ECE3 23S Final Report Template

Name#1	Lana	Lim	UID#1	105817312
	First (Given)	Last (Family)		
Name#2	Anthony	Wang	UID#2	706028136
	First (Given)	Last (Family)		

PLAN THE DEVELOPMENT

Step #1

Get calibration data. Worked on it in the lab by using a small sample of track to get original sensor fusion values.

Step #2

Write code to process sensor fusion values. Determining the weight schema, get raw sensor fusion values, multiply values with weights, and normalize values.

Step #3

Calculate PID. Some questions to consider at this stage: How to calculate the correction value? What is the significance of Kp, Ki, Kd?

Step #4

Start testing the system with the final track and implement cases like donut turn and final stop.

Step #5

After finding a working base with associated Kp, Ki, and Kd values (speed is not important), start tweaking the variables to achieve faster speeds.

ECE3 23S Final Report Template

Name#1	Lana First (Given)	Lim Last (Family)	UID#1 105817312
Name#2	Anthony First (Given)	Wang Last (Family)	UID#2 706028136

CONDUCT TESTS

Parameters

Initial speed, Kp, Kd, Ki, time and speed it takes a car to do a donut, encoder count timestamps, weighting schema.

Controlled Variables

Correction added to the wheel speeds depending on sensor input, normalized sensor values, LED states, speeds on certain encoder counts.

Observed Variables

Raw sensor input, calculated error, sensor mins/maxes.

Procedure

1. Choose a starting position, initial speed, Kp, Kd, and Ki values. Record them in a table entry.
2. Start the car. If it's a practical test run, record and time the car until it completes/fails the track. If it's a debugging test run, record fused sensor values outputted in the serial monitor. Please note that while debugging, the car may operate less efficiently since it must also write to the serial monitor.
3. Write notes of the trial after the trial is finished.
4. Adjust initial speed, Kp, Kd, and Ki values from trial results. Repeat steps 1-3 for a new trial.

ECE3 23S Final Report Template

Name#1	Lana	Lim	UID#1	105817312
	First (Given)	Last (Family)		
Name#2	Anthony	Wang	UID#2	706028136
	First (Given)	Last (Family)		

ANALYZE & INTERPRET DATA

Sections of your hand-written logbook (full logbook to be included in the Appendix)

Found working base with fixed speed 40:

initial speed	Kp	Kd	Ki	starting position	weights	
40	8.5	0	0	0	-	- fix the tick - lowered the speed so car can react to sharper turn. - increase time it takes for car to do donut - detects u-turn for part of track for 2 thick black bars.
40	8.5	0	0	0	-	- changed how car detects all black
40	8.5	0	0	0	13,14,13,8	- car thinks vertical black is track, aligns itself with black bar → black bar ends → undefined
40	8.5	2	0	0	15,14,12,0	- same behavior, but worse
40	8.5	-2	0	1	-	- does not complete first turn
[redacted]						
40	8.5	0	0	1	-	- process sensor input differently, check if it detects a maximum weight in any of the sensors
40	8.5	0	0	1	-	- does not complete first turn
40	8.5	0	0	1	-	- detects 2 maximum weight values = sees vertical black bar - switches left/right wheel speeds
[redacted]						
40	8.5	0	0	1	-	- track completed!
40	8.5	0	0	2	-	- " "
40	8.5	0	0	3	-	- " "
40	8.5	0	0	4	-	- " "

This is where we built our intuition for adjusting Kp, Ki, and Kd values based on our observations from each trial:

- A low Kp results in little to no correction and the car goes straight off the track.
- A big Kp results in big oscillations that throw the car off the track.
- The intuition behind the value of Kd is to minimize the amplitude of the oscillations caused by Kp.
- Low speeds work most of the time, but higher speeds require more tweaking of Kp, Ki, and Kd so the car can react fast enough and correctly when it's offset to the track.

ECE3 23S Final Report Template

Name#1	Lana	Lim	UID#1	105817312
	First (Given)	Last (Family)		
Name#2	Anthony	Wang	UID#2	706028136
	First (Given)	Last (Family)		

ANALYZE & INTERPRET DATA

Graphs & tables, each accompanied by interpretations of data being displayed

Figure 1. Relationship with Kp and Ki where the car's initial speed is 80. Red marks a failed trial and blue marks a successful trial.

Applied our knowledge and intuition from the base case to find a working system where car's initial speed is 80. Figure 1 is the relationship between Kp and Kd. We chose to leave out Ki in this case since we believe changing the value for Ki is only necessary for making fine-tune adjustments to an already working system. We approached this by having a mix of fractional and noticeable changes to Kp and Kd.

ECE3 23S Final Report Template

Name#1	Lana	Lim	UID#1	105817312
	First (Given)	Last (Family)		
Name#2	Anthony	Wang	UID#2	706028136
	First (Given)	Last (Family)		

ANALYZE & INTERPRET DATA

Graphs & tables, each accompanied by interpretations of data being displayed

Fused sensors were on a 15, 14, 12, 8 weighting schema.

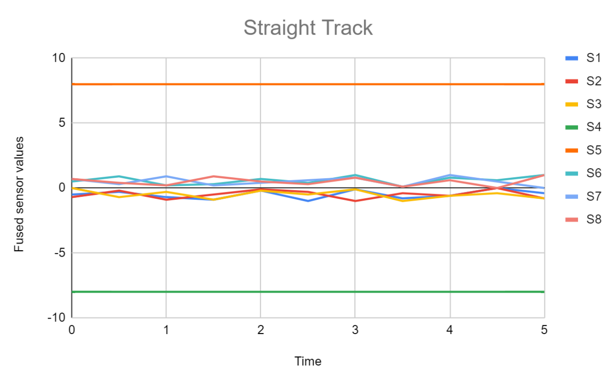

Figure 2. Serial monitor output of fused sensors when the car is perfectly aligned with the straight track.

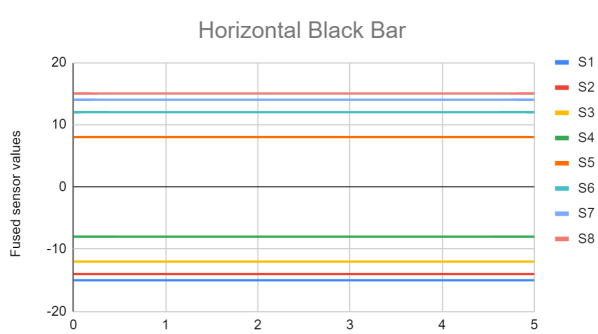

Figure 3. Serial monitor output of fused sensors when the car is on the horizontal black bar of the track.

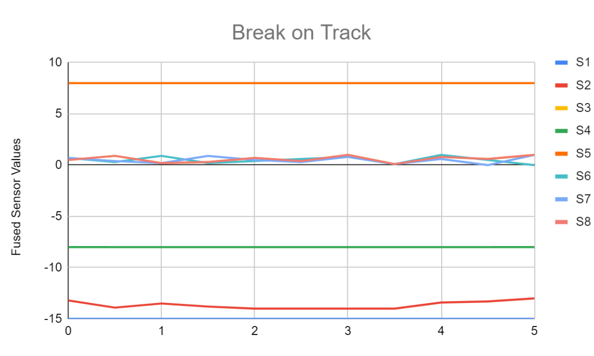

Figure 4. Serial monitor output of fused sensors when the car is directly on the break of the track

Our biggest challenge in the project was how to make the car respond correctly when reaching the break on the track. In all other instances, there is only one focused area of the car's sensors that detects all black (or all sensors if it's on the horizontal black bar). Though, when the car reaches the break on the track, there are two areas where some sensors see all black. After noticing this in the serial monitor, we chose to encode a separate case for when the car reaches the break.

ECE3 23S Final Report Template

Name#1	Lana	Lim		UID#1	105817312
	First (Given)	Last (Family)			
Name#2	Anthony	Wang		UID#2	706028136
	First (Given)	Last (Family)			

APPENDIX

Your entire hand-written log.

Day 1 05/01/23
- took path calibration values (8 min, max, weights)

Day 2 05/08/23
- Goal: straight line path
- Test 1:
  - car corrected itself halfway, then turned right ⇒ kept going right
  - modulized code.

Day 3 05/13/23
- made it PID capable

TEST CASES

speed	Kp	Kd	Ki	start pos	weights	Result/Notes
50	15	0	0	ϕ	13, 14, 12, 8	- adjusts accordingly for straight line, overcorrects for first turn
50	0.5	0	0	ϕ	-	- overcorrects for straight line.
50	0.5	0	0	ϕ	-	- passes straight line, does not do donut at all-black.
50	0.5	0	0	ϕ		- did a turn, no donut at all-black - placed it on actual track, completes first turn but fails sharper turn. - it ticks randomly - when car is perfectly centered, thinks it sees all black
40	0.5	0	0	ϕ	-	- fix the tick - lowered the speed so car can react to sharper turn. - increase time it takes for car to do donut - detects u-turn for part of track for 2 thick black bars.
40	0.5	0	0	ϕ	-	- changed how car detects all black
40	0.5	0	0	ϕ	13, 14, 13, 8	- car thinks vertical black is track, aligns itself with black bar → black bar ends → undefined
40	0.5	2	0	ϕ	15, 14, 12, 8	- same behavior, but worse
40	0.5	-2	0	1	-	- does not complete first turn
30	0.5	0	0	1	-	- speed is not the problem, slower speed does not make car correct itself
40	0.5	0	0	1	-	- process sensor input differently, check if it detects a maximum weight in any of the sensors
40	0.5	0	0	1	-	- does not complete first turn
40	0.5	0	0	1	-	- detects 2 maximum weight values = sees vertical black bar - switches left/right wheel speeds
50	0.5	0	0	1	-	- make it complete faster. - goes off track
40	0.5	0	0	1	-	- track completed !
40	0.5	2	0	1	-	- track completed
40	0.5	50	0	1	-	- track not completed
40	0.5	15	0	1	-	- track not completed
40	0.5	10	0	1	-	- track completed

ECE3 23S Final Report Template

Name#1	Lana	Lim	UID#1	105817312
	First (Given)	Last (Family)		
Name#2	Anthony	Wang	UID#2	706028136
	First (Given)	Last (Family)		

APPENDIX

Your entire hand-written log.

speed	Kp	Kd	notes
45	10		- worked, was a little faster
50	11.5		- worked, got a little faster, oscillated a lot.
55	11.5		" "
60	17	5	- did not work
60	17	-2	- did not work
65	19	-7	- did not work
65	19	-5	- did not work
65	21	-5	- did not work
65	21	0	- worked first half, still oscillates too much.
65	21	1	" "
65	21	-.5	" "
70	21	-2	- wobbly, but worked.
85	35	-5	- wobbly, did not recognize first black bar.
85	35	-7	- completed first half, but then got too wobbly.
85	35	7	- did not work
87	35	-7	- did not work.
85	35	-12	- did not work.
85	35	-20	- did not work.
85	40	-7	- did not pass vertical black bars.
85	50	-10	- did not work.
85	35	7.5	- worked first half.
85	35	-7.75	- worked first half.
85	35	-7.25	- did not work.
85	35	-8	- did not work.
85	30	-7.75	- did not work.
85	21	-7	- did not work.
85	25	-7	- did not work.
85	27.5	-7	- worked first half.
85	27.5	-7.25	- did not work.
85	20	7	- did not work.
85	20	0	- did not work.
85	27.6	-7	- did not work.
85	21	-2	- worked, fastest working version
			- tested on 3 other positions ⇒ worked.

ECE3 23S Final Report Template

Name#1	Lana	Lim	UID#1	105817312
	First (Given)	Last (Family)		
Name#2	Anthony	Wang	UID#2	706028136
	First (Given)	Last (Family)		

APPENDIX

Your entire hand-written log.

Code outline:

1. Sensor fusion
   - global uint16_t sensorValues[8]
     - array of sensor values that we operate on, global so we don't have to deal with pointer issues
   - void normalizeSensorValues()
     - normalizes sensorValues based on calibration data
   - void weightSensorValues()
     - multiply sensorValues with weighting scheme
2. Calculate PID
   - void calculateMotorOutputs()
     - takes final array and turns weighted + normalized sensorValues into motor outputs
   - void setMotorSpeeds()
     - sets up the car, pwm values
   - calculatePID()
     - uses array of sensor values to calculate PID constants

Main headers J1-J4:

	Energia pin #	J1 1	J3 21	Energia pin #			Energia pin #	J4 40	J2 20	Energia pin #	
CC2650/CC3100	1	3.3V	5V	21	CC2650/CC3100	PWML, Left Motor PWM	40	P2.7	GND	20	CC2650/CC3100
CC2650	2	P6.0	GND	22	CC2650/CC3100	PWMR, Right Motor PWM	39	P2.6	P2.5	19	CC2650/CC3100
CC2650/CC3100	3	P3.2	P6.1	23	Center IR Distance / OPT3101	PWM Arm Height Servo	38	P2.4	P3.0	18	CC3100, SPI_CS, GPIO
CC2650/CC3100	4	P3.3	P4.0	24	Bump 0 [3]	CC3100, UART1_CTS	37	P5.6	P5.7	17	available GPIO? / OPT3101 RST?
nHIB	5	P4.1	P4.2	25	Bump 1 [3]	CC3100, UART1_RTS	36	P6.6	!RST	16	CC2650/CC3100
Bump 2 [3]	6	P4.3	P4.4	26	TExaS scope input	CC2650	35	P6.7	P1.6	15	CC3100 SPI MOSI
CC3100, SPI_CLK	7	P1.5	P4.5	27	Bump 3 [3]	CC3100, NWP_LOG_TX	34	P2.3	P1.7	14	CC3100 SPI MISO
Bump 4 [3]	8	P4.6	P4.7	28	Bump 5 [3]	CC3100, WLAN_LOG_TX	33	P5.1	P5.0	13	ERB (3.3V) [1]
UCB1SCL [4]	9	P6.5	P5.4	29	DIR_L	PWM Arm Tilt Servo	32	P3.5	P5.2	12	ELB (3.3V)[1]
UCB1SDA [4]	10	P6.4	P5.5	30	DIR_R	nSLPL [2] / nSLPR [2]	31	P3.7	P3.6	11	PWM Gripper Servo

Notes:
[1] This is encoder output. Sever VPU=VREG jumper and connect VPU to 3.3V
[2] This disables a motor driver. 0 to sleep/stop. Sever VCCMD=VREG jumper and connect VCCMD to 3.3V. Consider severing nSLPL=nSLPR jumper.
[3] Use Port 4 for edge-triggered interrupts
[4] Primary I2C channel supported by Energia
Bump 0 is right side of robot, Bump 5 is left side
CTRL on the motor board is a power switch. A high pulse (>1V) turns on the switch; a low pulse turns off the switch and power to the microcontroller. Leave this pin floating (an input) for normal operation.
Yellow highlights changes from previous pin assignments
Red highlights changes from version 4
Grey is changes from version 5
Orange needs to verify with Jan if routing possible to combine nSLP to free up an additional PWM pin

J5:

				no energia															
	Yellow Front Right LED	Right IR Distance / OPT3101	Analog Arm Height Servo	Analog Gripper Servo	Reflectance LED Illuminate (odd)	AUXR IR Distance / OPT3101	Reflectance 3	Reflectance 1	Nokia5110 CS	Nokia5110 CD	Yellow Front Left LED	Reflectance 4	Reflectance 6	UCB3CLK (not available in	UCB3SDA (not available in	ERA (3.3V)[1]			
Energia #	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	5V	3.3V	GND
	P8.5	P9.0	P8.4	P8.2	P9.2	P6.2	P7.3	P7.1	P9.4	P9.6	P8.0	P7.4	P7.6	P10.0	P10.2	P10.4	5V	3.3V	GND
	P8.6	P8.7	P9.1	P8.3	P5.3	P9.3	P6.3	P7.2	P7.0	P9.5	P9.7	P7.5	P7.7	P10.1	P10.3	P10.5	5V	3.3V	GND
Energia #	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	5V	3.3V	GND
	Red Back Left LED	Red Back Right LED	Left IR Distance / OPT3101	Analog Arm Tilt Servo	Reflectance LED Illuminate (even)	Nokia5110 RST	AUXL IR Distance / OPT3101	Reflectance 2	Reflectance 0	Nokia5110 Clock	Nokia5110 MOSI	Reflectance 5	Reflectance 7	AUXC IR Distance / OPT3101	UCB3SCL (not available in	ELA (3.3V) [1]			

11/19/2018 changes from version 6, jan@pololu.com
2/11/2019 changes from rom05a02