

# An implementation of control system in UGV

1<sup>st</sup> Given Name Surname  
dept. name of organization (of Aff.)  
name of organization (of Aff.)  
City, Country  
email address or ORCID

2<sup>nd</sup> Given Name Surname  
dept. name of organization (of Aff.)  
name of organization (of Aff.)  
City, Country  
email address or ORCID

3<sup>rd</sup> Given Name Surname  
dept. name of organization (of Aff.)  
name of organization (of Aff.)  
City, Country  
email address or ORCID

4<sup>th</sup> Given Name Surname  
dept. name of organization (of Aff.)  
name of organization (of Aff.)  
City, Country  
email address or ORCID

**Abstract**—Navigation of Unmanned Ground Vehicles(UGVs) plays a key important role. Imperfections in vehicle designs, motors and its driver will underpins such navigation capabilities. We propose a simple PID controller with a gyroscope feedback to control the path of UGV.

**Index Terms**—UGV, PID

## I. INTRODUCTION

Navigation in space using UGV is very challenging task if we design the controller assuming ideal situations. Imperfections arise in imbalanced chassis design, voltage fluctuations and gear slippages in motors. Giving same control input to same motors would'nt result in same speed due to the mentioned non-idealities, so tracing a straight line is almost impossible. Due to difference in torques of motors this would result in rotation in UGV. To correct this we use a simple PID controller with gyroscope feedback (MPU6050) to correct the path. We get angular velocities and accelerations from the MPU6050 then from which we estimate angular orientations. This can be done using *kalman filters* but considering computation capability of embeded controllers we use *complementary filters*.

We then use these estimated angles to estimate the error with actual desired angle and then use PID controller to correct the speeds of motors.

## II. ARCHITECTURE

First we estimate angular orientations of UGV using gyroscope data. We are working with UGV which has 2 degrees of freedom , we try to correct the angle  $\theta_z$ (or Yaw). We use *complementary filter* to estimate the angles from initial conditions.

$$\theta_x[n] = (1 - \alpha)(\theta_x[n - 1] + \omega_x[n]\delta t) + \alpha \tan^{-1} \left( \frac{\dot{\omega}_y}{\dot{\omega}_x^2 + \dot{\omega}_z^2} \right) \quad (1)$$

$$\theta_y[n] = (1 - \alpha)(\theta_y[n - 1] + \omega_y[n]\delta t) + \alpha \tan^{-1} \left( \frac{\dot{\omega}_x}{\dot{\omega}_y^2 + \dot{\omega}_z^2} \right) \quad (2)$$

$$\theta_{actual}[n] = \theta_z[n] = \theta_z[n - 1] + \omega_z[n]\delta t \quad (3)$$

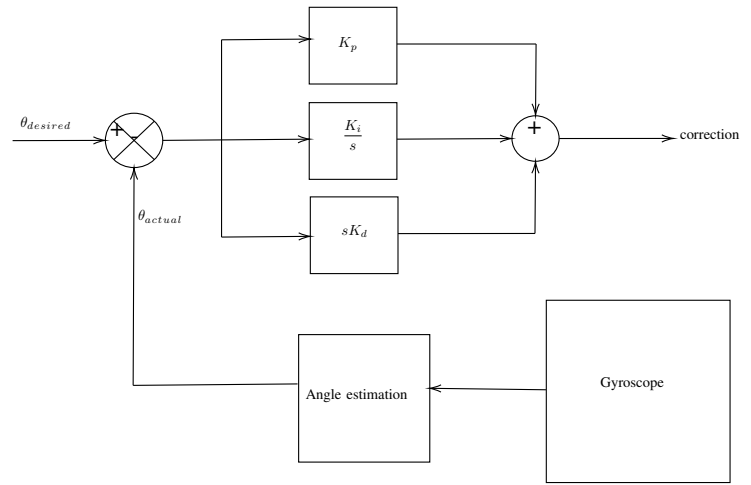


Fig. 1. System architecture

Let  $\theta_{desired}$  denote target or desired angle. Figure II depicts the overall control system of the UGV. Proportionality constants  $K_p$ ,  $K_i$  and  $K_d$  are tuned using Ziegler–Nichols method.

The *correction* which is obtained is used to correct the speeds of the motors in the following fashion.

$$Speed_{left}^{new} = Speed_{left}^{old} - \frac{correction}{2}$$

$$Speed_{right}^{new} = Speed_{right}^{old} + \frac{correction}{2}$$

We keep desired angle as 0 which ensures that UGV travels in straight path. We change this target angle dynamically according to our path. For example to traverse a square we increment the angle by 90 starting from 0 frequently.

## III. EXPERIMENTS

We use *TiHAN UGV Kit* to perform these experiments. ESP32 as microcontroller and MPU6050 is used as gyroscope.

After several episodes of tuning the proportionality constants we find that  $K_p = 30$ ,  $K_i = 2$  and  $K_d = 1$  gives best performance.