Bang-Bang Control (on the robot!!)

ROB 102: Introduction to AI & Programming

2021/09/15

Administrative

Project 1 is out! Due **October 4th, at 11:59 PM**.

https://robotics102.github.io/projects/a1.html

Project 0 will be due **October 4th, at 11:59 PM**.

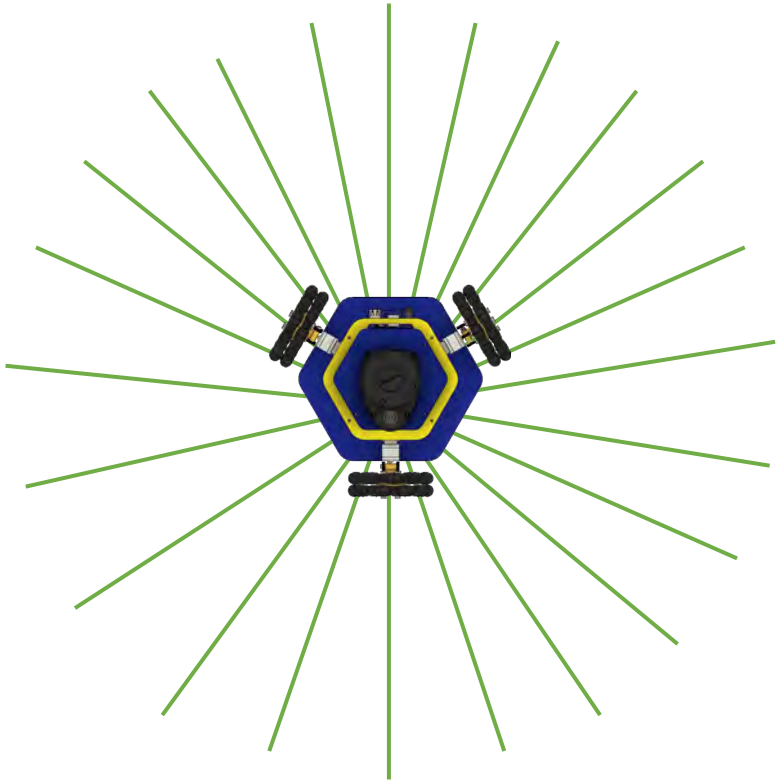
Friday's lab: Robot workflow (for Project 1)

Laser scan data

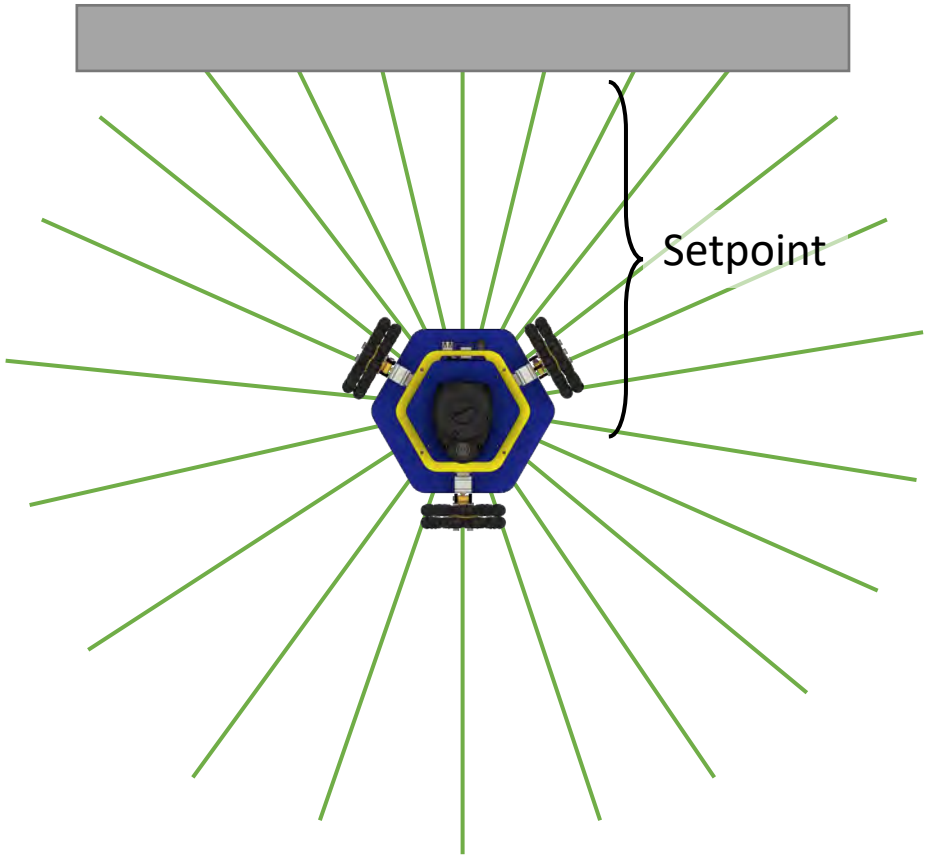
The Lidar sends out a series of rays.

Each scan is a list of rays with the following data:

- range (length in meters)
- angle (in radians)
- Intensity
- Time of scan



1D Control Problem

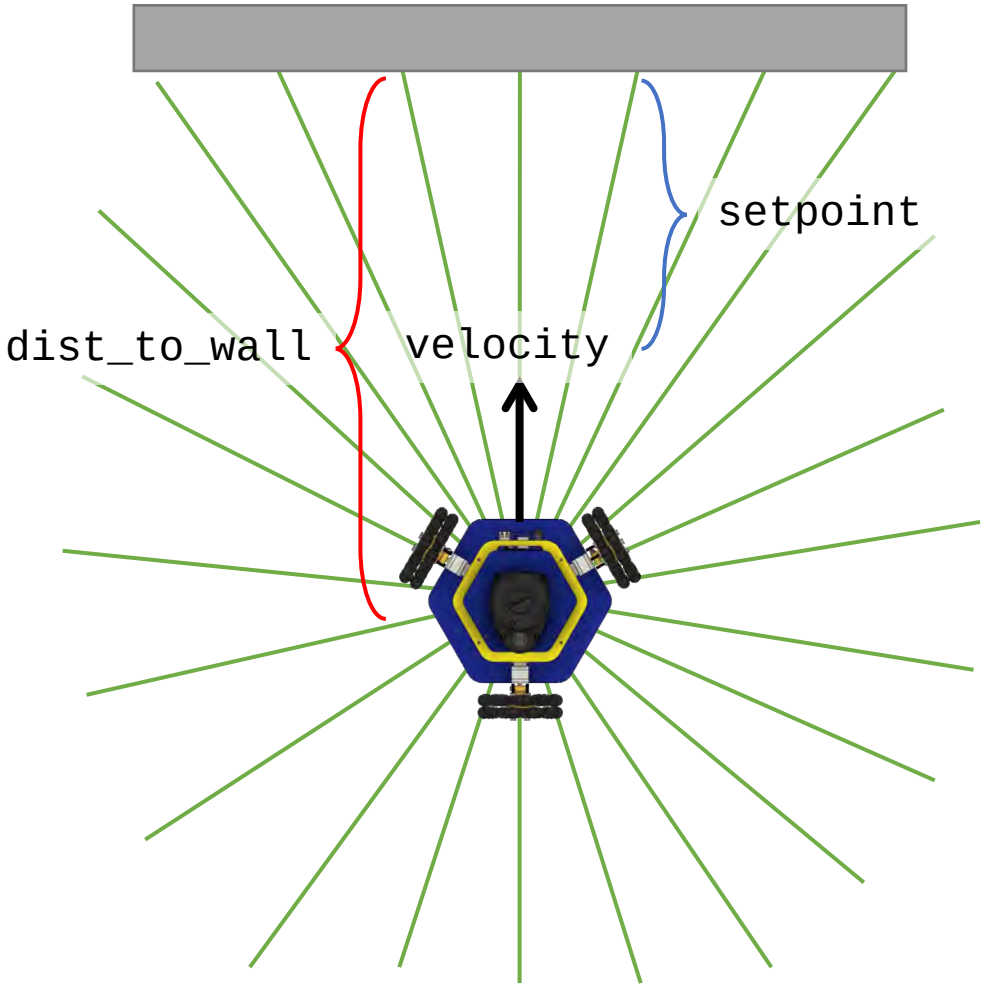


Goal: Write a controller so that the robot drives towards the wall and stops a certain distance from the wall.

The desired distance from the wall is called the **setpoint**.

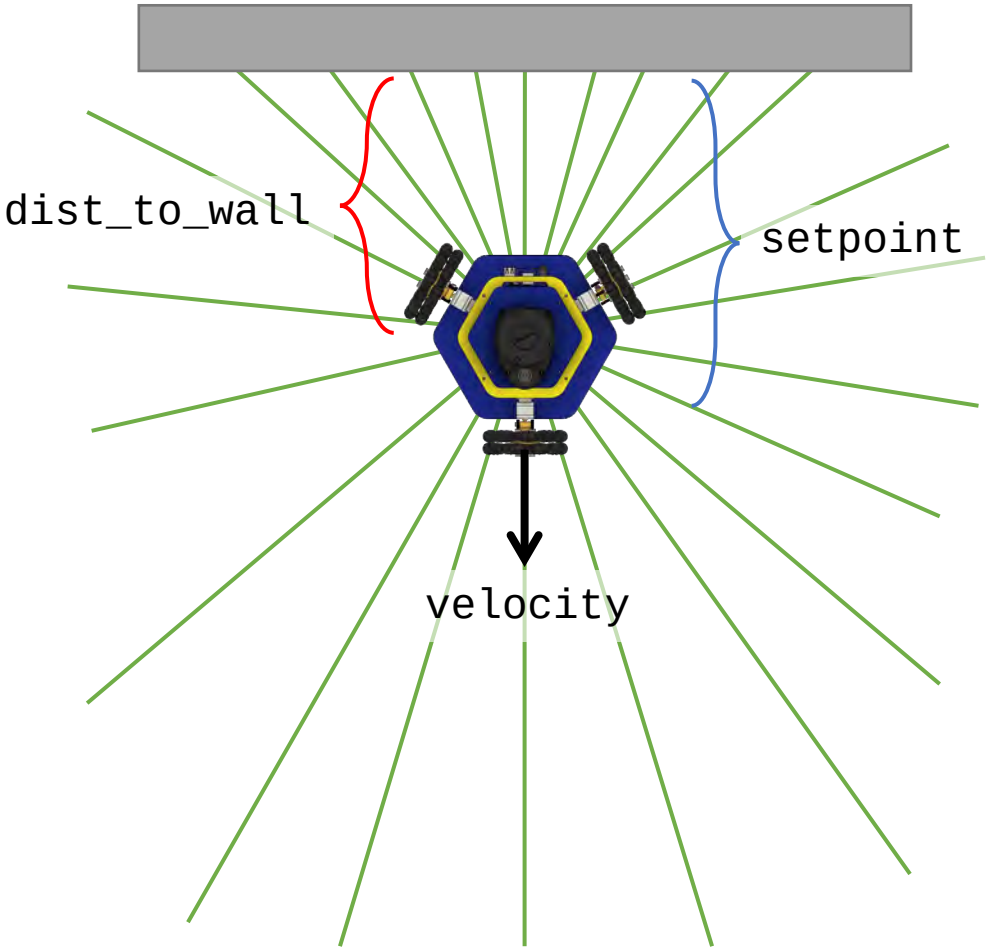
In Project 1, we will do this in 2D, so the robot drives along the wall instead of stopping!

Bang-Bang Control



If the robot is **too far** from the wall, drive **forward**.

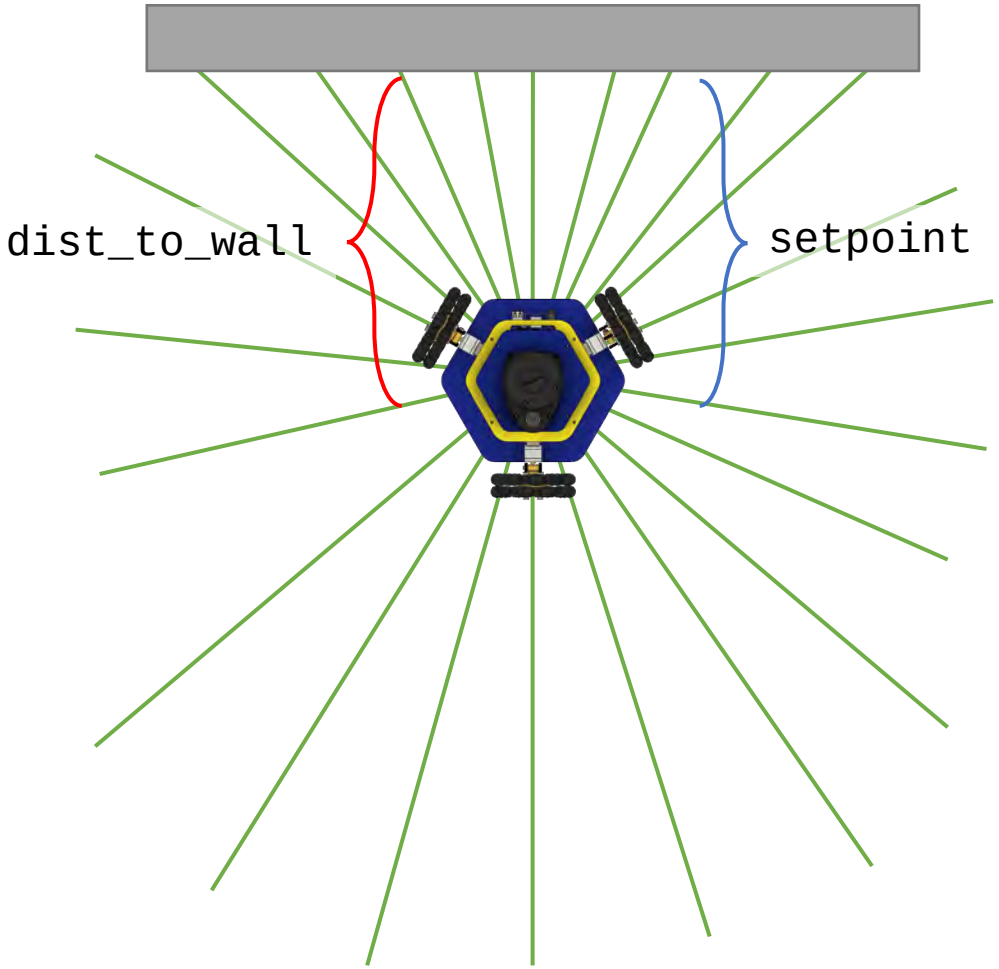
Bang-Bang Control



If the robot is **too far** from the wall, drive **forward**.

If the robot is **too close** to the wall, drive **backward**.

Bang-Bang Control



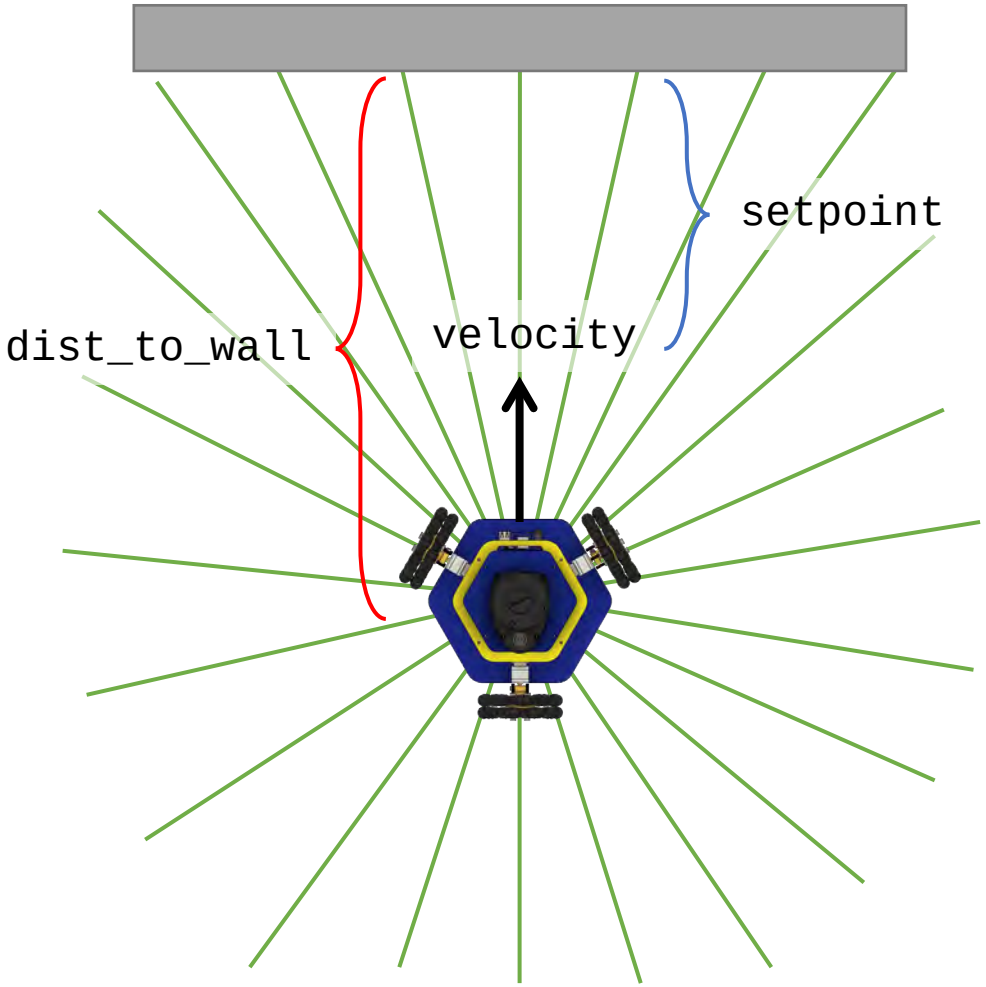
If the robot is **too far** from the wall, drive **forward**.

If the robot is **too close** to the wall, drive **backward**.

If the robot is within an allowable **margin** from the setpoint, **stop**.

Need to pick: velocity, margin.

P-Control



Apply a control signal **proportional to the error** between the current distance and the setpoint:

```
velocity = kp * error
```

Some constant

If the robot is within an allowable **margin** from the setpoint, **stop**.

Need to pick: velocity, margin.

Coding activity for today

```
float dt = 0.01;
float setpoint = 0.35;

while (true) {
    LidarScan scan = readLidarScan(drv);

    if (scan.good)
    {
        // Get the distance to the wall.
        float dist_to_wall = findFwdDist(scan);
        if (dist_to_wall < 0) continue;

        // Calculate the appropriate control signal.
        float vel = feedbackControl(dist_to_wall, setpoint);

        std::cout << "Setpoint: " << setpoint << " Current distance: " << dist_to_wall;
        std::cout << " Velocity command: " << vel << "\n";

        // Apply the control signal.
        drive(vel, 0, 0);
    }

    sleepFor(dt);

    if (ctrl_c_pressed) break;
}
```

Setpoint in meters

Loop forever

Read a scan

Get the distance to the wall
(this code is provided)

Calculate the control signal
(your code!!)

Send the velocity signal to
the robot

Coding activity for today

```
float dt = 0.01;
float setpoint = 0.35;

while (true) {
    LidarScan scan = readLidarScan(drv);

    if (scan.good)
    {
        // Get the distance to the wall.
        float dist_to_wall = findFwdDist(scan);
        if (dist_to_wall < 0) continue;

        // Calculate the appropriate control signal.
        float vel = feedbackControl(dist_to_wall, setpoint);

        std::cout << "Setpoint: " << setpoint << " Current distance: " << dist_to_wall;
        std::cout << " Velocity command: " << vel << "\n";

        // Apply the control signal.
        drive(vel, 0, 0);
    }

    sleepFor(dt);

    if (ctrl_c_pressed) break;
}
```

```
float feedbackControl(float dist_to_wall, float setpoint)
{
    Your code here!
}
```

Today:

1. Find your teammate!
2. Write a function which accepts the distance to the wall and the setpoint and returns the control signal using bang-bang control
3. Send your function to Jana on Slack
4. Test your code on the robot!
5. Repeat steps 2-4 with P-control.

```
float feedbackControl(float dist_to_wall, float setpoint)
{
    Your code here!
}
```