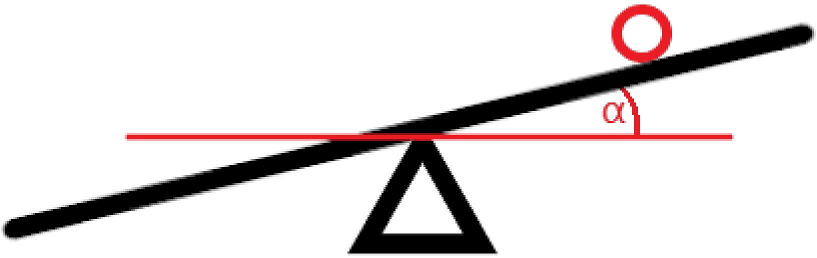
BALANCING A BALL

OBJECTIVES

You should implement a simulator for balancing a ball on a beam. The ball position can be controlled by controlling the angle α of the beam relative to the ground. The beam can rotate around its center point.

The only force acting on the ball is gravity ($9.81 m/s^2$), all other forces (friction, air drag,…) can be ignored. You can also ignore the rotational energy of the ball and assume that it only translates along the beam. The force acting on the ball and the resulting acceleration in our case are therefore:

$$F = m \cdot g \cdot \sin(\alpha)$$
$$a = \frac{F}{m} = g \cdot \sin{(\alpha)}$$



The following limitations apply:

- The angle α lies between -22.5° and 22.5°, so $\frac{-\pi}{8} \leq \alpha \leq \frac{\pi}{8}$
- The angle α may change by no more than 5.625° per second (which is $\frac{\pi}{32}$)
- The length of the beam is 2m

You should control the beam in such a way that the ball is as close to the beam's center as possible. It should not drop off the beam. The ball's initial position is at the center of the beam.

To make things a bit more challenging, it should be possible to push the ball to either side. A push results in a change in velocity.

You should implement an application that tries to find the optimal parameters for controlling the beam's angle. "Optimal" in this case means that the integrated deviation of the ball's position from the beam's center over 10s is minimized for the following scenario:

- The ball lies in the center of the beam
- The ball is pushed to the right, its velocity changes by 0.5m/s
- After 5s, the ball is pushed to the left, its velocity changes by -0.5m/s

You should also implement an application optimized for real-time-constraints that can be run on a Raspberry Pi using the balance and ball simulation you already implemented. The “user interface” of this application should consist of 3 LEDs and two buttons:

- One LED should light up in case the ball is centered (ball position is less than 10cm away from the beam’s center)
- One LED should light up in case the ball is more than 10cm to the left of the beam’s center
- One LED should light up in case the ball is more than 10cm to the right of the beam’s center
- All three LEDs should light up in case the ball falls off the beam
- When the first button is pressed, the ball’s velocity changes by 1m/s for each second the button is pressed (button down for 0.5s => ball’s velocity changes by 0.5m/s)
- When the other button is pressed, the ball’s velocity changes the same way as when pushing the first button, but to the other direction
- When pushing both buttons at the same time, the ball is reset to the beam’s center, its velocity is reset to 0m/s