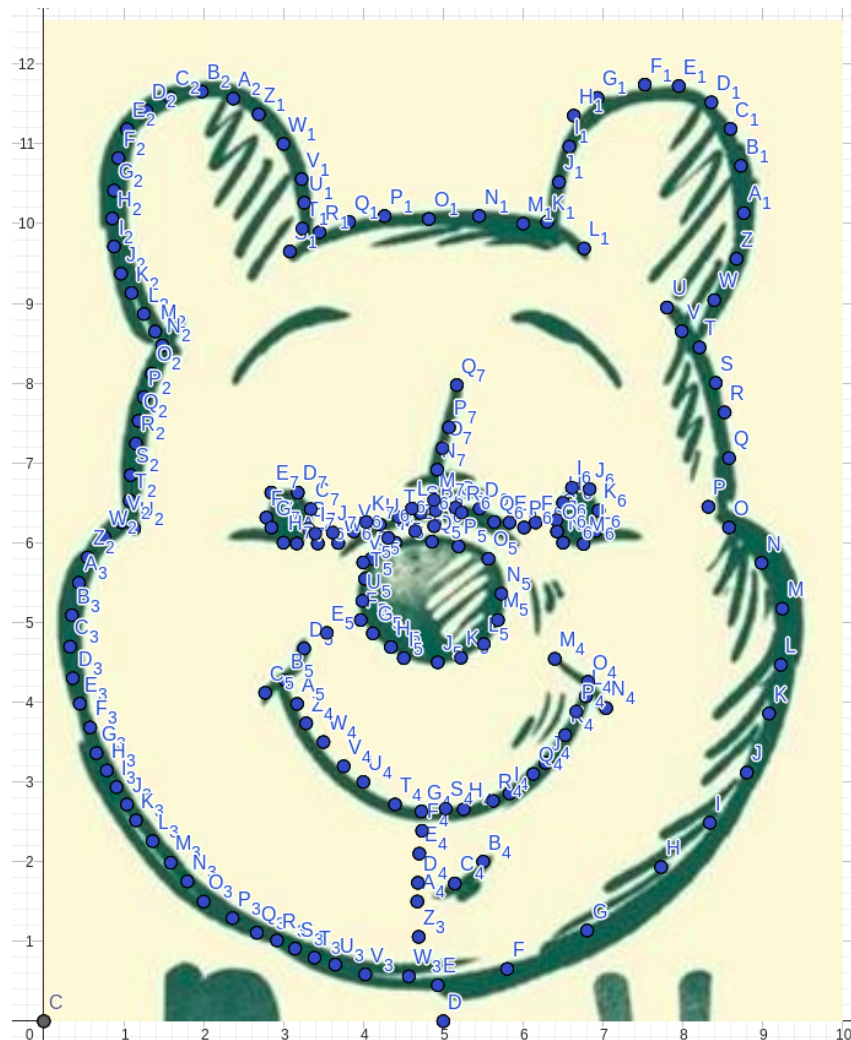


Evaluacion 7.2 (Planeación de trayectorias)

Jonathan Josafat Vázquez Suárez - A01734225

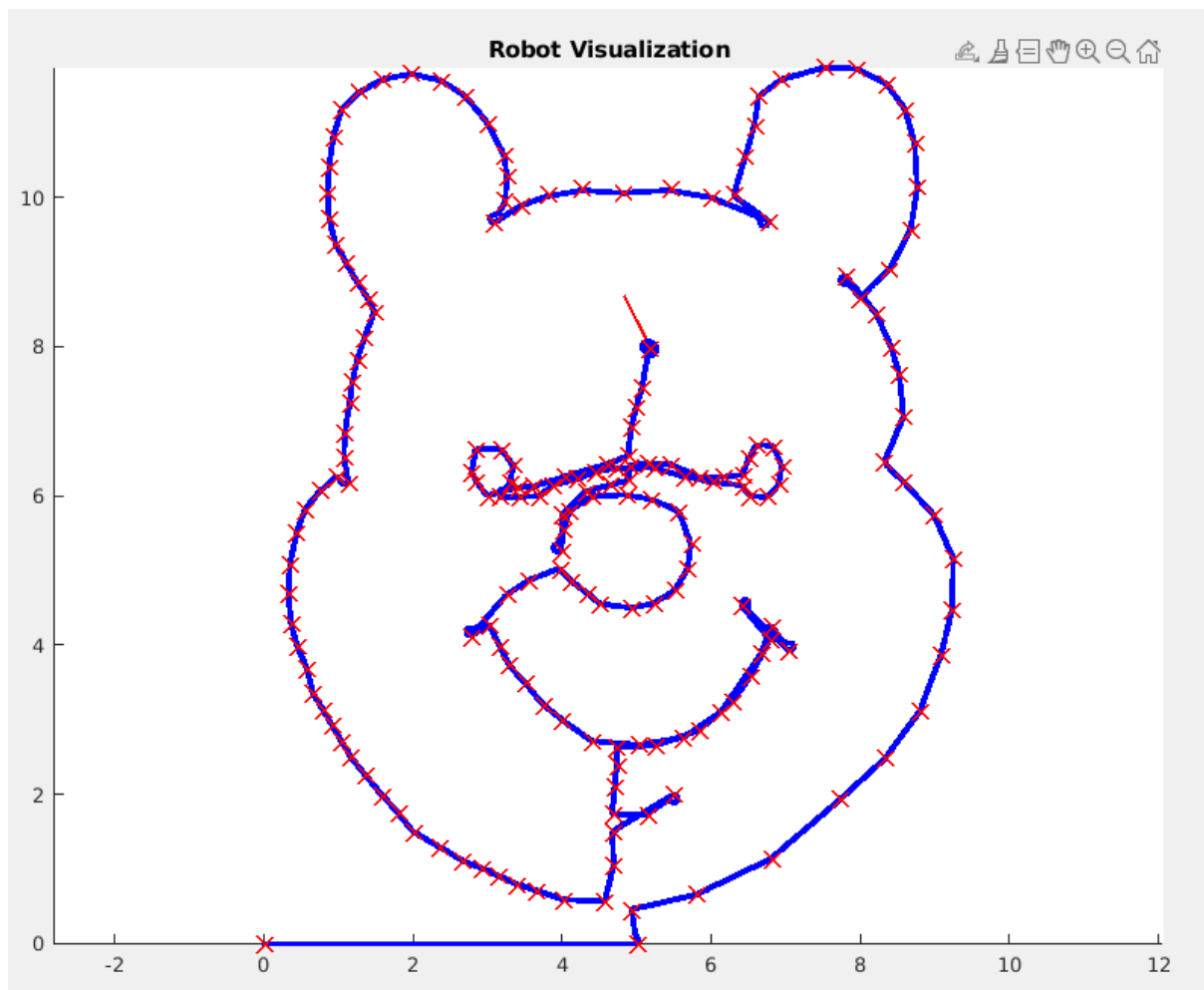
Generación del recorrido

Para la generación de la trayectoria se generaron los puntos pensando en el recorrido que pueda tener el robot, es decir, en orden comenzando desde el origen 0,0 hasta el punto final deseado que es **Q7**, la cercanía entre los puntos influirá en el recorrido final, pero la precisión del movimiento es mayormente afectada por sus velocidades angulares y lineales, tener velocidades grandes hará que el robot pierda precisión en el movimiento o deba hacer correcciones con rotaciones o oscilatorias, es por eso, que estimando una velocidad idónea, establecer los puntos en orden de trayectoria de la misma forma que el robot lo haría da un bonus en el recorrido, hablando de puntos donde el giro sea menor a los 90° es importante replantear y no existe una segunda posibilidad, es decir, intentar dejar un recorrido lo más suave posible comprendiendo el movimiento que lleva el robot.



Control de trayectoria

Definida la trayectoria y el orden en que funcionarán los waypoints, es cuestión de seleccionar el script a utilizar, para este caso el mejor de ellos es implementado por *MathWorks Inc.* llamado Pure Pursuit Algorithm, el cual utiliza las coordenadas y ajustando solo velocidad lineal y angular junto a un LookaheadDistance podemos tener trayectorias curvas o las mejor ajustadas, en este caso el parámetro que más debemos tener en cuenta es LookaheadDistance el cual es un rango de error para seguir la trayectoria, este resultado ya dependerá de lo deseado. Las velocidades angulares y lineales dependiendo de la distancia de los puntos y la precisión que se busca, deberán tener una velocidad específica, tener velocidades lineales grandes en puntos cercanos, causará que se pase del punto muy fácilmente, tener una velocidad angular muy rápida puede causar oscilaciones no deseadas y una muy lenta que el giro no se complete lo que causa desfases en la trayectoria. Para este caso, decidimos usar velocidades lineales bajas y considerablemente una mayor para angular, esto hace que el ajuste angular sea más rápido y al tener puntos cercanos nuestra trayectoria se define como curva con la misma suavidad con la que fueron seleccionados los puntos.



```

% Pure Pursuit algorithm
%
% Copyright 2018-2019 The MathWorks, Inc.
%% Define Vehicle
R = 0.1; % Wheel radius [m]
L = 0.5; % Wheelbase [m]
dd = DifferentialDrive(R,L);
%% Simulation parameters
sampleTime = 0.1; % Sample time [s]
tVec = 0:sampleTime:6000; % Time array
initPose = [0;0;0]; % Initial pose (x y theta)
pose = zeros(3,numel(tVec)); % Pose matrix
pose(:,1) = initPose;
% Define waypoints
waypoints = [0, 0; 5, 0; 4.9312224395809, 0.4517315410275; 5.801209151112,
0.655345452237; 6.8007683515946, 1.1366146969138; 7.7262861298192,
1.9325599861869; 8.3371278634475, 2.4878706531217; 8.7998867525598,
3.1172227423145; 9.0775420860272, 3.8576369648942; 9.2256249305431,
4.4684786985224; 9.2441352861076, 5.1718722099731; 8.9849903082047,
5.7456932324724; 8.5777624857859, 6.1899417660202; 8.318617507883,
6.4490867439231; 8.5777624857859, 7.0599284775514; 8.5222314190924,
7.6337495000506; 8.4111692857055, 8.0039566113405; 8.207555374496,
8.4482051448883; 7.8003275520772, 8.9479847451296; 7.9854311077221,
8.6518190560977; 8.392658930141, 9.0405365229521; 8.6703142636083,
9.5588264787578; 8.7628660414308, 10.1326475012571; 8.7258453303018,
10.7249788793209; 8.5962728413504, 11.1877377684332; 8.355638219012,
11.5209241685941; 7.9484103965931, 11.7245380798035; 7.5226722186098,
11.743048435368; 6.9303408405461, 11.5764552352875; 6.6341751515142,
11.3543309685136; 6.5786440848207, 10.9656135016593; 6.4490715958692,
10.5213649681115; 6.3009887513533, 10.0215853678702; 6.7637476404656,
9.6883989677093; 6, 10; 5.4495123953866, 10.0956267901281; 4.8201603061939,
10.0586060789991; 4.2648496392591, 10.0956267901281; 3.8206011057113,
10.0215853678702; 3.4503939944215, 9.8920128789187; 3.0801868831316,
9.6513782565803; 3.2340021301762, 9.9381144278378; 3.2609004223518,
10.2617795138511; 3.2279627617822, 10.5582184589778; 3, 11; 2.6899809724783,
11.3706807530285; 2.3715835869719, 11.5683067164463; 1.9763316601364,
11.6561404779653; 1.5810797333009, 11.5792859366362; 1.2846407881743,
11.414597633788; 1.043097943997, 11.1840340098006; 0.9333057420982,
10.8217197435348; 0.8784096411489, 10.4154885965094; 0.8564512007691,
10.0641535504334; 0.8784096411489, 9.7128185043574; 0.9662434026679,
9.3724626784712; 1.0979940449464, 9.130919834294; 1.2517031276046,
8.867418549737; 1.394432990073, 8.6478341459395; 1.482266751592,
8.4721666229015; 1.3505161093135, 8.1208315768254; 1.2517031276046,
7.8243926316988; 1.1858278064654, 7.5279536865722; 1.1528901458958,
7.2424939616354; 1.0870148247565, 6.8472420347999; 1.0760356045666,
6.5288446492935; 1.130931705516, 6.1775096032175; 0.9772226228577,
6.2543641445466; 0.7576382190602, 6.0677174013188; 0.5490330354526,
5.8151953369516; 0.4392408335539, 5.4967979514453; 0.3514070720349,
5.0905668044199; 0.3294486316551, 4.6953148775844; 0.3623862922247,
4.3000629507488; 0.4502200537437, 3.9816655652425; 0.5803788035201,
3.6827599313206; 0.6586144116436, 3.3611246534795; 0.7890070918494,

```

```
3.1438035198031; 0.9107069267081, 2.9351752314738; 1.041099606914,  
2.7178540977975; 1.1541065964257, 2.5179186548152; 1.362734884755,  
2.2571332944035; 1.5887488637784, 1.9876550886448; 1.7973771521078,  
1.7529482642744; 2, 1.5; 2.3624120996663, 1.2922274608804; 2.6666616868133,  
1.1096777085923; 2.9187542018779, 1.0140564097747; 3.1447681809013,  
0.90974226561; 3.3881678506188, 0.7967352760983; 3.6489532110305,  
0.7098068226278; 4.0227455609538, 0.588106987769; 4.5703948178183,  
0.5620284517278; 4.6920946526771, 1.05752063651; 4.674708961983,  
1.5008557492098; 5.5, 2; 5.1441226107239, 1.7268697282332; 4.68340180733,  
1.7355625735802; 4.7007874980241, 2.1006620781566; 4.7355588794123,  
2.3875259746094; 4.7268660340653, 2.6309256443269; 5.2571296002356,  
2.6570041803681; 5.8308573931413, 2.8569396233503; 6.2828853511881,  
3.2394248186208; 6.5262850209057, 3.587138632503; 6.7870703813173,  
4.073937971938; 6.3958923406998, 4.543351620679; 7.0391628963819,  
3.9261596010381; 6.8131489173585, 4.2564877242262; 6.6653705464585,  
3.8826953743028; 6.1264141349411, 3.1003392930679; 5.622229104812,  
2.7613183245327; 5.0311156212122, 2.6656970257151; 4.3965379108772,  
2.7178540977975; 4, 3; 3.7532673551951, 3.1959605918855; 3.5, 3.5;  
3.2838537064542, 3.7349170034029; 3.1708467169424, 3.9783166731204;  
3.0230683460425, 4.2738734149203; 2.7709758309779, 4.1174021986733;  
3.257775170413, 4.6737443008848; 3.5422082955703, 4.8677373207071;  
3.9669073519054, 5.0299819040262; 4.1196081362056, 4.8629654211978;  
4.3438874131466, 4.69117703886; 4.5061319964657, 4.5527919530879;  
4.9308310528008, 4.5003010584846; 5.2219169228732, 4.5575638525972;  
5.5082308934361, 4.729352234935; 5.6847911752833, 5.0299819040262;  
5.7277382708678, 5.3592429701736; 5.5646061315529, 5.7982672369189;  
5.1912526717689, 5.9534815966044; 4.8598490389269, 6.0122113543233; 4.4, 6;  
4.1005571712762, 5.7982672369189; 4.0208525000863, 5.5465682752668;  
3.9872926385327, 5.2696994174494; 3.9988212793688, 5.7520825773859;  
4.3114640344686, 6.0584724773838; 4.6491182099765, 6.1460124488118;  
4.8867267038524, 6.2085409998317; 4.9054852691584, 6.4023795079936;  
5.1555994732383, 6.4398966386056; 5.4557365181341, 6.4273909284016;  
5.8309078242539, 6.2523109855457; 6.1560562895578, 6.2523109855457;  
6.4186762038417, 6.2835752610557; 6.5, 6.5; 6.6125147120036, 6.6900108426855;  
6.8313646405735, 6.6712522773795; 6.9439160324094, 6.4023795079936;  
6.9063989017974, 6.1460124488118; 6.7563303793495, 5.9834382161598; 6.5, 6;  
6.4249290589437, 6.1397595937098; 6.0122406222119, 6.1897824345257;  
5.637069316092, 6.2585638406477; 5.2306337344622, 6.3773680875856;  
4.7178996160985, 6.3711152324837; 4.4427739916106, 6.3210923916677;  
4.2051654977347, 6.2272995651377; 3.8800170324308, 6.1397595937098;  
3.6861785242689, 5.9959439263638; 3.4298114650871, 5.9896910712618;  
3.1671915508032, 5.9896910712618; 3.3110072181491, 6.1772767243217;  
3.3422714936591, 6.4211380732996; 3.1796972610072, 6.6274822916655;  
2.8482959406013, 6.6274822916655; 2.7857673895813, 6.3148395365657;  
2.8482959406013, 6.1897824345257; 3, 6; 3.3985471895771, 6.1147481733018;  
3.617397118147, 6.1272538835058; 4.0363384099808, 6.2585638406477;  
4.6074415381733, 6.427938187971; 4.8847562009386, 6.5357827790464;  
4.924688932392, 6.914942769673; 4.990037285099, 7.1845047245893;  
5.0717227259827, 7.4458981354173; 5.1697452550432, 7.9768535011615];
```

```
% Create visualizer
```

```
viz = Visualizer2D;
```

```

viz.hasWaypoints = true;
%% Pure Pursuit Controller
controller = controllerPurePursuit;
controller.Waypoints = waypoints;
controller.LookaheadDistance = 0.05;
controller.DesiredLinearVelocity = 0.1;
controller.MaxAngularVelocity = 2.0;
%% Simulation loop
close all
r = rateControl(1/sampleTime);
for idx = 2:numel(tVec)
    % Run the Pure Pursuit controller and convert output to wheel speeds
    [vRef,wRef] = controller(pose(:,idx-1));
    [wL,wR] = inverseKinematics(dd,vRef,wRef);

    % Compute the velocities
    [v,w] = forwardKinematics(dd,wL,wR);
    velB = [v;0;w]; % Body velocities [vx;vy;w]
    vel = bodyToWorld(velB,pose(:,idx-1)); % Convert from body to world

    % Perform forward discrete integration step
    pose(:,idx) = pose(:,idx-1) + vel*sampleTime;

    % Update visualization
    viz(pose(:,idx),waypoints)
    waitfor(r);
end

```