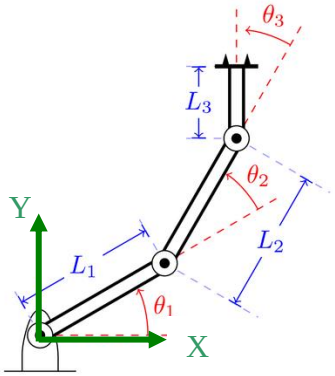
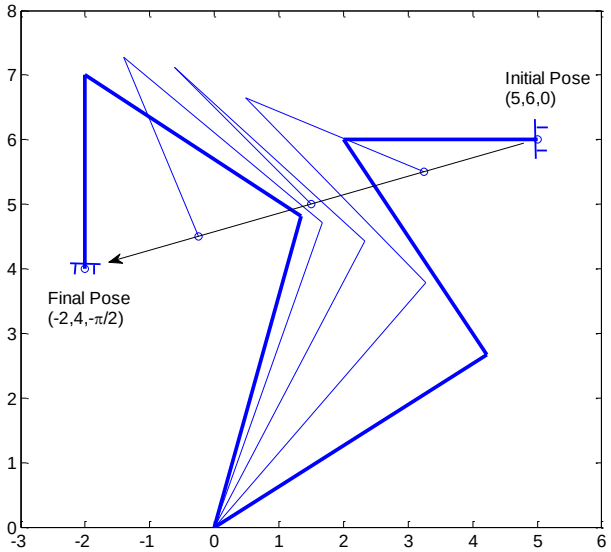
Intuitive Surgical
Systems Analyst Design Problem
Version 1.3.3

Problem statement:
Consider the three-link planar serial manipulator show below:



1. Derive, by hand, the analytic manipulator Jacobian matrix relating joint velocities to endpoint Cartesian *translational* velocities.
2. Indicate the joint configurations where the manipulator is singular, and explain why and the implications for control of the end-effector motion. How does the singularity condition change if orientation of the end-effector is included in the Jacobian?
3. You are asked to write control software to command the end-effector in a pick and place task. Initial and final configurations can be represented by the tuple (x,y,ϕ), where x and y describe Cartesian translation of the end-effector and ϕ is the orientation of the last link with respect to the X axis. The desired end-effector trajectory is a straight path at constant speed in translation and orientation. Write a C program that computes and prints out the required joint angles and velocities to execute a requested trajectory at N points evenly spaced between the initial and final configuration. Use lengths $L_1 = 5$, $L_2 = 4$, and $L_3 = 3$. As an example, the desired trajectory of the manipulator for a move from (5,6, 0) to (-2,4, -π/2) looks like:



Your program should read in the initial configuration (x_0,y_0,ϕ_0), final configuration (x_1,y_1,ϕ_1), desired time to complete the move and number of intermediate points, N, to compute. The program should print out N rows, where each row lists the manipulator's joint angles and velocities at the i^{th} point along the trajectory. Indicate if any pose is unreachable. Two matrix math functions provided below may be used or adapted in your program.

Provide your source code with comments, any instructions to compile and run, input file (if necessary), and a copy of the program output. As a test input, command a move from (3,3,0) to (5,5,0) over two seconds and output 10 points.

Please deliver the results in electronically readable form via email no later than 48 hours from receipt of the assignment. You are welcome to scan any handwritten work. Please include this cover page with your results.

Good luck!

Supplementary Code:

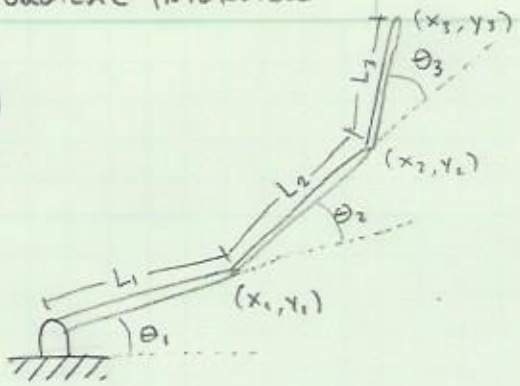
```
// Direct inverse of a 3x3 matrix m
void mat3x3_inv(double minv[3][3], const double m[3][3])
{
   double det;
   double invdet;

   det = m[0][0] * (m[1][1] * m[2][2] - m[2][1] * m[1][2]) -
         m[0][1] * (m[1][0] * m[2][2] - m[1][2] * m[2][0]) +
         m[0][2] * (m[1][0] * m[2][1] - m[1][1] * m[2][0]);

   invdet = 1.0 / det;

   minv[0][0] = (m[1][1] * m[2][2] - m[2][1] * m[1][2]) * invdet;
   minv[0][1] = (m[0][2] * m[2][1] - m[0][1] * m[2][2]) * invdet;
   minv[0][2] = (m[0][1] * m[1][2] - m[0][2] * m[1][1]) * invdet;
   minv[1][0] = (m[1][2] * m[2][0] - m[1][0] * m[2][2]) * invdet;
   minv[1][1] = (m[0][0] * m[2][2] - m[0][2] * m[2][0]) * invdet;
   minv[1][2] = (m[1][0] * m[0][2] - m[0][0] * m[1][2]) * invdet;
   minv[2][0] = (m[1][0] * m[2][1] - m[2][0] * m[1][1]) * invdet;
   minv[2][1] = (m[2][0] * m[0][1] - m[0][0] * m[2][1]) * invdet;
   minv[2][2] = (m[0][0] * m[1][1] - m[1][0] * m[0][1]) * invdet;
}


/* Explicit 3x3 matrix * vector.  b = A x  where A is 3x3 */
void mat3x3_vec_mult(double b[3], const double A[3][3], const double x[3])
{
   const int N = 3;
   int       idx, jdx;

   for( idx = 0; idx < N; idx++ )
   {
      b[idx] = 0.0;

      for( jdx = 0; jdx < N; jdx++)
      {
         b[idx] += A[idx][jdx] * x[jdx] ;
      }
   }
}
```

TAKE-HOME PROBLEMS FOR INTUITIVE SURGICAL INTERVIEW

IAN SYNDERGAARD

3/18/23

1)



$$\begin{bmatrix} x_3 \\ y_3 \end{bmatrix} = \begin{bmatrix} x_2 + L_3 \cos(\theta_1 + \theta_2 + \theta_3) \\ y_2 + L_3 \sin(\theta_1 + \theta_2 + \theta_3) \end{bmatrix}$$

$$= \begin{bmatrix} \overbrace{x_1 + L_2 \cos(\theta_1 + \theta_2)}^{x_2} + L_3 \cos(\theta_1 + \theta_2 + \theta_3) \\ \underbrace{y_1 + L_2 \sin(\theta_1 + \theta_2)}_{y_2} + L_3 \sin(\theta_1 + \theta_2 + \theta_3) \end{bmatrix}$$

$$= \begin{bmatrix} \overbrace{L_1 \cos(\theta_1)}^{x_1} + L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3) \\ \underbrace{L_1 \sin(\theta_1)}_{y_1} + L_2 \sin(\theta_1 + \theta_2) + L_3 \sin(\theta_1 + \theta_2 + \theta_3) \end{bmatrix}$$

$$J = \begin{bmatrix} \frac{\partial x_3}{\partial \theta_1} & \frac{\partial x_3}{\partial \theta_2} & \frac{\partial x_3}{\partial \theta_3} \\ \frac{\partial y_3}{\partial \theta_1} & \frac{\partial y_3}{\partial \theta_2} & \frac{\partial y_3}{\partial \theta_3} \end{bmatrix}$$

$$J = \begin{bmatrix} -L_1 \sin(\theta_1) - L_2 \sin(\theta_1+\theta_2) - L_3 \sin(\theta_1+\theta_2+\theta_3) & \cdots \\ L_1 \cos(\theta_1) + L_2 \cos(\theta_1+\theta_2) + L_3 \cos(\theta_1+\theta_2+\theta_3) & \cdots \end{bmatrix}$$

SECOND COLUMN:

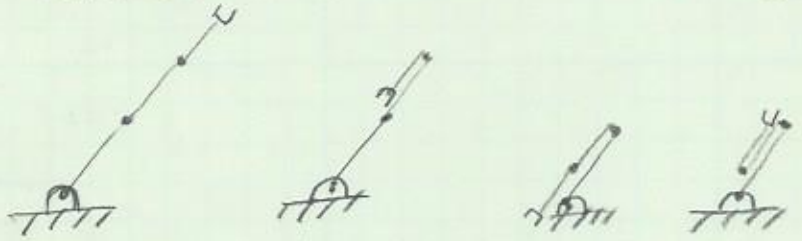
$$\begin{matrix} \cdots & -L_2 \sin(\theta_1+\theta_2) - L_3 \sin(\theta_1+\theta_2+\theta_3) & \cdots \\ \cdots & L_2 \cos(\theta_1+\theta_2) + L_3 \cos(\theta_1+\theta_2+\theta_3) & \cdots \end{matrix}$$

THIRD COLUMN:

$$\begin{matrix} \cdots & -L_3 \sin(\theta_1+\theta_2+\theta_3) \\ \cdots & L_3 \cos(\theta_1+\theta_2+\theta_3) \end{matrix}$$

2) THE JACOBIAN CAN HAVE UP TO RANK 2 (BECAUSE THE END EFFECTOR MUST LIE IN THE PLANE OF THE LINKAGES). THE JACOBIAN WILL BECOME RANK 1 WHEN TWO OF ITS LINKAGES ALIGN. THIS OCCURS WHEN θ_2 OR $\theta_3 = 180°$ AND WHEN $\theta_2 = \theta_3 = 0$. θ_1 DOES NOT AFFECT THE RANK OF J.

THE FOLLOWING ARE DRAWINGS OF THE POSES IN WHICH RANK(J) = 1 (AND HOLD FOR ANY θ_1):



IN POSES FOR WHICH THE JACOBIAN LOSES RANK, THERE EXISTS A DIRECTION IN WHICH THE END EFFECTOR CANNOT BE MOVED (AT LEAST INSTANTANEOUSLY - TO MOVE IN THAT DIRECTION, THE ROBOT MUST ENTER A DIFFERENT POSE). IF A MOVE IN THIS DIRECTION IS COMMANDED, THE JOINT VELOCITIES MUST BE (AT LEAST MOMENTARILY) INFINITE.

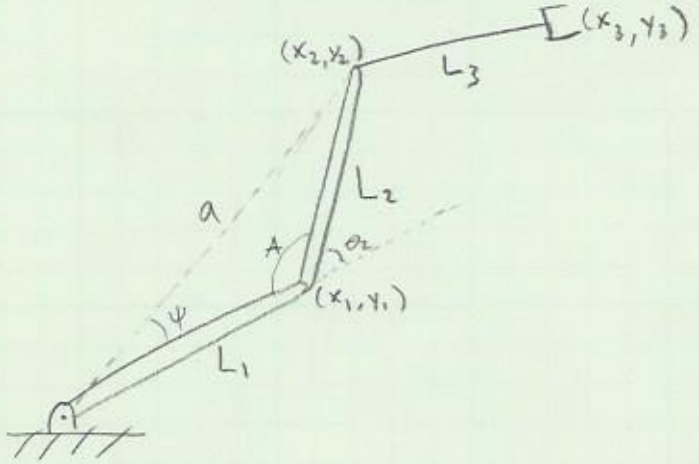
IF ORIENTATION IS ALSO INCLUDED IN THE JACOBIAN, THEN A ROW OF 1's WOULD BE ADDED TO THE JACOBIAN.

(BECAUSE ORIENTATION, $\phi = \theta_1 + \theta_2 + \theta_3$, THUS $\frac{\partial \phi}{\partial \theta_1} = \frac{\partial \phi}{\partial \theta_2} = \frac{\partial \phi}{\partial \theta_3} = 1$)

THIS ROW OF 1's DOES NOT CHANGE THE LINEAR INDEPENDENCE OF THE COLUMNS OF J, AND THUS DOES NOT CHANGE THE SINGULARITY CONDITIONS IN ANY WAY.

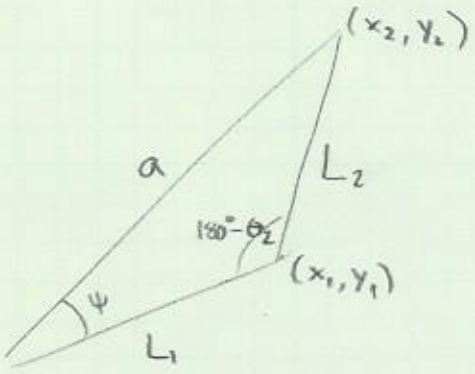
ONE WAY TO THINK ABOUT THIS CONCEPTUALLY IS THAT THE ORIENTATION OF THE END EFFECTOR IS EQUALLY CONTROLLABLE IN ALL ORIENTATIONS (AND THUS NEVER CAUSES LOSS OF RANK), WHILE THE SINGULARITY CONDITIONS FOR THE POSITION OF THE END EFFECTOR ARE NOT AFFECTED BY INCLUDING THIS TERM.

3)



I WILL DENOTE THE ORIENTATION OF THE END EFFECTOR BY ϕ

$$x_2 = x_3 - L_3 \cos(\theta_1 + \theta_2 + \theta_3) = x_3 - L_3 \cos(\phi)$$
$$y_2 = y_3 - L_3 \sin(\theta_1 + \theta_2 + \theta_3) = x_3 - L_3 \sin(\phi)$$



BY THE PYTHAGOREAN THEOREM:

$$a = \sqrt{x_2^2 + y_2^2}$$

BY THE LAW OF COSINES:

$$a^2 = L_1^2 + L_2^2 - 2L_1L_2\cos(180° - \theta_2)$$

$$\theta_2 = 180° - \cos^{-1}\left(\frac{L_1^2 + L_2^2 - a^2}{2L_1L_2}\right)$$

BY THE LAW OF COSINES:

$$L_2^2 = L_1^2 + a^2 - 2L_1a\cos(\psi)$$

$$\psi = \cos^{-1}\left(\frac{L_1^2 + a^2 - L_2^2}{2L_1a}\right)$$

$$\theta_1 + \psi = \tan^{-1}\left(\frac{y_2}{x_2}\right)$$

$$\Rightarrow \theta_1 = \tan^{-1}\left(\frac{y_2}{x_2}\right) - \psi$$

$$\phi = \theta_1 + \theta_2 + \theta_3$$

$$\Rightarrow \theta_3 = \phi - \theta_1 - \theta_2$$