Plot clock

Prashant(140010034)
Amey Mankar(14D170017)
Omkar Masurkar(14D170007)
Harshal Jalan(140110017)

GROUP NO 397

Goal articulation

The clock will write the time on a whiteboard. It’ll require a synchronous movement of multiple components. The clock will write the time in HH:MM format. The clock will write the time using servo motors, mechanical arms and gears. The movement of the arms should be such that to cover sufficient area of the board.

The bot will be given coordinates using the Cartesian coordinate system and the bot will convert it to radial coordinates. The bot will erase the previously written time and then print the current time.

implementation

Implementation of the project was done in two major stages.

- ❑ Mechanical
- ❑ Software

Mechanical model

We have made an entirely new bot , not using FIREBIRD V. The bot has been made using acrylic sheets and wood as the primary standalones. It also uses, servo motors and a castor wheel.

Here you can see the SolidWorks model for our bot . The working has been explained in the video.

Solidworks video

Control equations

We derived equations to calc dx and dy in terms of dr,dtheta(change in angle of the second servo) and theta(angle of the second servo)

Further we derived dr in in terms of phi(angle of first servo) and dphi(change in angle of first servo)

dx= dr

dy=

dr=

software

We are using an arduino uno microcontroller based on the AtMega328 microprocessor. A timer will run on the microcontroller which will act as the clock. The microcontroller will write the time by sending commands to the servo motors. This would be initiated by a user input and once initiated, will write the time at regular intervals until another user input is given. We have used servo and time inbuilt libraries from Arduino.

Screnshots of the code

Major challenges faced

Designing and an altogether new bot was a very big challenge. Construction of bot took more time than we expected. Even after we completed the construction we faced mechanical problems, due to friction between moving parts and various mechanical inaccuracies like;

1) asymmetry between arms of bot,
2) too much friction b/w the gears, and slipping b/w the servo and gears.

How we overcame them?

The part which did the radial motion of the bot could not be made to work even after a huge amount of efforts. All things including rubber bands, a glue gun, a soldering gun and even m-seal were tried but they all up to our expectations. Upon suggestion by our TA, we created a method to prove the working of the code by creating a c++ program that took as input the point to move to and gave as output a dump which contained all the points which it reached in sequential order. The data from the dump was plotted(using Excel) to show the path taken by the bot's arm, which indeed came out to be a straight line. This proved the accuracy of both the control equations and the code.

simulation

Major challenges faced

Apart from Mechanical we faced a problem while implementing the timer.

We needed to time each loop so that the system's time could be updated. Generally, to time a loop, the length of the timer is kept longer than the maximum length of the loop and the timer is register checked for overflow before starting next iteration of while loop. However that could not work in our case.

How we overcame them?

We used a timer interrupt to count the number of overflows taking place within one iteration of the while loop and added that to the system time, then reset the counter.

Future work

❑Remote controlled operation by implementing Xbee(can be done as a part of this