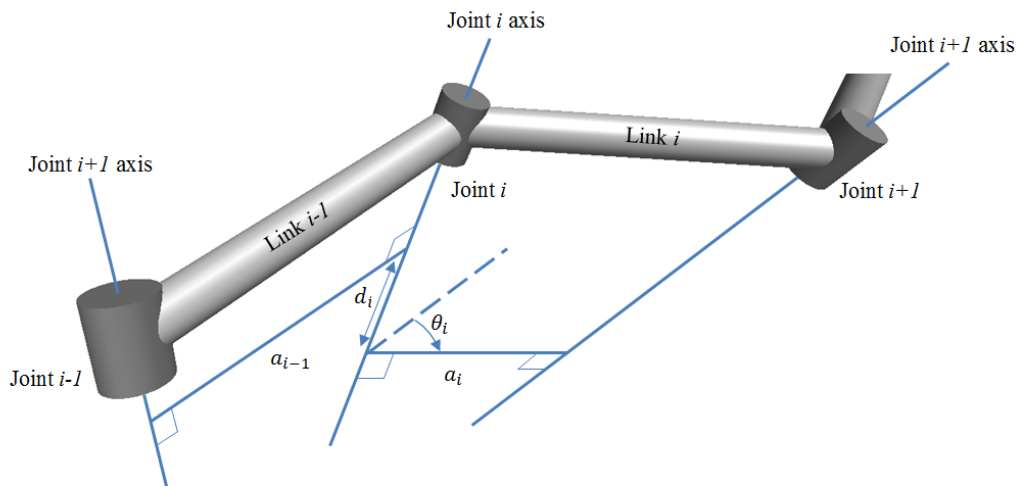


Fig. 4: Offset and angle

For a revolute joint,  $\theta_i$  is the only parameter that is variable and for a prismatic joint,  $d_i$  is the only parameter that is variable.

Table 1: Joint variables and parameters

Joint	Joint variable	Link parameters (constant)
Revolute	Angle $\theta_i$	Length $a_i$ Twist $\alpha_i$ Offset $d_i$
Prismatic	Offset $d_i$	Length $a_i$ Twist $\alpha_i$ Angle $\theta_i$

## Special Cases

1. If the  $i^{th}$  and  $(i+1)^{th}$  joint axes intersect then  $a_i$  is 0.
2. If the  $i^{th}$  and  $(i+1)^{th}$  joint axes are parallel then  $a_i$  is constant and an infinite number of common normals exist. Usually, the common normal that makes  $d_i=0$  is used for attaching the link coordinate system.

3. If the  $i^{\text{th}}$  and  $(i + 1)^{\text{th}}$  joint axes coincide then the origin of the  $i^{\text{th}}$  coordinate system is chosen arbitrarily. The choice of axes that makes most parameters 0 is preferable.

## Denavit-Hartenberg Homogeneous Transformation Matrices

Using the DH convention and parameters, transformation matrices relating two successive coordinate systems can be established. The  $i^{\text{th}}$  coordinate system can be obtained through successive rotations and translations of the  $(i - 1)^{\text{th}}$  coordinate system:

1. The  $(i - 1)^{\text{th}}$  coordinate system is translated along the  $z_{i-1}$ -axis by a distance  $d_i$ . The corresponding transformation matrix is

$$T_d = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

... Eq. (1)

2. The displaced coordinate system is rotated about the  $z_{i-1}$ -axis by an angle  $\theta_i$ . This aligns the displaced  $x_{i-1}$ -axis with the  $x_i$ -axis. The corresponding transformation matrix is

$$T_\theta = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

... Eq. (2)

3. The displaced coordinate system is translated along the  $x_i$ -axis by a distance  $a_i$ . The

corresponding transformation matrix is

$$T_a = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

... Eq. (3)

4. The displaced coordinate system is rotated about the  $x_i$ -axis by an angle  $\alpha_i$ . This is the final step that transforms the  $(i - 1)^{th}$  coordinate system to the  $i^{th}$  coordinate system. The corresponding transformation matrix is

$$T_\alpha = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_i) & -\sin(\alpha_i) & 0 \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

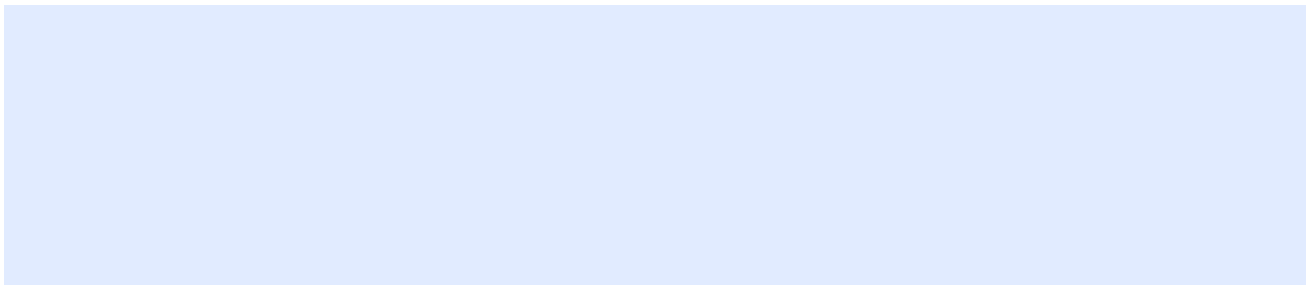
... Eq. (4)

The resulting transformation matrix is

$$T_{i-1,i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_i) & -\sin(\alpha_i) & 0 \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

... Eq. (5)

which can be expanded to



$$T_{i-1,i} = \begin{bmatrix} \cos(\theta_i) & -\cos(\alpha_i) \cdot \sin(\theta_i) & \sin(\alpha_i) \cdot \sin(\theta_i) & a_i \cdot \cos(\theta_i) \\ \sin(\theta_i) & \cos(\alpha_i) \cdot \cos(\theta_i) & -\sin(\alpha_i) \cdot \cos(\theta_i) & a_i \cdot \sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

... Eq. (6)

Multiplying this transformation matrix and the position vector of a point with respect to the  $i^{th}$  coordinate system will give the position vector of the point with respect to the  $(i - 1)^{th}$  coordinate system.

### Example 1: 3-DOF Planar Manipulator

A planar manipulator with 3 degrees of freedom is constructed with three revolute joints. A reference coordinate system is attached to the base such that the manipulator moves in this system's x-y plane. For this manipulator, all the twist angles and offset distances of the links are equal to 0.

a) Find the DH transformation matrices  $T_{01}$ ,  $T_{12}$  and  $T_{23}$  in terms of the following joint variables and link parameters:

Table 2: 3-dof planar manipulator joint variables and link parameters

Joint $i$	$\alpha_i$	$a_i$	$d_i$	$\theta_i$
1	0	$a_1$	0	$\theta_1$
2	0	$a_2$	0	$\theta_2$
3	0	$a_3$	0	$\theta_3$

b) Find the position vector of the end-effector coordinate system with respect to the base reference frame for the following values:

Table 3: 3-dof planar manipulator joint variable and link parameter values

--	--	--	--	--

Joint $t_i$	$\alpha_i$	$a_i$	$d_i$	$\theta_i$
1	0	1.2 m	0	0.3 <i>rad</i>
2	0	1.5 m	0	0.5 <i>rad</i>
3	0	1.7 m	0	-0.6 <i>rad</i>

## Analytical Solution

*restart :*

*with(LinearAlgebra) :*

- ▶ **Part a) Transformation matrices in terms of joint variables and link parameters**
- ▶ **Part b) Position vector of the origin of the end-effector coordinate system with respect to the base reference frame**

## Solution and visualization with MapleSim

### Constructing the model

- ▶ Create an "Axes" subsystem
- ▶ Create a "revolute DH" subsystem

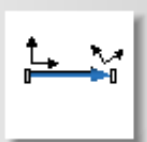
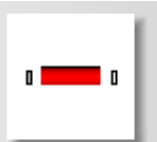

This subsection contains the steps to build a subsystem that can be used to build the manipulator arm link by link using the DH parameters and variables.



## Step 1: Insert components

1. Drag the following components into the main workspace:

Table 5: Components and locations

Component	Location
 Rigid Body Frame  (3 required)	Multibody > Bodies and Frames
 Cylindrical Geometry  (2 required)	Multibody > Visualization
 Revolute	Multibody > Joints and Motions

2. Make 3 additional copies of the Axes subsystem.
3. A dialog box will appear while making copies of the subsystem. Select **Convert to a shared subsystem (Recommended)**. Selecting this option causes any changes made to one subsystem to be automatically made to the others.
4. Right click the fourth copy of the Axes subsystem and click **Convert to Standalone Subsystem**.
5. Double click this subsystem, change the **radius** of all its **Cylindrical Geometry** components to **0.02 m** and all the **radius at frame\_a** values of the

**Tapered Cylindrical Geometry** components to **0.04 m**. This makes the axes of this particular subsystem thicker. This modified Axes subsystem will be used to visualize each link's coordinate system, and the other three axes subsystems will be used to visualize the intermediate axes that show the successive rotations and translations that transform one link's coordinate system to the next link's coordinate system.

### Step 2: Connect components

Arrange and connect the components as shown in the following diagram:

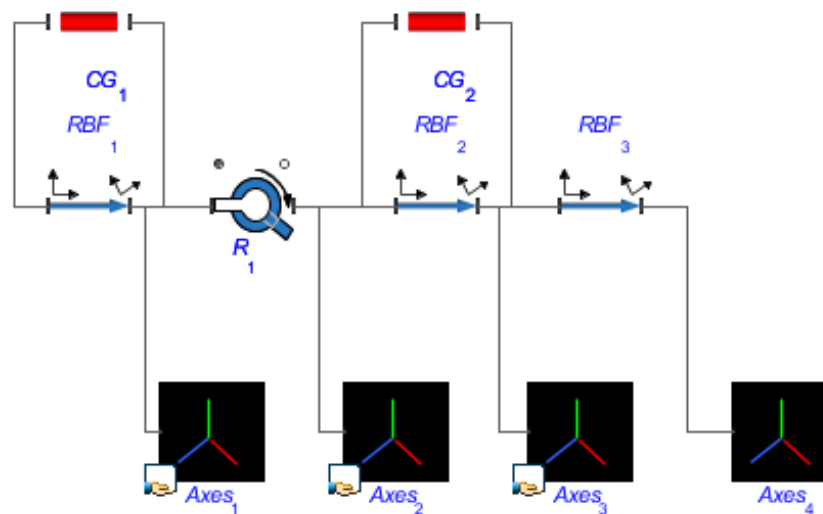


Fig. 8: Component diagram

### Step 3: Create a subsystem

1. Select all the components and their connections.
2. Press **Ctrl+G** to create a subsystem.
3. Enter **DH** for the name of the subsystem and click **OK**.
4. Double-click the subsystem component created in the main workspace.
5. Connect the left port (**frame\_a**) of **RBF<sub>1</sub>** to a point on the dashed border on the left, as shown in Fig. 9.
6. Connect the right port (**frame\_b**) of **RBF<sub>3</sub>** to a point on the dashed border on the right, as shown in Fig. 9.
7. Connect **flange\_b** of the **Revolute** component to a point on the top border, as

shown in Fig 9.

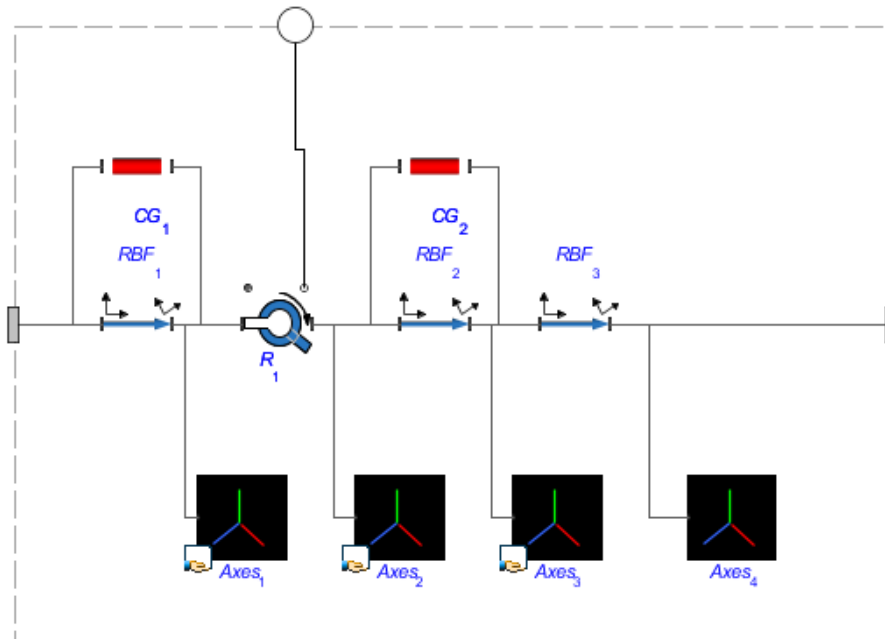


Fig. 9: DH subsystem

8. Click **Add or Change Parameters** in the **Inspector** pane of this subsystem.

9. Create parameters as shown in Fig. 10.

DH subsystem default settings

Name	Type	Default Value	Default Units	Description
$a_i$	Real	1		
$\alpha_i$	Real	1		
$d_i$	Real	1		

Fig. 10: DH parameters

10. Return to the diagram (click ) , click  $RBF_1$  and enter  $[0,0,d_i]$  for the x, y,z offset ( $\vec{r}_{XYZ}$ ).

11. Click  $RBF_2$  and enter  $[a_i,0,0]$  for the x,y,z offset ( $\vec{r}_{XYZ}$ ).

12. Click  $RBF_3$ , enter  $[0,0,0]$  for the x,y,z offset ( $\vec{r}_{XYZ}$ ) and enter  $[\alpha_i,0,0]$  for the **Rotations about the axes** ( $\vec{\theta}_{\zeta\eta\xi}$ ).

13. Click  $CG_1$  , enter  $0.008\ m$  for the **radius** and change the color to purple.


14. Click  $CG_2$ , enter  $0.008\ m$  for the **radius** and change the color to orange

(these **CG** components help visualize the link lengths and offsets).

► **Create a revolute joint visualization subsystem**

► **Construct the manipulator**

▼ **Add a parameter block and run simulation**

1. Click the **Add a parameter block** icon (  ) and click on the main workspace.
2. Double-click the **Parameter block** icon and create the following parameters.

Parameters subsystem default settings

Name	Type	Default Value	Default Units	Description
<i>alpha_1</i>	Real	1		
<i>a_1</i>	Real	1		
<i>d_1</i>	Real	1		
<i>theta_1</i>	Real	1		
<i>alpha_2</i>	Real	1		
<i>a_2</i>	Real	1		
<i>d_2</i>	Real	1		
<i>theta_2</i>	Real	1		
<i>alpha_3</i>	Real	1		
<i>a_3</i>	Real	1		
<i>d_3</i>	Real	1		
<i>theta_3</i>	Real	1		

**Fig. 13: Link parameters and joint variables**

3. Return to the main workspace, click on the **Parameter Block** icon and enter the specified parameter values for the parameters.
4. Click the leftmost **Constant** component (see Fig. 12) and enter **theta\_1** for the constant  $k$ .
5. Click the next **Constant** component and enter **theta\_2** for the constant  $k$ .
6. Click the rightmost **Constant** component and enter **theta\_3** for the constant  $k$ .
7. Click the first **DH** component (connected to the **Fixed Frame**) and enter **a\_1**, **alpha\_1** and **d\_1** for **a<sub>i</sub>**, **alpha<sub>i</sub>** and **d<sub>i</sub>** respectively.
8. Similarly, enter the corresponding parameters for the remaining two **DH** components.
9. For each **Rotational Position** component, click the component and select **exact** under parameters.
10. Attach a **Probe** to the **r** port of the sensor.
11. Click the probe and select **1**, **2** and **3** in the **Inspector** pane.



12. Click the **Absolute Translation Sensor** and select **Inertial** for the **Frame**.
13. Click the **Cylindrical Geometry** components and change their **color** to gray.
14. Reduce the **Simulation duration** to 1 sec (this is a static model).
15. Click **Run Simulation** (  ).
16. Use  in the 3-D workspace toolbar to hide implicit geometry.

Fig. 14 shows the 3-D visualization of the MapleSim model. The intermediate coordinate systems (thinner arrows) allow us to follow the successive rotations and translations of the  $(i - 1)^{th}$  coordinate system that result in the  $i^{th}$  coordinate system.

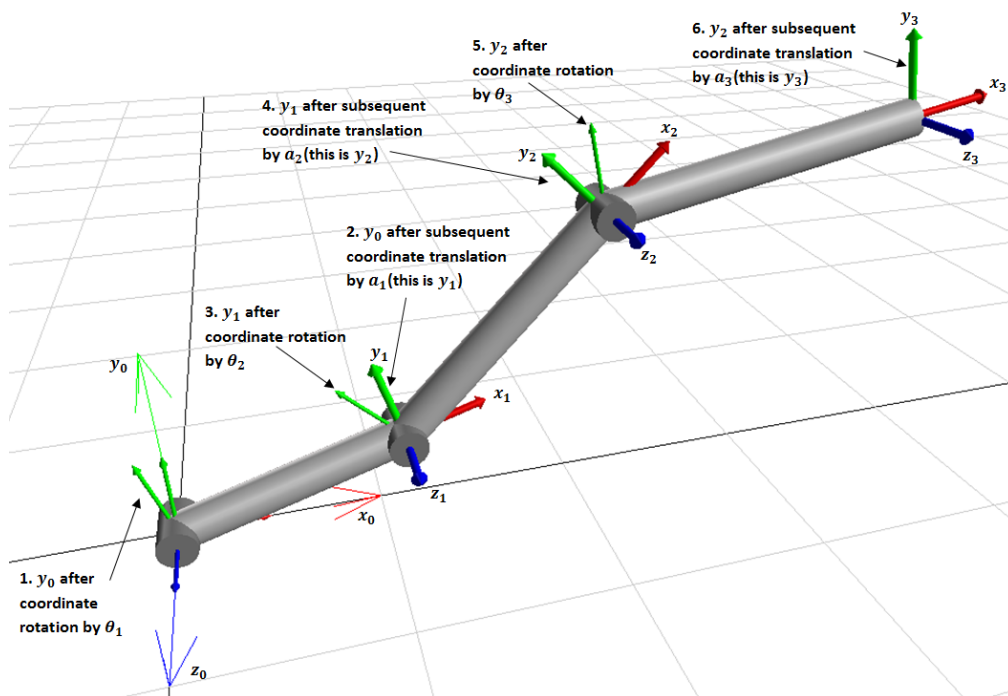
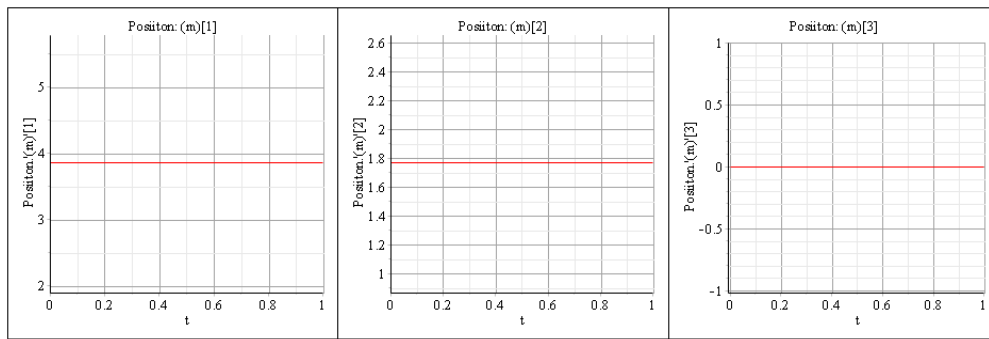


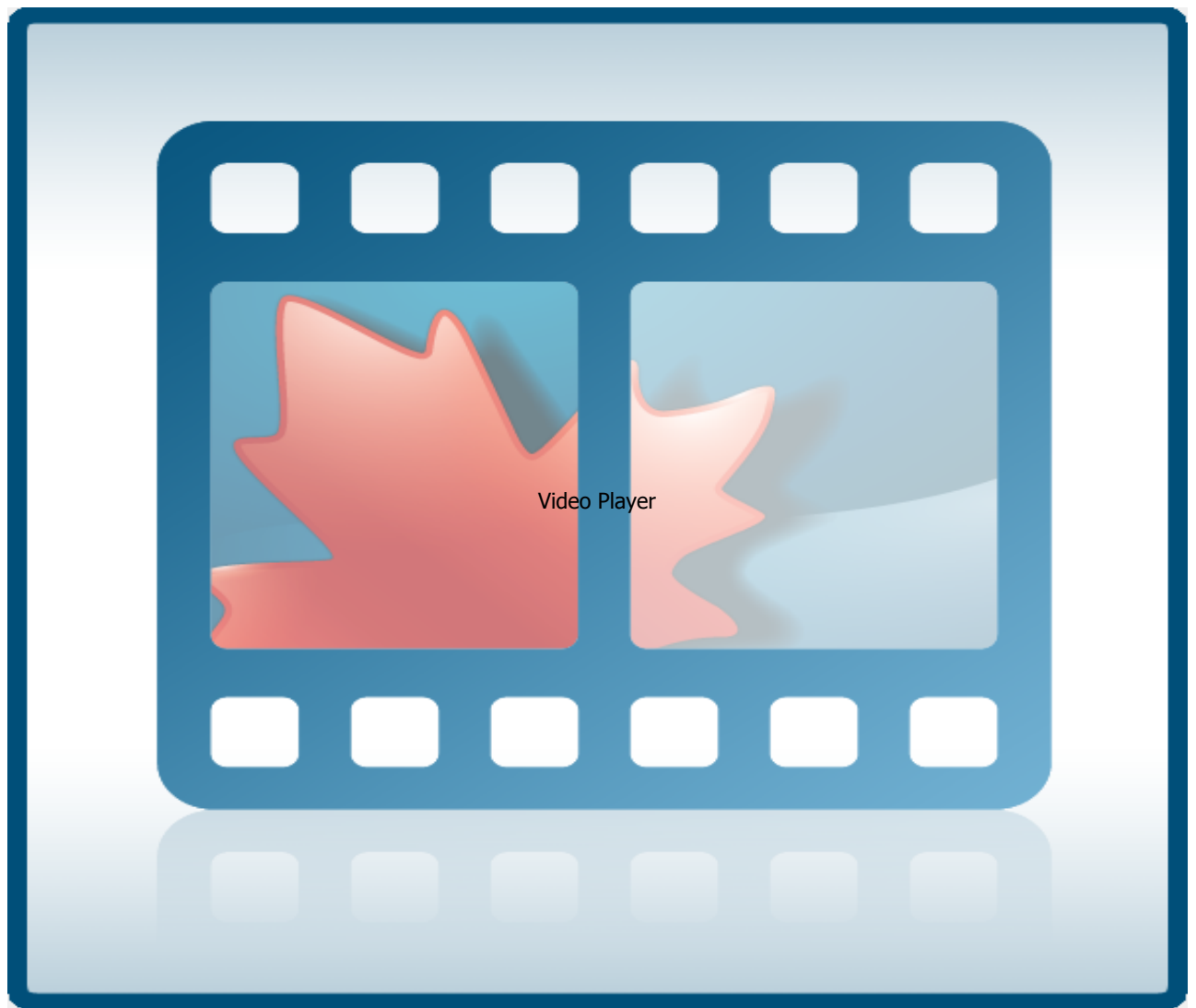
Fig. 14: 3-dof planar manipulator visualization

Fig. 15 shows the results of the simulation. These results match the answer obtained analytically in part b) of the problem.



**Fig. 15: 3-dof planar manipulator simulation results**

The **Constant** components connected to the **Rotational Position** components can be replaced with **Ramp** components to visualize the motion of the arm as the joint variables change from 0 to the specified values. The following video shows this:

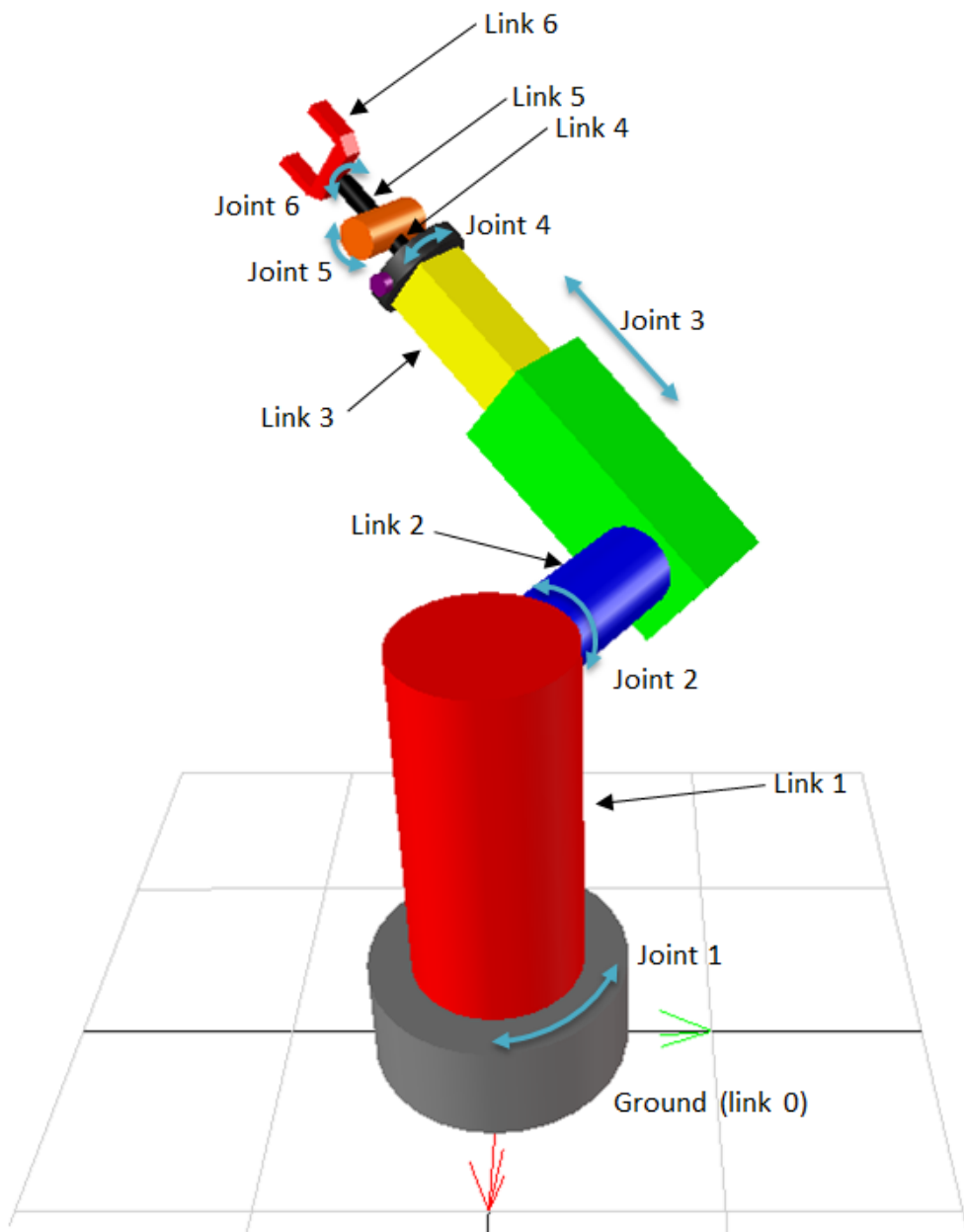


**Video 1: P3-dof planar manipulator motion visualization**

The following diagram shows the modified component diagram with **Ramp** components.







**Fig. 19: Stanford Arm links and joints**

a) Find the DH transformation matrices  $T_{01}$ ,  $T_{12}$ ,  $T_{23}$ ,  $T_{34}$ ,  $T_{45}$  and  $T_{56}$  in terms of the following joint variables and link parameters:

**Table 10: Link parameters and joint variables**

Joint $i$	$\alpha_i$	$a_i$	$d_i$	$\theta_i$
1	$-\frac{\text{Pi}}{2}$	0	$d_1$	$\theta_1$
2	$\frac{\text{Pi}}{2}$	0	$d_2$	$\theta_2$
3	0	0	$d_3$	$-\frac{\text{Pi}}{2}$
4	$-\frac{\text{Pi}}{2}$	0	$d_4$	$\theta_4$
5	$\frac{\text{Pi}}{2}$	0	0	$\theta_5$
6	0	0	$d_6$	$\theta_6$

b) Find the position vector of the end-effector coordinate system with respect to the base reference frame for the following values:

Table 11: Link parameter and joint variable values

Joint $i$	$\alpha_i$	$a_i$	$d_i$	$\theta_i$
1	$-\frac{\text{Pi}}{2}$ <i>rad</i>	0 m	1.3 m	1 rad
2	$\frac{\text{Pi}}{2}$ <i>rad</i>	0 m	1.4 m	0.7 rad
3	0 rad	0 m	0.5 m	$-\frac{\text{Pi}}{2}$ <i>rad</i>
4	$-\frac{\text{Pi}}{2}$ <i>rad</i>	0 m	0.9 m	0.6 rad
5	$\frac{\text{Pi}}{2}$ <i>rad</i>	0 m	0 m	0.6 rad

	$ra\backslash$ $d$			
<b>6</b>	$0 \text{ rad}$	$0 \text{ m}$	$0.4 \text{ m}$	$1.3 \text{ rad}$

## Analytical Solution

*restart :*

*with(LinearAlgebra) :*

- ▶ **Part a) Transformation matrices in terms of joint variables and link parameters**
- ▶ **Part b) Position vector of the origin of the end-effector coordinate system with respect to the base reference frame**


## Solution and Visualization with MapleSim

### Constructing the model

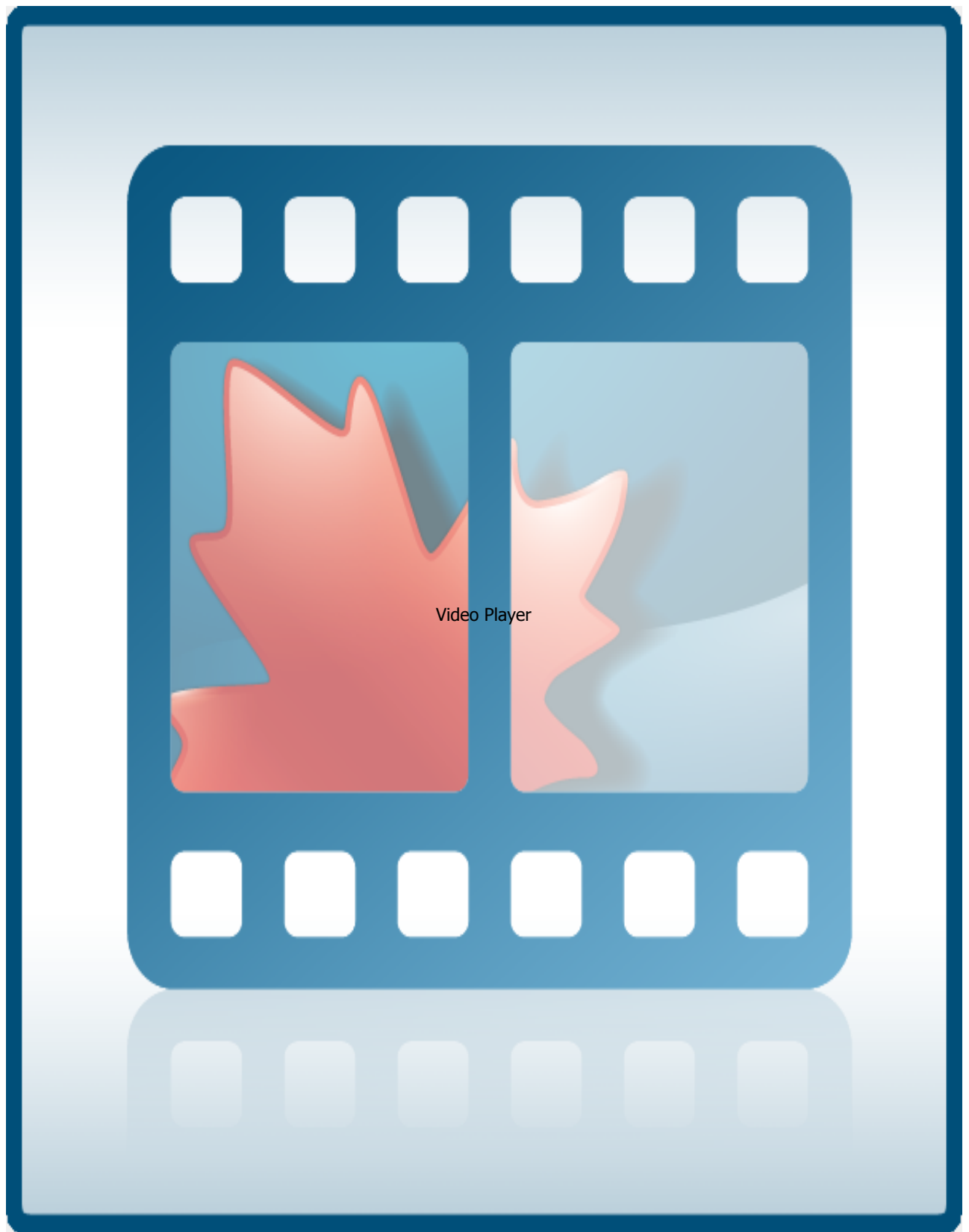
- ▶ **Modify the revolute DH subsystem and create a prismatic DH subsystem**
- ▶ **Set up the simulation**
- ▶ **Set up the visualization**

### Run the simulation

1. Attach an **Absolute Translation** sensor component to the rightmost port of the 6<sup>th</sup> **DH** component.
2. Click the sensor and select **Inertial** for the **Frame**.
3. Attach a **Probe** to the **r** port of this sensor.
4. Click the probe and select **1,2** and **3** in the **Inspector** pane.

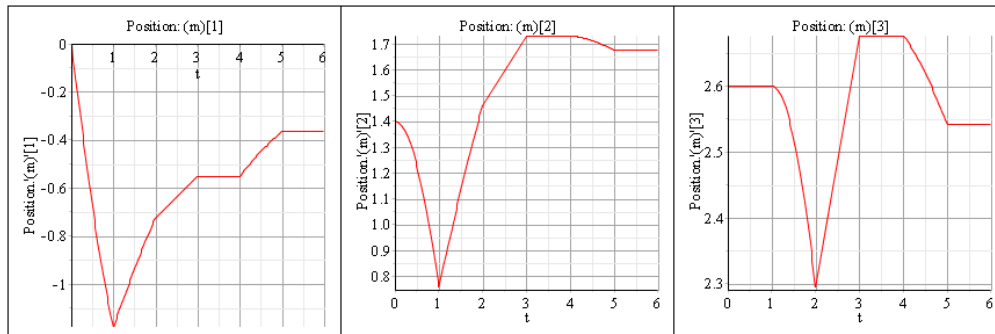
5. Reduce the **Simulation duration** to 1 sec (this is a static model).
6. Click **Run Simulation** (  ).

Similar to Examples 1 and 2, the **Constant** components connected to the **Rotational Position** components can be replaced with **Ramp** components to visualize the motion of the arm as the joint variables change from 0 to the specified values. The following video shows this motion:



**Video 3: Stanford arm motion visualization**

Fig. 31 shows the simulation results of this simulation (using **Ramp** components). Each joint variable changes linearly from 0 to its specified value over a period of 1 second. The sequence of motions is in the same order as the joint numbering.



**Fig. 31: Simulation results with motion**

The position at the end of the simulation matches the position vector obtained in part b) of the analytical solution.

#### References:

1. L-W. Tsai. "Robot Analysis: The Mechanics of Serial and Parallel Manipulators". NY, 1999, John Wiley & Sons, Inc.
2. W. W. Melek. "ME 547: Robot Manipulators: Kinematics, Dynamics, and Control". Waterloo, ON, 2010, University of Waterloo.