Trevor Oshiro
MAE263B

Project 3 Report

Part 1: Jacobian Derivation

Within this project, the derivation of the Jacobian of the SCARA manipulator was carried out with the implementation of the robotics toolbox for this course. Three m function files were generated from this toolbox to automate the calculations for the Jacobian through the explicit, velocity propagation, and force-torque propagation methods. These scripts can be found within the repository linked below, where the main file titled “MAE263B_Project3_TrevorOshiro.m” outputs the jacobian matrix from the respective methods when run.

Github repository folder link:
https://github.com/rovertio/MAE263B_TrevorOshiro/tree/main/JacobianDerivation

```
-----------------------------------------------------------------
Explicit method of derivation: Jmethod_exp.m file
ans =
[- a3*sin(th1 + th2) - a2*sin(th1), -a3*sin(th1 + th2), 0, 0]
[  a3*cos(th1 + th2) + a2*cos(th1),  a3*cos(th1 + th2), 0, 0]
[                                0,                  0, 0, 1]
[                                0,                  0, 0, 0]
[                                0,                  0, 0, 0]
[                                1,                  1, 1, 0]

-----------------------------------------------------------------
```

Figure 1: Output jacobian from explicit method

```
-----------------------------------------------------------------
Velocity Propagation method of derivation: Jmethod_vp.m file

Variables used for Jacobian
vel_var =
[vt1, vt2, vt3, vd4, vd5]

ans =
[- a3*sin(th1 + th2) - a2*sin(th1), -a3*sin(th1 + th2), 0, 0]
[  a3*cos(th1 + th2) + a2*cos(th1),  a3*cos(th1 + th2), 0, 0]
[                                0,                  0, 0, 1]
[                                0,                  0, 0, 0]
[                                0,                  0, 0, 0]
[                                1,                  1, 1, 0]

-----------------------------------------------------------------
```

Figure 2: Output jacobian from velocity propagation method

```
------------------------------------------------------------------
Force Torque Propagation method of derivation: Jmethod_ft.m file

End Effector force values used for Jacobian
F_var =
[fx, fy, fz]

End Effector moment values used for Jacobian
T_var =
[nx, ny, nz]

ans =
[- a3*sin(th1 + th2) - a2*sin(th1), -a3*sin(th1 + th2), 0, 0]
[  a3*cos(th1 + th2) + a2*cos(th1),  a3*cos(th1 + th2), 0, 0]
[                                0,                  0, 0, 0]
[                                0,                  0, 0, 0]
[                                0,                  0, 0, 0]
[                                1,                  1, 1, 1]

------------------------------------------------------------------
```

Figure 3: Output jacobian from force torque propagation method

If the derivation code is analyzed, the offset of the tool end effector can be seen to be treated as a link within the calculations. This was, however, done for convenience in generating the necessary matrices for the derivation, and the contribution of the “link” from the offset was omitted in the final result by the removal of the column associated with its contribution in the final result. The jacobian was also derived such that the input joint values come in the following vector: [theta1, theta2, theta3, d4].

Part 2: Singularities

Within the system, there exists four degrees of freedom from the three revolute joints and the prismatic joint. However, there exists 6 total degrees of freedom within translational and rotational movement in space, so the jacobian would be reduced to allow for inversion during the motion planning calculations. As seen with the two rows of zeros in the derivations of the jacobian matrices in the above figures, one can see that the motion of the joints of the manipulator don’t impact rotational velocities along the x and y axis in the base frame. Thus, these rows can be eliminated in the reduction of the jacobian matrix to make the motion of the joints impacting the translational movements and rotation along the z axis only. This reduction can be seen with the output below from MATLAB:

```
-----------------------------------------------------------------
Reduced jacobian matrix:
Red_Jac =
[- a3*sin(th1 + th2) - a2*sin(th1), -a3*sin(th1 + th2), 0, 0]
[  a3*cos(th1 + th2) + a2*cos(th1),  a3*cos(th1 + th2), 0, 0]
[                               0,                   0, 0, 1]
[                               1,                   1, 1, 0]

-----------------------------------------------------------------
```

Figure 4: Reduced jacobian output from MATLAB

Singularity Occurrence:

From the mathematical definition, singularities within the robot arise when the jacobian for the system has a determinant of zero, which prevents inversion of the jacobian. This is then found within the analytical derivation shown in the figure below:

$$(a2)(a3)\cos(\theta_1+\theta_2)\sin(\theta_1) - (a2)(a3)\sin(\theta_1+\theta_2)\cos(\theta_1) = 0$$

$$(a2)(a3)\left[\sin(\theta_1)\cos(\theta_1+\theta_2) - \cos(\theta_1)\sin(\theta_1+\theta_2)\right] = 0$$

$$\sin(\theta_1 - (\theta_1+\theta_2)) = 0$$

$$\sin(\theta_2) = 0$$

$$\theta_2 = 0, \pi, 2\pi \cdots$$

Figure 5: Analysis of the determinant of the jacobian

This shows that a singularity is reached when the elbow joint, theta 2, has a value that is an integer multiple of pi, regardless of the value of the shoulder joint angle. This indicates there being a singularity when the elbow of the SCARA manipulator is fully extended or fully bent inwards/outwards.

Singularity Implications:

Velocity Implications:

Conceptually, this can be described as there being an extremely limited degree of freedom in motion possible at the fully extended orientation of the joints. Each joint would be able to only contribute a motion along a single curve, as opposed to the ellipsoid shape generated general configurations, represented with a radius of the link length's sum. When another position is mapped to the joint velocity values from the jacobian from a singularity, this then implies excessively high joint velocities necessary to achieve the desired position.

Load Capacity Implications:

Within the context of load bearing, the singularity position of the manipulator indicates the system being prone to failure with the singularity position. This corresponds to loads primarily applied along the axis of the arm's extent. When this singularity is reached, the torques/forces from the joints are unable to counteract the load applied at the end effector, leading to collapse and damage of manipulator components. This can also be seen mathematically when setting the angle values of the jacobian to be within the singularity configurations, and the figure below shows an example calculation for the forces generated at a singularity configuration:

```
-----------------------------------------------------------------
Stretched along x axis:
ans =
[     0,      0, 0, 0]
[0.5500, 0.2250, 0, 0]
[     0,      0, 0, 1]
[     1,      1, 1, 0]

-----------------------------------------------------------------
-----------------------------------------------------------------
Stretched along y axis:
ans =
[-0.5500, -0.2250, 0, 0]
[      0,       0, 0, 0]
[      0,       0, 0, 1]
[      1,       1, 1, 0]

-----------------------------------------------------------------
-----------------------------------------------------------------
Stretched when 60 degrees and force along arm axis:
s30 =
[-0.4763, -0.1949, 0, 0]
[ 0.2750,  0.1125, 0, 0]
[      0,       0, 0, 1]
[      1,       1, 1, 0]

Row reduction of the euqations for force
ans =
[1, 0,      0]
[0, 1, 0.4091]

-----------------------------------------------------------------
```

Figure 6: Jacobian matrix when elbow is fully extended and the shoulder is rotated 60 degrees

The row reduction in figure six was implemented by forming a matrix with the contributions of force from the shoulder and elbow joints (as they were the only non-zero contributions), with a force directed along the axis of the extended arm represented by the left-most column. From the row reduction, it can be seen that the components of force from joints and the jacobian aren't independent of each other. This indicates the necessary components of force can't then be solved for, making there no possible input torque from the joints to generate force along the direction. On the other hand, the manipulator would then be only able to resist loads perpendicular to the extent of the arm. Seen with the singularity jacobians along the x and y axis, the force component perpendicular to the axis is then the only component that is solvable as the corresponding row in the jacobian contains non-zero values.

Mathematical Implications:

From a mathematical perspective, singularities configurations introduce redundancy within the Jacobian matrix. This can be seen with rows of zeros present in the jacobian when row-reduced. As discussed earlier, a row of zeros indicates loss of a degree of freedom in control of the force or velocity of the manipulator's tool. The jacobian enables control of the manipulator through mapping between the task and joint space of the system. Thus, singularities interrupt mapping from the jacobian matrix by eliminating degrees of freedom.

Mapping can be represented through the graph of the joint space as a unit circle, representing achievable values of the rotational velocity of the joints. For general configurations of the manipulator, the jacobian maps the unit circle in the joint space to an ellipsoid shape in the task space generated by the resulting possible directions of velocities/forces achievable. Closer proximity to angle values associated with singularity configurations then decreases the mapped plot in the task space from the jacobian. This is graphically seen as the plot from mapping transforming into a single line at singularity configurations from the normal ellipsoid shape.

From the jacobian matrix, one can also determine that a possible method of reducing instances of singularities includes adjustment of the link lengths. Depending on the reachable locations necessary, one may consider lengthening or shortening lengths of links to prevent the manipulator from reaching configurations of the elbow fully bent inwards or extended. Performance can be analyzed mathematically through the area of the ellipsoid in the task space, as maintaining a large area from the possible motion in the task space optimizes the effort/power necessary to move the end-effector to the position as well.