SELF-BALANCING BOT USING CONCEPT OF INVERTED PENDULUM

A project by

Sayan Saha *(U14EC030)*
Abhirup Panja *(U14EC033)*
Jagadeesh Kaviti *(U14EC044)*
M.V.Nikil *(U14EC043)*

Under the supervision of

Prof. Pinalkumar J. Engineer

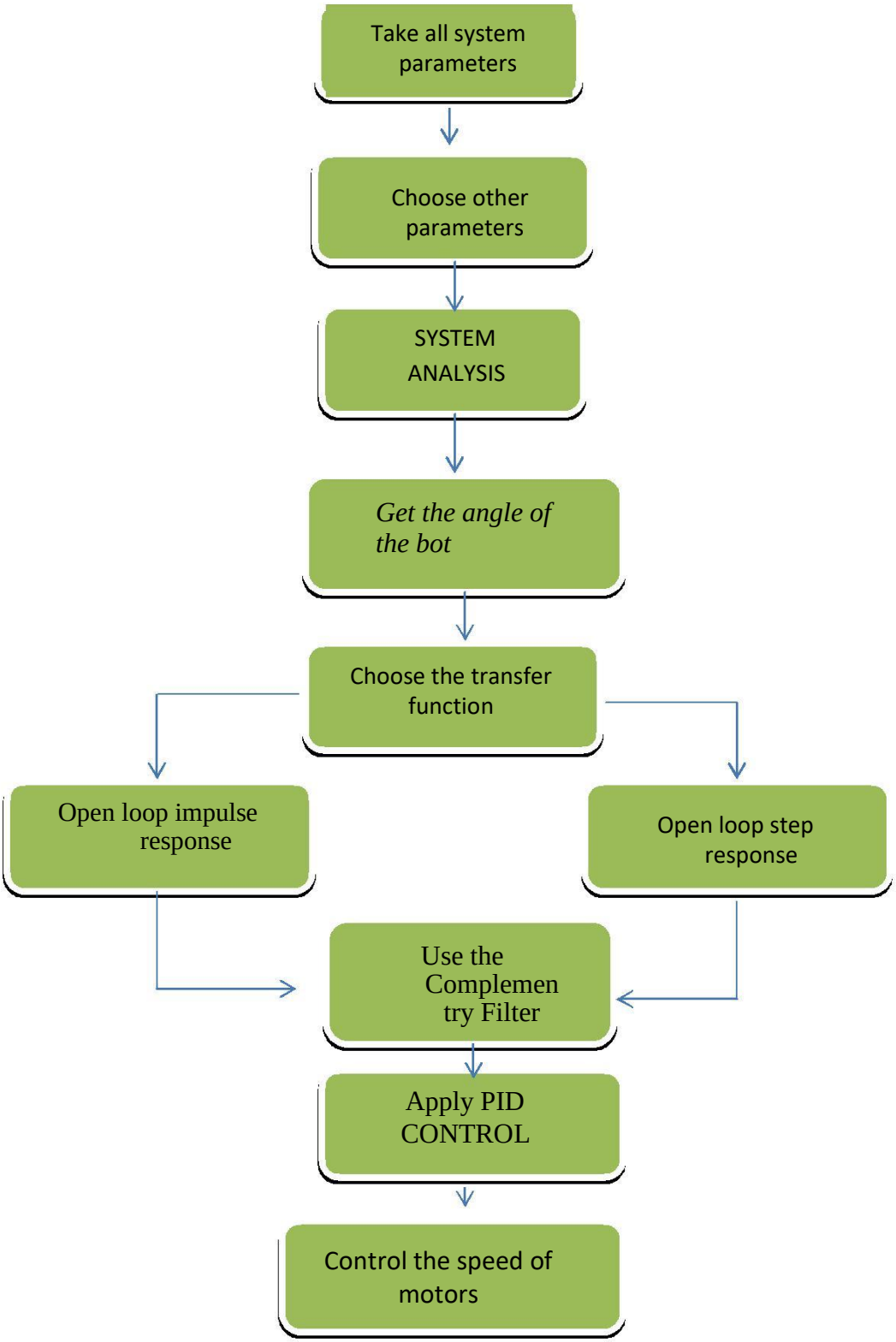
Department of Electronics and Communication Engineering
S.V. National Institute of Technology, Surat
Gujarat - 395007, India

ABSTRACT

Self-balancing robot is based on the principle of Inverted pendulum, which is a two wheel vehicle balances itself up in the vertical position with reference to the ground. It consist both hardware and software implementation. Mechanical model based on the state space design of the cart, pendulum system. To find its stable inverted position, I used a generic feedback controller (i.e. PID controller). According to the situation we have to control both angel of pendulum and position of cart. Mechanical design consist of two dc motor, one Arduino Uno microcontroller, IMU (inertial mass unit) sensor (MPU6050) and motor driver (Adafruit motor-shield) as a basic need. IMU sensor which consists of accelerometer and gyroscope gives the reference acceleration and angle with respect to ground (vertical). These parameters are taken as the system parameter and determine the external force needed to balance the robot up.

It will be prevented from falling by giving acceleration to the wheels according to its inclination from the vertical. If the bot gets tilts by an angle, than in the frame of the wheels; the centre of mass of the bot will experience a pseudo force which will apply a torque opposite to the direction of tilt.

Flow Chart


Take all system parameters
Choose other parameters
SYSTEM ANALYSIS
Get the angle of the bot
Choose the transfer function
Open loop impulse response
Open loop step response
Use the Complementry Filter
Apply PID CONTROL
Control the speed of motors

Working

An inverted pendulum balance-bot is inherently unstable. Conveniently, the high center of gravity creates a large moment of inertia that slows the rate at which it will fall. We can leverage this slow fall by continually moving the wheels under the vehicle as it falls. If it leans forward, the wheels roll forward to counteract the fall.

A simple PID loop in the robot's software is the basis of the balance control:

The proportional term takes the angle error of the bot and sends that scaled value to the motors, to keep the wheels rolling into the fall.

The integral term is used the same way, but is the sum of all angle errors over time and helps to cancel out center-of-gravity issues.

The derivative term is critical. Without it, we couldn't control acceleration.

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de}{dt}$$

Applications

- Rocket propeller works on the principle of inverted pendulum.
- Segway Robot can be used in such a way to travel forward or backward.They can be even used as vehicles by humans.

Works Done

The mechanical part is completely done.Now we are trying to complete the software portion. We have already achieved the orientation angle of the bot by reading the register values of IMU. And we also designed the PID controller which we are going to use to control the wheels.

Future Work

In future we will complete the remaining part of balancing the bot and will try to control the bot using joystick and Xbee module.