Revamping Robotica

Mohammad Sultan, Aaron T. Becker
mmsultan@uh.edu, atbecker@uh.edu .

UNIVERSITY of HOUSTON
CULLEN COLLEGE of ENGINEERING
Department of Electrical & Computer Engineering


New Robotica

Robotica is a collection of useful robotics problem-solving functions encapsulated in a Mathematica package, authored by John Nethery and Mark W. Spong in 1993.

New Functions

`dhInput[ex]` Function:

- Now accepts DH parameters as a matrix as well as old method using .txt file.
- Alternately, user can enter the DH parameters using a dialogue box.
- Both new ways work inside Mathematica.

`drawRobot[]` Function:

- After setting DH parameters using dhInput[ex] function…
- …call `drawRobot[]` function to draw your own robot!

New Robotica version available at:
https://github.com/RoboticSwarmControl/robotica/

Evolution of `dhInput[ex]`

Method 1 (old):
For full backwards compatibility, we still support using .txt files

```
A Robotica Input data for 6-Link Arm
------------------------------------
DOF = 6
The Denavit-Hartenberg parameters:
joint1  = revolute
a1      = 0
alpha1  = -Pi/2
d1      = 1
theta1  = q1
joint2  = revolute
a2      = 0
alpha2  = Pi/2
d2      = 1
theta2  = q2
joint3  = prismatic
a3      = 0
alpha3  = 0
d3      = d3
theta3  = 0
joint4  = revolute
a4      = 0
alpha4  = -Pi/2
d4      = 1/2
theta4  = q4
joint5  = revolute
a5      = 0
alpha5  = Pi/2
d5      = 0
theta5  = q5
joint6  = revolute
a6      = 0
alpha6  = 0
d6      = 1/2
theta6  = q6
```

`DataFile["arm6dof.txt"]` used to result in

```
Kinematics Input Data
---------------------

Joint   Type        a   alpha    d     theta

1       revolute    0   -Pi/2.   1     q1
2       revolute    0   Pi/2.    1     q2
3       prismatic   0   0        d3    0
4       revolute    0   -Pi/2.   0.5   q4
5       revolute    0   Pi/2.    0     q5
6       revolute    0   0        0.5   q6
```

Method 2:
Using `dhInput[ex]` easily create a robot specified by DH parameters:

```
ex={{r,R,p,Revolute,revolute, r},
{0,0,0,0,0,0},
{-Pi/2,Pi/2,0,-Pi/2,Pi/2,0},
{1,1,q3,1/2,0,1/2},{q1,q2,0,q4,q5,q6}};
dhInput[ex];
```

Method 3:
Also using `dhInput[ex]` with `ex = dof`:

```
ex = 6;
dhInput[ex];
```

shows the following dialogue box to fill:

after filling

The new `dhInput[ex]` methods all give the same result, but now with:

Joint	Type	r	α	d	θ
1	revolute	0	$-\frac{\pi}{2}$	1	θ_1^*
2	revolute	0	$\frac{\pi}{2}$	1	θ_2^*
3	prismatic	0	0	d_3^*	0
4	revolute	0	$-\frac{\pi}{2}$	$\frac{1}{2}$	θ_4^*
5	revolute	0	$\frac{\pi}{2}$	0	θ_5^*
6	revolute	0	0	$\frac{1}{2}$	θ_6^*

- Formatted output
- Compact representation
- Aligned rows
- Symbolic representation

All three methods support the following functions:

```
FKin[];                          computes A & T matrices and Jacobian
SimplifyTrigNotation[];          converts math to a common shorthand
Table[A[i]//MatrixForm,{i,1,6}]  displays formatted output
Simplify[T[0,6]]//MatrixForm     simplifies expression
```

drawRobot[]

This example shows how to build and animate a robot in Mathematica:

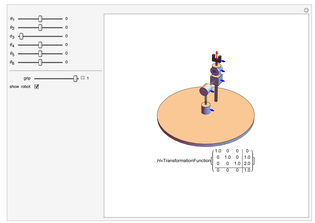
```
SetDirectory[NotebookDirectory[]]
<<robotica_dhInput_drawingRobot.m

ex={{r,R,p,Revolute,revolute, r},{0,0,0,0,0,0},{-Pi/2,Pi/
2,0,-Pi/2,Pi/2,0},{1,1,q3,1/2,0,1/2},{q1,q2,0,q4,q5,q6}};
```

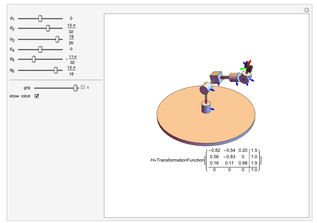
`dhInput[ex];`

Joint	Type	r	α	d	θ
1	revolute	0	$-\frac{\pi}{2}$	1	θ_1^*
2	revolute	0	$\frac{\pi}{2}$	1	θ_2^*
3	prismatic	0	0	d_3^*	0
4	revolute	0	$-\frac{\pi}{2}$	$\frac{1}{2}$	θ_4^*
5	revolute	0	$\frac{\pi}{2}$	0	θ_5^*
6	revolute	0	0	$\frac{1}{2}$	θ_6^*

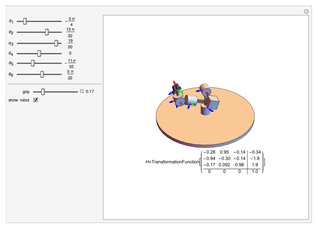
`drawRobot[]` This function shows your robot initialized with joints set to zero. You can manipulate each joint using the sliders



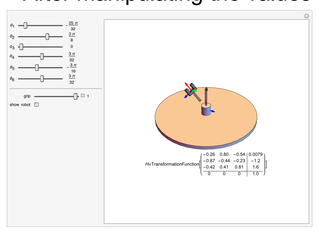
With default values



After manipulating the values

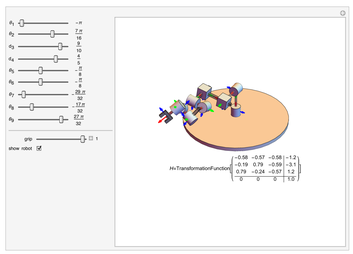


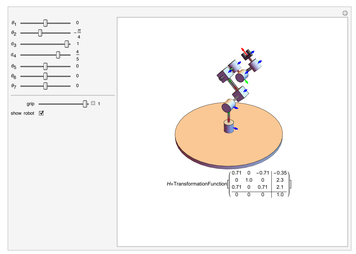
After changing grip value

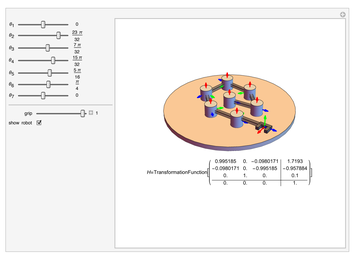


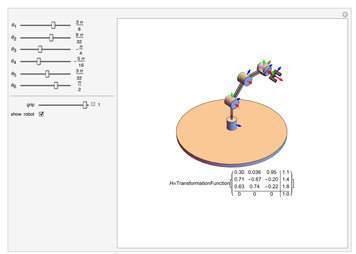
After turning off *show robot*

The following are different robots with a range of DOF:



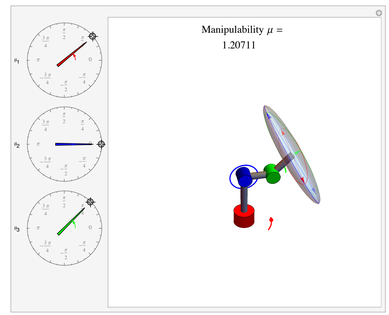


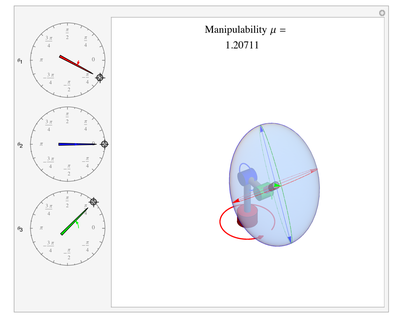




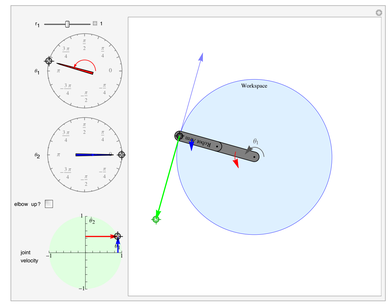
Wolfram Demonstrations Projects

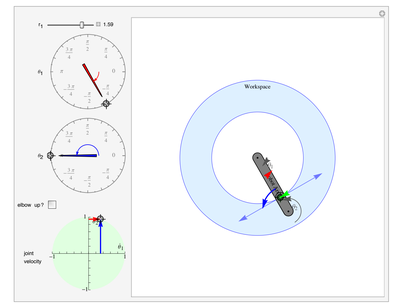
Demonstrations built by Intro to Robotics students, available at demonstrations.wolfram.com



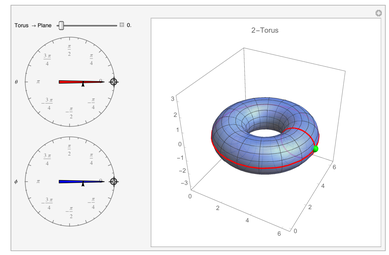


Manipulability Ellipsoid of Robot arm





Two Link Arm workspace



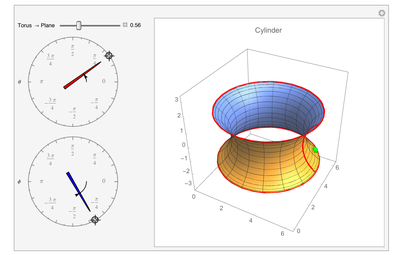
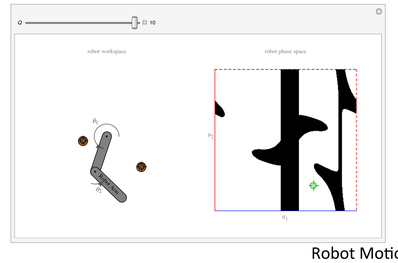
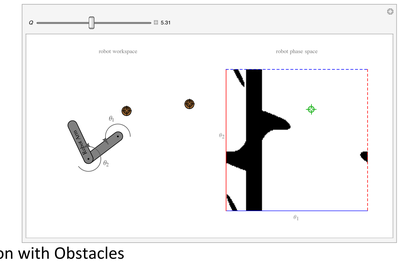
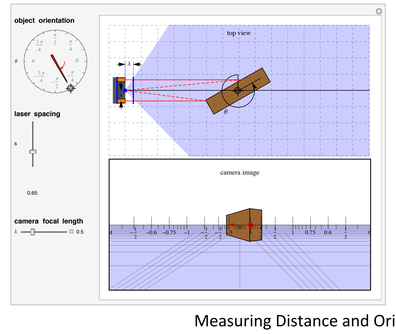


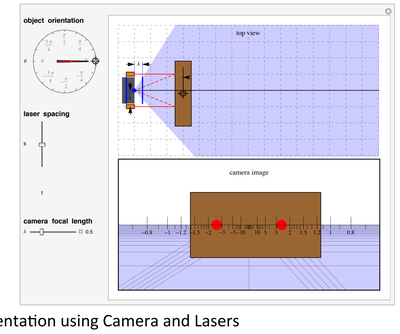
Chart for a Torus





Robot Motion with Obstacles





Measuring Distance and Orientation using Camera and Lasers