**The design and simulation of the control system:**

Chart, line chart

Description automatically generatedPWM signal = 30000

Sampling time = 50ms

Right wheel:

Chart

Description automatically generated

\*The desired 0 dB frequency of the left = 10rad/second

The desired gain = 70dB

Using PI controller,

For the left wheel,

High frequency gain =

Low frequency = one decade below the desired 0 dB frequency,

, where The step response of the right wheel:

Chart

Description automatically generated

By Tustin method and using sampling time = 2.5ms,

A picture containing text, sky, map, day

Description automatically generatedBode plot of the compensated system: Comparison with the uncompensated:

Chart

Description automatically generated

Left wheel:

Graphical user interface, chart

Description automatically generated

Using the desired 0 dB frequency of right wheel = 10 rad/second

The desired gain = 67dB

Using PI controller,

For the left wheel,

High frequency gain =

A picture containing chart

Description automatically generatedLow frequency = one decade below the desired 0 dB frequency,

* , where The step response of the left wheel:

By Tustin method and using sampling time = 2.5ms,

Bode plot of the compensated system: Comparison with the uncompensated:

Chart

Description automatically generatedChart

Description automatically generated

Therefore,

Bode plot of two closed loop transfer function(): Step Response of

Graphical user interface

Description automatically generated with low confidenceChart

Description automatically generated

Using PI controller,

The gain of the outer loop PI controller = -6dB

Low frequency = one decade below the desired 0 dB frequency,

* , where

Bode plot of compensated closed loop transfer function: Comparison:

Chart

Description automatically generatedChart, histogram

Description automatically generated

Step Response of the whole closed loop system:

A picture containing chart

Description automatically generated

By Tustin method and using sampling time = 2.5ms,

Graphical user interface, text, application, email

Description automatically generated

Octave code for the joint lab

**Graphical user interface

Description automatically generated**

**The result**

The data collected right after the recording

Since Tustin equation is used, the encoder value raises with a slope.

Noise generated maybe caused by the defect of physical components.

**The implementation of the system:**

According to the guidelines, the imputed data at the very beginning, named “left” and ”right”, should be the current counter value of the left and right wheel. The minimum and maximum value should be within 0 to 15. In order to apply the Tustin equation, the difference between the desired encoder value and current encoder value should be calculated and saved as the last encoder difference. The adjusted speed and desired speed also change along the run.

Text, letter

Description automatically generated

…



The code of the individual PI controller

The wheels can perform speed control through their very own Tustin equation. However, in order to regulate their changes, one more PI controller is added to the system. The change in speed should be in form of the desired speed +- (regulated speed). Some optimization is done by increasing the regulated speed of the right wheel.

Text

Description automatically generated

…



The code of the outer PI controller

**The difficulties in the implementation**

Since Tustin equation is used, the calculation of the speed is not a difficult, but requires carefulness to process. The major difficulties I had is how to find out the error/difference between encoder value. In order to generate a difference along time, 100 sets of data are added before calculation. Thus, the speed is corrected every 0.25s, which results in slightly jiggle of the cart. Also, the code for the adding loop before the calculation is also hard to be implemented. However, with the use of dummy case and test value, the loop is completed.

Moreover, at the early stage, the equation of L\_inter and R\_inter will produce strange results. Later, it is known that the initial value of the last encoder value difference should not be zero. It should be some constant to cancel out the change due to current encoder difference for the first speed adjustment. After that, the last encoder value difference will be updated, and calculation can be done after the first adjustment.

Finally, the adjustments are quite unstable before, and it is caused by the difference of the counting of encoder value. By resetting the counting, a more stable adjustment can be archived. At the same time, the cart need to be run at least one run to allow the wheels to reach a steady state for the adjustment.