6.100 Fall17 Project report

Wilson Li
{litianyi@mit.edu}
May 16,2018

1.Poject discription

Achieving Stable Steering for Gizmo Robots

"Gizmo Robots" (small toy cars with Arduino processors) are used to simulate driving using both "car following" and "bilateral control" modes. The robots follow a closed loop track of black tape (or black track printed on paper).Before the emergent dynamic behavior of a set of twenty or so robots can be demonstrated, the robots have to be able to reliably follow the edge of the road using their road sensors, without injecting perturbations into longitudinal control.

The task is to arrange for robots to reliably follow the road. This involves programming the Arduino using a cross-compiler on a laptop in a simplified C language. For debugging, telemetry information returned from the robot has to be obtained and analysed. This can be done in a USB cord tethered mode, but the cord interferes somewhat with the driving, so a telemetry log file on an SD card on the robot is preferred. Steering is controlled using a software PID controller (proportional, integral, derivative)based on outputs from two road sensors, one ahead of the driven wheels and one behind.

Most of the infrastructure for this projects exist. The task is to use it to achieve stable non-oscillatory steering. This involves tuning the control system gains, rewriting some of the servo code, calibrating the road sensors,possibly moving the road sensors up or down or sideways, applying visible light blocking filters to the infrared sensors to suppress ambient light and so on.

2.Context

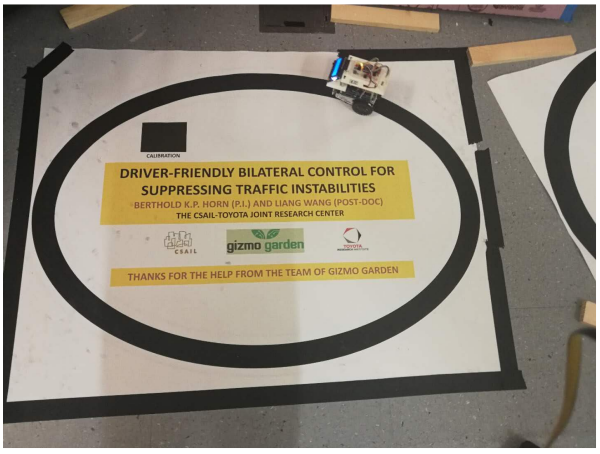
(1)***Critical Damping***:Critical damping provides the quickest approach to zero amplitude for a damped oscillator,where the system returns (exponentially decays) to equilibrium without oscillating. With less

damping (underdamping) it reaches the zero position more quickly, but oscillates around it. With more damping (overdamping), the approach to zero is slower. Critical damping occurs when the damping coefficient is equal to the undamped resonant frequency of the oscillator.Critical Damping is important so as to prevent a large number of oscillations and there being too long a time when the system cannot respond to further disturbances. In our steering control case,we use critical damping to impose constraints on control-related parameters in control algorithms to achieve fast-stabilized steering system.

If we use y to indicate one measurement of our controlled variable and y0 to be its desired value,then given a second-order system:

(2)***PID control***:A proportional–integral–derivative controller (PID controller or three term controller) is a control loop feedback mechanism widely used in industrial control systems and a variety of other applications requiring continuously modulated control. A PID controller continuously calculates an *error value* e(t) as the difference between a desired setpoint (SP) and a measured process variable (PV) and applies a correction based on proportional, integral, and derivative terms (denoted *P*, *I*, and *D* respectively) which give the controller its name.We'll implement such a linear controller for our steering.Our measured variable will be to what extent are we off the road and desired setpoint will be 0.

(3)***Setup***:



3.Workflow

4.Hardware

(1)**Apply visible light filter**

Visible light filter is installed on road sensor to filter ambient light in the room which could mess up the measurement of light reflected from the road,in turn,messing up the lateral distance measurement.

(2)**Recalibrate road sensor transfer function**

The transfer function describes relationship between phototransistors readings and lateral distance relative to the road

edge.Considering that lateral distance is our controlled variables,so the accuracy of this transfer function is crucial to our system.To derive such a more accurate transfer function,we'll move the robot along the lateral axis with small variations at each step by conducting this experiment on top of a piece of paper. Under current robot configuration,we have the following plot:

The middle linear area from approximately -4.5mm to 4.5mm is our ideal working range for the robot.Slope for this linear range can be approximated as yslope=-48.9/mm.With yslope,we can link yb,yf(front lateral position,back lateral position) with the raw phototransistors reading from the road sensors.Real test on robots show that the robot normally won't exceed this range while turning if robot doesn't lose control.

5.System construction

It’s reasonable to approximate our discrete system with a 20ms of sampling rate to a continuous system.Then we could apply more straightforward differential equations than difference equations.
We use a block diagram to represent our closed-loop control system in the following graph:

1. Controller

For controller,we implement PD control and ignore integral control,where cp is proportional gain,cd is derivative gain and ci=0.
We base on real-time error of lateral position to impose extra lateral acceleration to steer. We control the wheel rotations to achieve a desired acceleration while keeping the speed of robots moving forward unchanged.

2. Plant

Plant represents the system dynamics.

(1)Let w be the wheel base (measure w) and let α be the fraction of w that the road sensors are from the left wheel. Here α = 0 would correspond to the road sensors being in line with the left wheel, while α = 1 would correspond to the road sensors were in line with the right wheel. Actually, line connection the two sensors is a bit to the left of the right wheel. The perpendicular distance of the left wheel from the line

connecting the road sensors is αw, while that from the right wheel is $(1-\alpha)w$.We measure that $\alpha=15/64$ and $w=107mm$.

(2)Two road sensors are mounted at the same lateral offset from the center of the robot and sense the distance from the road edge at two different positions in the direction of travel. Let d_f and d_r be the offsets in the direction of travel with respect to the wheel base (line connection the places where the two wheels contact the ground). Note that the front sensor is mounted ahead of the wheel base (d_f = +30 mm) while the rear sensor is mounted behind it (d_r = −25 mm).

(3)*y* and *θ* each represents the lateral position of the entire robot by linear interpolation and to what extent robot depart from the road relative to running direction.*v* represents the longitudinal speed of the entire robot,while *vl* and *vf* corresponds to the speed of the left wheel and right wheel.

3. Critical Damping

We can analyze the resulting second order linear differential equation in controller where y_0 is the desired lateral position.
If cd = 0, then the system will oscillate with angular frequency -cp (rad/sec) in response to disturbances (without decay). If cd > 0, oscillatory responses will be damped out anbd decay as $e^{-cd/2}$. If cd^2 > -4cp, both roots are real and the response is not a damped oscillation. The transition between the two cases occurs when cd^2 = -4cp. This is the case of critical damping.

4. Arduino Gains

Now we'll turn the constraints on parameters in controller into constraints on gains in Arduino code.The key point here is to consider the units of all variables in code and in reality.

In Arduino,we use the following control command:

“turn” indicates the extra lateral acceleration we enforce on the robot.In code,”turn” has a unit of mm/s^2.To correctly implement this desired acceleration,we need to make corresponding changes in the speed of both wheels in the following way:

y in reality has a unit of mm.Thus,compared with what we have in the controller in block diagram, we could derive that cp = -ksa.

From the plant,we have .

Value of *v* in the code has a range of 0 to 100,which corresponds to 0mm/s to 155mm/s in reality.During the turning,we keep *v* remain unchanged.Combined together,we have:

cd = ksb/(70*155)

Given that the condition for critical damping is $cd^2 = -4cp$,we can derive that:

ksb = 217 $\sqrt{ksa}$

If we increase ksb and ksa at the same time,we could reach a faster convergence time for error.But higher combination of gains will result in high sensitivity to small disturbances and cause system to generate many small vibrations.Here we have a trade-off.We pick a moderate ksa=9 and ksb=650.

6.Testing

We'll then test this set of parameters(ksa=9,ksb=650) on Gizmo robots.After collecting SD card data for 2 rounds of run,we make the following three plots:

(1)

From the graph,we could see that the angle oscillates around 0.03 due to noise.We could also know that our angle offset is 0.03.The variations from 0.03 due to turning only reach around 0.04 or 0.05,which also show that no sharp turning is required for steering.Thus,it indirectly indicates smooth steering.

(2)

From the graph,we could see that the lateral position oscillates around 0 with tolerable noise when not turning.When turning,there’s a sharp peak in the graph,which directly indicates that oscillation caused by feedback steering decays nearly exponentially.Critical damping is thusly achieved.

(3)

From the graph,we could see that the speed of robot indicated by the middle yellow line did remain unchanged around 70 mm/s.