Adit Jain, 200038
June 24, 2022

SURGE '22 Mid-Term Progress Report

guided by Dr. Twinkle Tripathy

1. Introduction

One of the important aspects of autonomous trajectory generation is to plan one such that anyone targeting the bot from outside is unable to accurately locate it's position at any given point in time. That's why we propose a switching pattern of trajectories, that can successfully cover the entire terrain of interest without being able to be precisely located.

2. Problem Description

The kinematics of the bot can be given by:

$$\dot{x}(t) = v\ cos(\alpha(t))$$
$$\dot{y}(t) = v\ sin(\alpha(t))$$
$$\dot{\alpha}(t) = u(t)$$

where $(x(t), y(t))$ and $\alpha(t)$ denote the position coordinates and heading angle of the bot at any time, t. Assuming that the angular speed, denoted by $u(t)$ is a function of range, r, we can write the equations in polar coordinates as follows:

$$\dot{r} = -v\ cos(\alpha - \theta)$$
$$\dot{\theta} = -\frac{v}{r}\ sin(\alpha - \theta)$$
$$\dot{\alpha} = f(r)$$

Further, replacing θ by $(\phi - \alpha)$, we can get the simplified kinematics of the bot defined by the following equations:

$$\dot{r} = -v\ cos(\phi)$$
$$r\dot{\phi} = rf(r) + v\,sin(\phi)$$

The above equations give the relation between r and ϕ from which $dr/d\phi$ can be calculated and then rearranged to get:

$$(rf(r) + v\,sin(\phi))\,dr + vr\,cos(\phi)\,d\phi = 0$$

which, on integration gives,

$$\tilde{f}(r) + vr\,sin(\phi) = K$$

where K, a constant, can be calculated from the initial conditions of r_0 and ϕ_0. Moreover, we can also define another function, $g(r)$, termed as the *generating function* for $f(r)$ by:

$$g(r) = \tilde{f}(r) - K = -vr\,sin(\phi)$$
$$\Rightarrow f(r) = \frac{1}{r}\frac{d}{dr}g(r)$$

Since we are interested in movement in an annular region, we can define 2 constants, namely R_{min} and R_{max}, representing the inner and outer radii of the annular region. Hence, in the region, we will satisfy the condition of $|g(r)| < vr$.

Also, switching between two trajectories generated by two generating functions, $g_1(r)$ and $g_2(r)$ is possible if the solution of the two functions lies in (R_{min}, R_{max}).

3. MATLAB Implementation

Elementary code was written in MATLAB to randomly make a generating function on specifying $R_{min} = 20m$ and $R_{max} = 40m$, at a constant velocity, $v = 1m/s$ the results of which are as plotted:

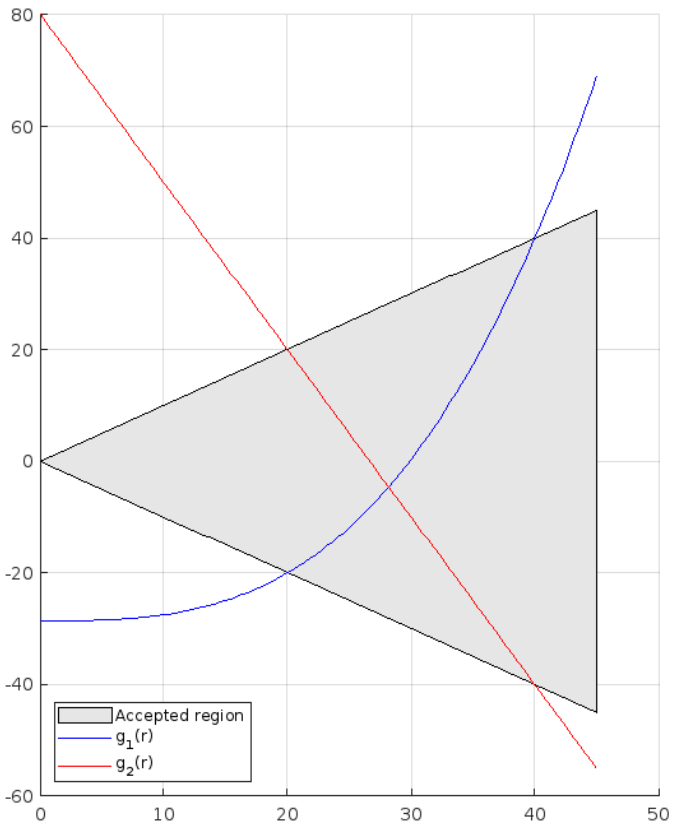

Figure 1: Generating Functions

Based on the two generating functions, the switching point, R_{switch}, was calculated. As is visible, R_{switch} was calculated at a range of $28.19m$. The initial conditions passed to the bot were as follows:

$$r_0 = 20m,\quad \phi_0 = \frac{3\pi}{2},\quad \alpha_0 = \frac{3\pi}{2},\quad \theta_0 = 0$$

Using this data, the trajectory was generated by solving the following simultaneous ODEs, represented by a state-space matrix:

$$\begin{bmatrix} \dot{r} \\ \dot{\phi} \\ \dot{\alpha} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -v\,cos(\phi) \\ f(r) + \frac{v}{r}\,sin(\phi) \\ f(r) \\ -\frac{v}{r}\,sin(\phi) \end{bmatrix}$$

where $f(r)$ represents the function $f_1(r)$ or $f_2(r)$ based on which generating function's, $g_1(r)$ or $g_2(r)$ respectively, trajectory the bot was following.

Initially, the switching was done the first time the bot crossed R_{switch}. Then switching was done every 6^{th} time the bot crossed the point R_{switch}. The following trajectory was generated when the ODEs were solved for a fixed time interval:

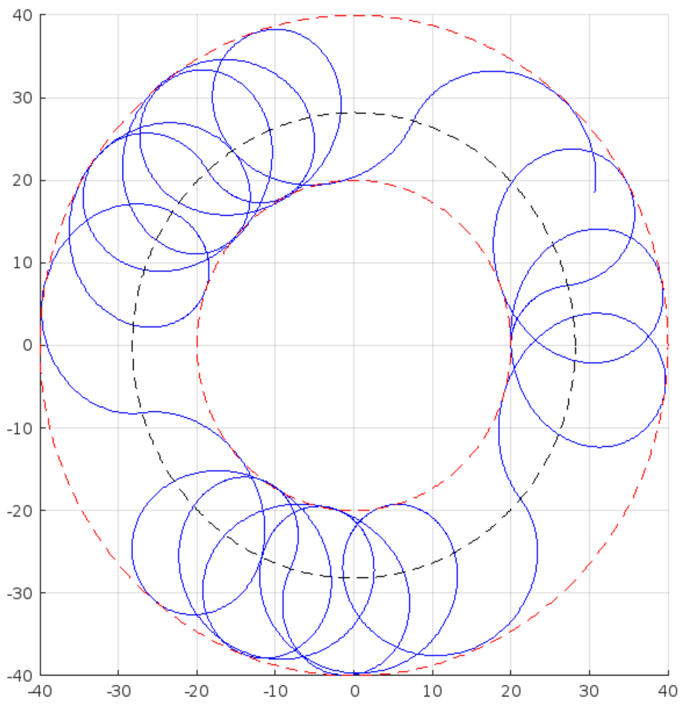

Figure 2: Switching Trajectories

4. Simulation and Testing

The obtained trajectory will now be tested on TurtleBot3 waffle bots, before being simulated on a system with the following specifications:

- Ubuntu 20.04
- ROS Noetic
- Gazebo 11
- GNU gcc-17
- Python 3.8.10

An overhead camera will be used for coarse estimation of the bot in real time, while onboard IMU sensors will be used for finer tracking.

5. Keywords

- Autonomous Trajectory generation
- Target avoidance
- Path planning