Emotional BWI Segway Robot

Sangjin Shin

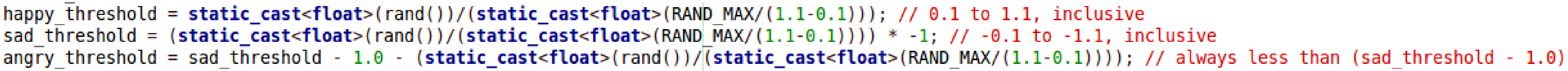
Introduction

- Current BWI Segway Robot is too boring!
- Make the robot more human-like.
- Make the interaction with the robot more interesting and unpredictable.
- Give the robot personality and emotions.
- The robot gives commands, not you.

Implementation

- Current Mood Level: double, default 0.0

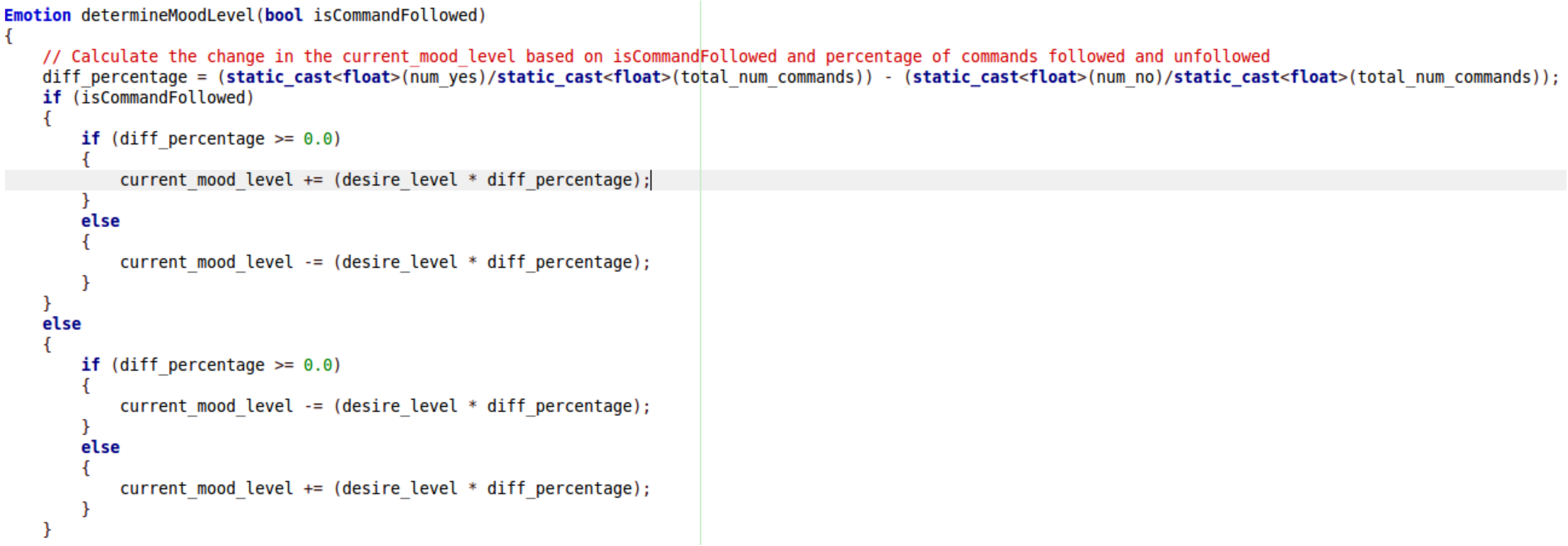
- Emotions: Neutral, Happy, Sad, Angry
  - Neutral Threshold: 0.0 <= current_mood_level < Happy Threshold
  - Happy Threshold: Randomly generated double 0.1 to 1.1, inclusive
  - Sad Threshold: Randomly generated double -0.1 to -1.1, inclusive
  - Angry Threshold: Randomly generated double 0.1 to -1.1 less than Sad Threshold - 1.0



```
happy_threshold = static_cast<float>(rand())/(static_cast<float>(RAND_MAX/(1.1-0.1))); // 0.1 to 1.1, inclusive
sad_threshold = (static_cast<float>(rand())/(static_cast<float>(RAND_MAX/(1.1-0.1)))) * -1; // -0.1 to -1.1, inclusive
angry_threshold = sad_threshold - 1.0 - (static_cast<float>(rand())/(static_cast<float>(RAND_MAX/(1.1-0.1)))); // always less than (sad_threshold - 1.0)
```

Implementation

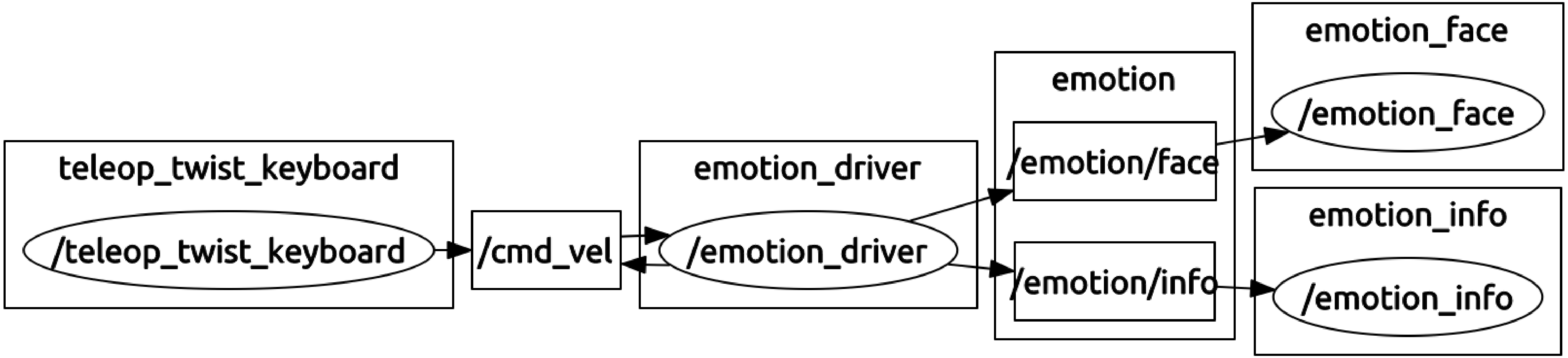
- Robot randomly gives one of four commands to the user:
  - Move forward
  - Move backward
  - Turn left
  - Turn right
- Each command is given a random weight value 0.0 to 1.0, inclusive
- Current Mood Level changes based on the difference of percentage of commands followed and percentage of commands not followed, and also the desire_level of each command.



```
Emotion determineMoodLevel(bool isCommandFollowed)
{
    // Calculate the change in the current_mood_level based on isCommandFollowed and percentage of commands followed and unfollowed
    diff_percentage = (static_cast<float>(num_yes)/static_cast<float>(total_num_commands)) - (static_cast<float>(num_no)/static_cast<float>(total_num_commands));
    if (isCommandFollowed)
    {
        if (diff_percentage >= 0.0)
        {
            current_mood_level += (desire_level * diff_percentage);
        }
        else
        {
            current_mood_level -= (desire_level * diff_percentage);
        }
    }
    else
    {
        if (diff_percentage >= 0.0)
        {
            current_mood_level -= (desire_level * diff_percentage);
        }
        else
        {
            current_mood_level += (desire_level * diff_percentage);
        }
    }
```

emotion_driver Package

- emotion_driver : main
- emotion_info: displays variable information
- emotion_face: displays the current mood of the robot in ASCII art face



Implementation

1. Robot gives movement command.
2. Wait until user input and linear and angular data from cmd_vel have been received.
3. Determine if user followed command.
4. Calculate change in current_mood_level and determine new emotion.
5. Publish variable data to emotion_info and publish ASCII art face to emotion_face.
6. Repeat.

Implementation

Pros

- Robot given a personality like a human.
- No way to know how the robot’s mood will change.
- Trust between robot and human.

Cons

- Random generator from C++ standard library.
- Mood-level-changing algorithm questionable.
- Robot unaware of surroundings and could be damaged with user error.

Demo

Future Improvements

- Better face
- LED Lights
- Robot chooses destination in 2D costmap.

Questions and Comments?