Assignment 2

Deadline: Friday 23 Feb 2018, 10:59 PM

In this assignment, we will solve problem for Locomotion and Kinematics of some types of mobile robots.

Grading

1. Locomotion: `4 pts`
   1.1 Relative Rotation Matrix: `1 pt`
   1.2 Homogeneous Transformation: `1 pt`
   1.3 Jacobians and Differential Kinematics: `2 pts`
2. Kinematics: `12 pts`
   2.1 The 90 Degree Swedish Wheel: `4 pts`
   2.2 The Robot Parameters: `3 pts`
   2.3 The Stacked Wheel Equations: `5 pts`
3. Differential-Drive Robot `4 pts`

> **Note:** Assignment content is credited to ETH Zürich.

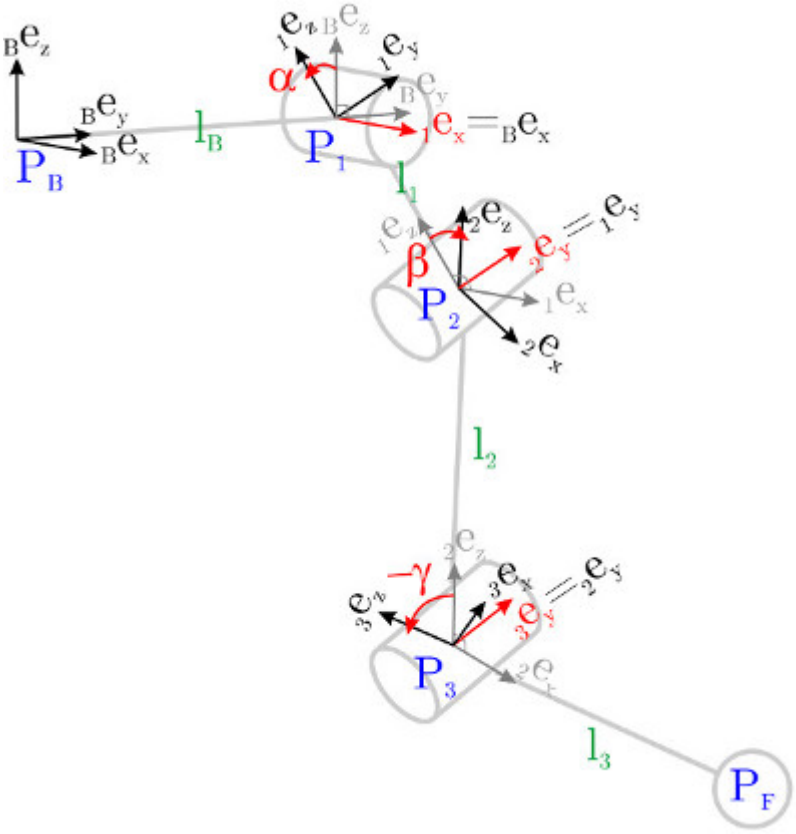
---

1. Locomotion

A Robotic Leg: Overview

In this exercise, we are going to analyze the kinematics of a single robot leg with a point foot. Starting with a set of generalized coordinates, we will elaborate relative rotations, translations, and homogeneous transformations. Subsequently, using foot point Jacobians, we will describe contact constraints and perform a trajectory tracking task using inverse kinematics.

The picture below describes a single leg that is attached to a body fixed frame B. This leg has three degrees of freedom consisting of relative rotations about α (alpha) around ${}_B\mathbf{e}_x$, about β (beta) around ${}_1\mathbf{e}_y$, and about γ (gamma) around ${}_2\mathbf{e}_y$. Hence, the generalized coordinates are given by $\mathbf{q} := [\alpha\ \beta\ \gamma]^T$.



1.1 Relative Rotation Matrix [1 pt]

Given the kinematic description of a single leg with three degrees of freedom $\mathbf{q} := [\alpha\ \beta\ \gamma]^T$ we determine the relative rotation matrices $\mathbf{R}_{AC}$ rotating a vector $\mathbf{r}$ from an arbitrary coordinate system C to A :

$$ {}_A\mathbf{r} = \mathbf{R}_{AC}\ {}_C\mathbf{r} $$

As a function of the generalized coordinates $alpha, beta, gamma$, what are the three relative rotation matrices?

```
syms alpha beta gamma real

% write down the rotation matrices using the symbolic parameters alpha, bet
R_B1 = ...;
R_12 = ...;
R_23 = ...;
```

1.2 Homogeneous Transformation [1 pt]

Given the relative rotation matrices from the previous problem and chosing unitary link lengths (), we determine the homogeneous transformation that tranforms the footpoint position represented in coordinate frame to coordinate frame :

$$\begin{pmatrix} {}_{B}\mathbf{r}_{BF} \\ 1 \end{pmatrix} = \mathbf{H}_{B3} \begin{pmatrix} {}_{3}\mathbf{r}_{3F} \\ 1 \end{pmatrix}$$

As a function of the generalized coordinates $alpha, beta, gamma$, what are the three relative position vectors and homogeneous transformation matrices?

```
% write down the 3x1 relative position vectors for link length l_i=1
r_B1_inB = ...;
r_12_in1 = ...;
r_23_in2 = ...;
r_3F_in3 = ...;

% write down the homogeneous transformation matrices
H_B1 = ...;
H_12 = ...;
H_23 = ...;

% create the cumulative transformation matrix
H_B3 = ...;

% find the foot point position vector
r_BF_inB = ...;
```

1.3 Jacobians and Differential Kinematics [2 pts]

Given the endeffector $\mathbf{r}_{BF}$ position as a function of generalized coordinates $\mathbf{q}$, we will determine the corresponding Jacobian $\mathbf{J}_{BF} = \frac{\partial \mathbf{r}_{BF}}{\partial \mathbf{q}}$ and velocity.

As a function of the generalized coordinates $\mathbf{q} = (alpha, beta, gamma)^T$, what is the foot point Jacobian and generalized velocity for a desired Cartesian motion?

```
syms alpha beta gamma real

q = [alpha;beta;gamma];
r_BF_inB = ...;

% determine the foot point Jacobian J_BF_inB=d(r_BF_inB)/dq
J_BF_inB = ...;

% what generalized velocity dq do you have to apply in a configuration q =
```

```
% to lift the foot in vertical direction with v = [0;0;-1m/s];

dq = ...;
```

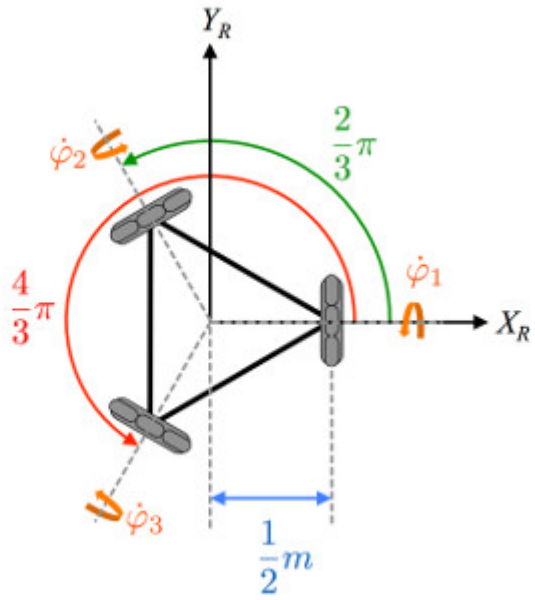
2. Kinematics

An Omnidirectional Robot : Overview

In this problem set, we will derive the equations of motion for the robot shown below. The content of this problem set closely follows the worked example so it is a good idea to go through that video segment first.

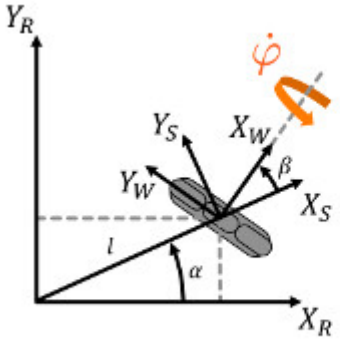
This robot has three non-steerable wheels. Each wheel is a “Swedish 90 degree wheel” as described in the first lecture segment of this section. The robot frame is placed at the center of the robot and each wheel is 0.5 meters from the robot center. Pay careful attention to the defined turning direction of each wheel. Making your equations match the defined turning directions will be key to success in this problem set.

$$\dot{\varphi} := \begin{bmatrix} \dot{\varphi}_1 \\ \dot{\varphi}_2 \\ \dot{\varphi}_3 \end{bmatrix}$$



2.1 The 90 Degree Swedish Wheel [4 pts]

Recall that the Swedish wheel has many small wheels around its circumference. In these questions we derive the equations for this wheel. The radius of the wheel is r.



2.1.1 Degrees of Freedom [1 pts]

How many degrees of freedom does this wheel have?

```
% number of degrees of freedom
dof = ...
```

2.1.2 The Wheel Equation [3 pts]

Please enter the wheel equation for the 90 degree Swedish wheel pictured above. Pay attention to the turning direction defined in the diagram. This is different than the turning direction defined in the worked example.

In the case of the Swedish wheel, the only actuator/encoder available is the one that spins the large wheel. Movement orthogonal to the large wheel is free due to the rolling of the small wheels. Therefore, the no-sliding-constraint matrix, $\mathbf{C}$, is empty and the wheel equation has the form

$$\mathbf{J}\dot{\xi}_R = r\dot{\varphi},$$

where $\dot{\xi}_R = [\dot{x}\ \dot{y}\ \dot{\theta}]^T$ is the robot's velocity expressed in the robot frame, r is the radius of the big wheel, and $\dot{\varphi}$ is the turning speed (radians per second) of the big wheel. The matrix $\mathbf{J}$ is 1×3 and has components

$$\mathbf{J} =: [j_1 \quad j_2 \quad j_3]$$

Write an expression for each component of in terms of α (alpha), $beta$ (beta) and ℓ (l). Use the trigonometric functions sin and cos. Use $*$ for multiplication.

```
j_1 =
j_2 =
j_3 =
```

2.2 The Robot Parameters [3 pts]

In this problem, we will write down the parameters for this robot. Pay careful attention to the wheel's turning direction defined on the diagram. This turning direction is different than the one used for a standard wheel in the worked exercise.
Given the constraint equations for this wheel, what are α, β, and ℓ for each wheel?

```
% There are several possible answers to this so the check
% is based on the stacked wheel equations.

% Wheel 1, the far right wheel
alpha1 =
beta1 =
ell1 =

% Wheel 1, the top left wheel
alpha2 =
beta2 =
ell2 =

% Wheel 3, the bottom left wheel
alpha3 =
beta3 =
ell3 =
```

2.3 The Stacked Wheel Equations [5 pts]

In this problem, we will derive the stacked wheel equation for this robot. The radius of each wheel is 0.1, and the wheels are numbered counter-clockwise starting from the wheel on the far right (wheel 1).

Given the constraint equations for this wheel, and the parameters from the last question, what are and for this robot such that

$$\mathbf{J}\dot{\xi}_R = \mathbf{R}\dot{\varphi},$$

where $\mathbf{J}_i$ is the 1×3 rolling constraint matrix for wheel i, φ_i is the rotation speed of wheel i,

$$\mathbf{J} := \begin{bmatrix} \mathbf{J}_1 \\ \mathbf{J}_2 \\ \mathbf{J}_3 \end{bmatrix}, \text{ and } \dot{\varphi} := [\dot{\varphi}_1 \;\; \dot{\varphi}_2 \;\; \dot{\varphi}_3]^T.$$

After this is complete, compute the forward differential kinematics matrix, $\mathbf{F}$, such that

$$\dot{\xi}_R = \mathbf{F}\dot{\varphi}.$$

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Step 1: paste values for alpha, beta, and ell from   %
%         the answer to the last question               %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Wheel 1, the far right wheel
alpha1=
beta1=
ell1=

% Wheel 2, the top left wheel
alpha2=
beta2=
ell2=

% Wheel 3, the bottom left wheel
alpha3=
beta3=
ell3=

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Step 2: derive the matrices J and R                   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
J =
R =

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Step 3: Compute the forward differential              %
%         kinematics matrix, F                          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
F =
```

2.4 Driving the Omnidirectional Robot [0 pts]

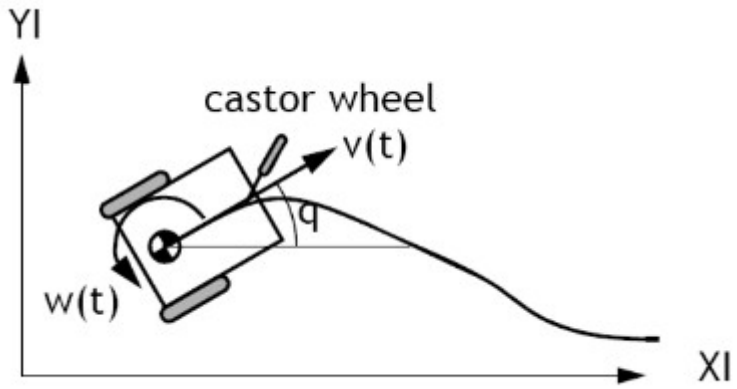
This is not a problem. Here we will simulate the forward differential kinematics equations from the previous questions.

```
%% Try changing the wheel speeds to see what motions the robot does.
numSeconds = 10;
% The speed of the first wheel (rad/s)
phi1 =
% The speed of the second wheel (rad/s)
phi2 =
% The speed of the third wheel (rad/s)
phi3 =
```

4. Differential-Drive Robot [4 pts]

The last section of this assignment is to try-out the kinematics equations for the differential-drive robot in real-life scenario. For this, we will use the LEGO EV3 model provided in V-Rep simulator.

One experiment on motion control will be done on the robot. First, you have to test your odometry model running an open-loop motion controller (line and circle segments) and verify if the position is estimated correctly. Second, the feed-back controller presented in the course will be used for more precise motion. The controller can be implemented in MATLAB with a fixed sampling time Δt.



From MATLAB, you will have direct access to the Lego model to read the encoder values and set speed of the wheels:

1. Establish a way to generate the line and circle segments of a trajectory.
2. Setup all the required relations for feedback control of the Lego model.
3. How will you implement it in MATLAB?

---

3. Handing-In

You are required to write MATLAB script to control and move the EV3 through the maze (in the V-Rep scene) without using any feedback from the sensors of the EV3 robot. You will only use the motors (aka actuators). Please write a script to move the EV3 forward, then turn right, move forward, then turn left, then move forward till it gets out of the maze. Please bear in mind that the path of the EV3 is hard-coded. Hence, trial and error is enough for this part of the assignment.

Write your code for the two aforementioned tasks in two separate MATLAB script files. For example: `task1.m` and `task2.m`. Then, compress them into one `.zip` file, for example `assignment_2_solution.zip`. Then, upload it to Blackboard.