Introduction

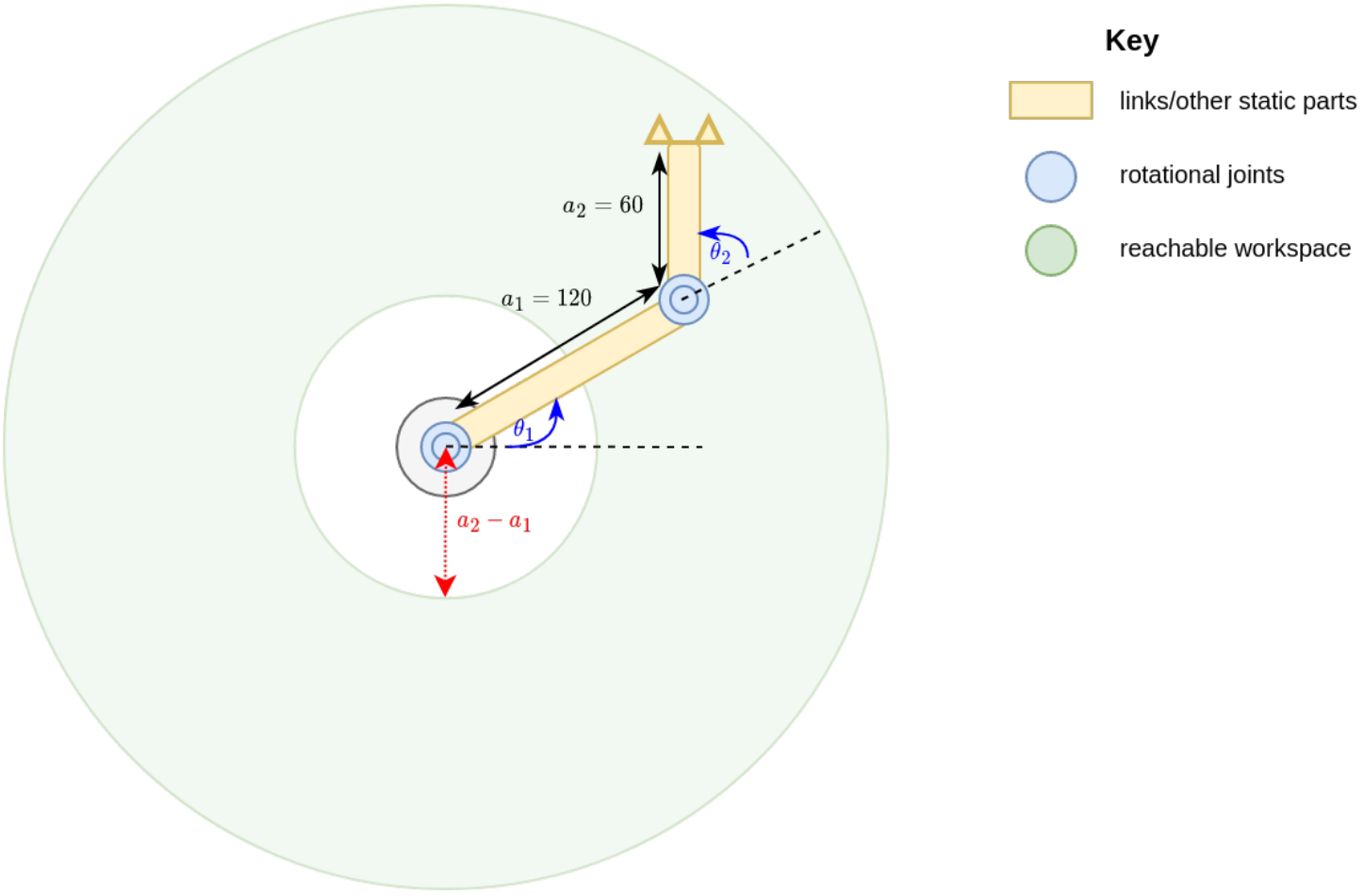
This live script contains examples of reachable workspaces for 2D robots. For each example this includes:

- DH Parameters
- Theoretical Workspace
- Generated Workspace

```
syms theta1 theta2 theta3 real
syms d1 d2 real
```

Example 1

This is a 2 DOF planar robot with rotational joints (unconstrained), where the appropriate reachable workspace that can be determined by the extreme positions is highlighted.



Below is the according code with DH parameters to approximate this space.

```
% DH parameters
test_dh = [0 0 0 theta1; ...
           120 0 0 theta2; ...
```

```
            60 0 0 0]
```

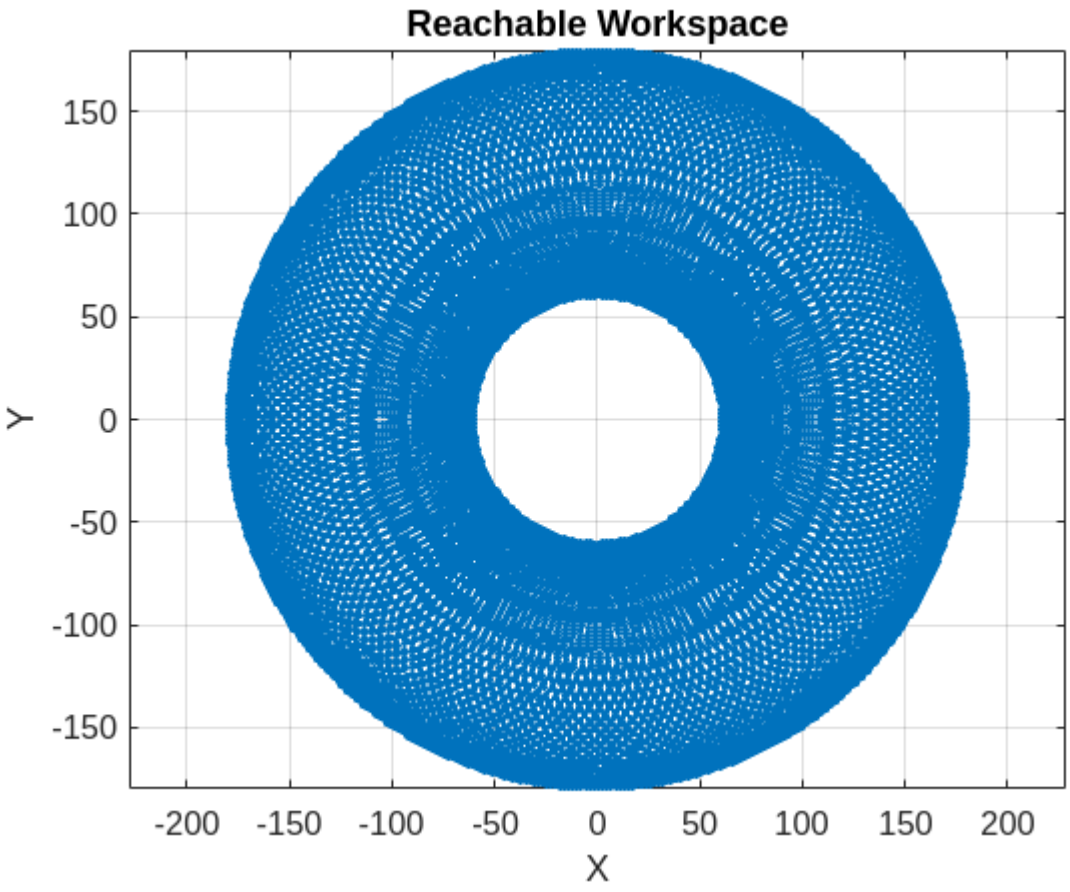
```
test_dh =
```

$$\begin{pmatrix} 0 & 0 & 0 & \theta_1 \\ 120 & 0 & 0 & \theta_2 \\ 60 & 0 & 0 & 0 \end{pmatrix}$$

```
% Parameter ranges
theta1_range = linspace(0,2*pi, 180);
theta2_range = linspace(0,2*pi, 180);
test_map = containers.Map({'theta1', 'theta2'},{theta1_range,theta2_range})
```

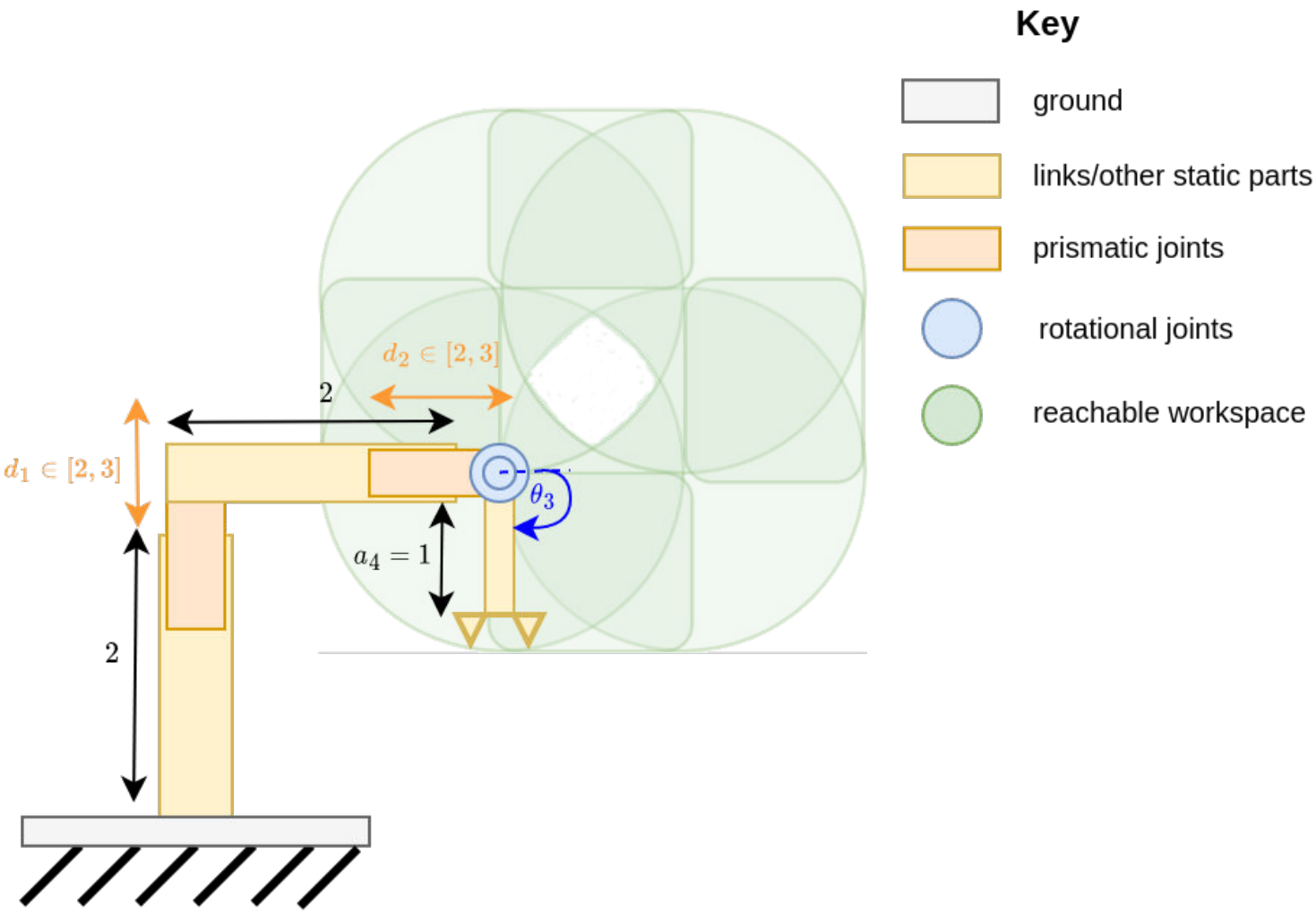
```
test_map =
  Map with properties:

       Count: 2
     KeyType: char
   ValueType: any
```

```
% Workspace plotting function + timing
plot2dworkspace(test_dh, test_map)
```



Example 2

This is an example of a 3 DOF robot (2 prismatic, 1 rotational). Note here that the two prismatic joints each have a maximum extension of 1, which ranges the DH parameters $d_1, d_2 \in [2, 3]$ when we factor in the lengths of the links they extend from. The rotational joint θ_3 is unconstrained. The reachable workspace is less intuitive in this case, but can be determined by drawing out the extreme positions of each joint.



Below

```
% DH parameters
test_dh = [0 0 d1 0; ...
           0 -pi/2 d2 pi/2; ...
           0 -pi/2 0 theta3; ...
           1 0 0 0]
```

```
test_dh =
```

$$\begin{pmatrix} 0 & 0 & d_1 & 0 \\ 0 & -\frac{\pi}{2} & d_2 & \frac{\pi}{2} \\ 0 & -\frac{\pi}{2} & 0 & \theta_3 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

```
% Parameter ranges
d1_range = linspace(2,3, 50);
d2_range = linspace(2,3, 50);
theta3_range = linspace(0,2*pi, 180);
test_map = containers.Map({'d1', 'd2', 'theta3'},{d1_range,d2_range,
theta3_range})
```

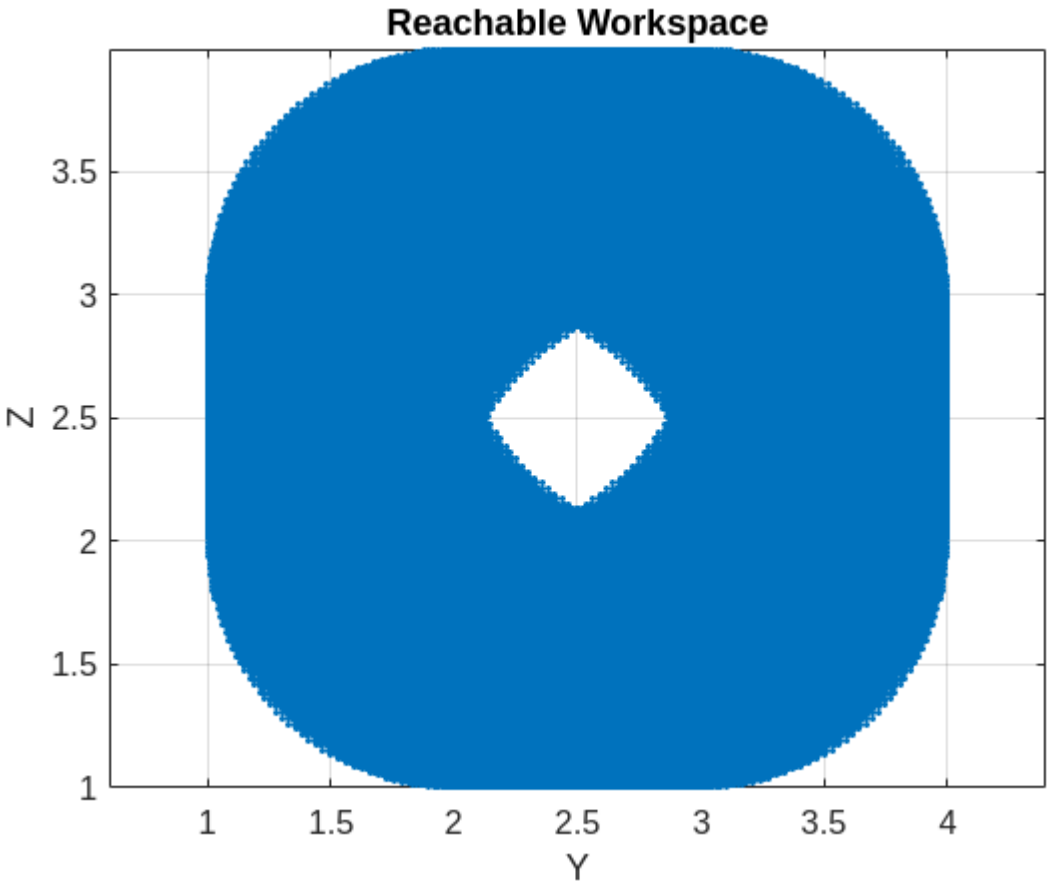
```
test_map =
  Map with properties:

        Count: 3
      KeyType: char
    ValueType: any
```

```
% Runs plot command with additional optional parameters, including verbose
% logging. Shows various details like transformation matrix used, number of
% positions calculated, etc.
tic
plot2dworkspace(test_dh, test_map, 0.001, true)
```

```
final transformation matrix from base to end-effector:
```

$$\begin{pmatrix} 0 & 0 & -1 & 0 \\ -\sin(\theta_3) & -\cos(\theta_3) & 0 & d_2 - \sin(\theta_3) \\ -\cos(\theta_3) & \sin(\theta_3) & 0 & d_1 - \cos(\theta_3) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

```
expressions for position:
x = 0
y = d2 - sin(theta3)
z = d1 - cos(theta3)
allocating 450000 possible DH parameter combinations...
calculating positions from expressions...
```




```
toc
```

```
Elapsed time is 0.200674 seconds.
```

```

```

```
% A somewhat convoluted PDF export procedure to save a PDF of the .mlx file
% for github. You can ignore this.
file_name = 'plot2dworkspace_examples';
current_mlx = which(file_name);
[path_to_file, name, ext] = fileparts(current_mlx);
mlx_path = fullfile(path_to_file, (file_name + ".mlx"));
pdf_path = fullfile(path_to_file, (file_name + ".pdf"));
export(mlx_path, pdf_path);
```