CSE 535 Project 2: Context-Aware Adaptation

In Project 1 you learnt about developing a monitoring application using on-board sensors in a smart phone. In this project, we will design a context aware adaptation strategy for the development of autonomous braking system for Level 3 autonomous cars which assumes an underlying distributed monitoring and actuation cyber-physical system specifically human-in-the-loop and human-in-the plant. Before we delve into the details let us understand some basic definitions.

Level 3 autonomy: The vehicle operates autonomously, however, requires the driver to be attentive at all times for potential switch to manual mode.

Controller: It is a software that takes sensor data from the vehicle such as speed, distance between cars and acceleration or deceleration and outputs the braking pressure to apply deceleration to stop the car.

Controller gain: It is a property of the controller that determines how aggressively a controller will apply brakes and stop the car.

Deceleration Limit: This is the maximum limit on the deceleration applied by the controller. High deceleration can lead to unwanted harm to the driver behind the wheel. The controller can never apply deceleration that is higher than the maximum limit.

Controller frequency: It is the number of times the controller needs to compute a new braking input per second.

Reaction time: It is the time taken for the human to decide on an action after a switch to manual mode

Action time: Time taken to execute the action decided by the human and stop the car.

Autonomous vehicle braking system example

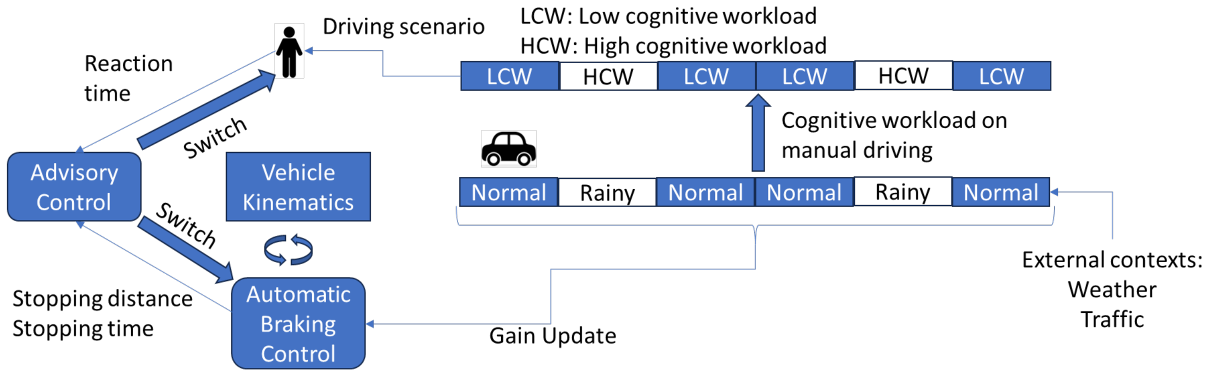

Figure 1: Autonomous Driving example

Consider a user driving a Level 3 autonomous car (Fig. 1). We are specifically focusing on the autonomous braking system. As a part of the autonomous braking system, there is an **advisory control** that:

a) processes a driving scenario, by sensing environmental conditions
b) Predicts if the autonomous braking system will be able to stop the car
c) Determines if the control should be transferred to the human behind the wheel

Autonomous braking system has a feed-back loop controller that senses the distance between the autonomous car A and the car in front I, and computes a braking force as a function of the A's initial speed v_A, and distance between A and I. The braking force applied is also a function of the controller gain G, which is a design parameter that you will have to tune.

The braking force is constrained by the maximum deceleration limit a_{max}, which is dependent on the road condition. For example, on dry road you can apply more deceleration than on wet road in rainy season. The braking force is thus given by the following equation:

$$Braking\ force\ b = \min\left(f(v_A, d_{AI}, G), a_{max}\right).$$

Once the braking force b is applied, the vehicle kinematics show how the vehicle behaves in the real world. Given the initial speed v_A and initial distance between A and I, d_{AI}, the vehicle kinematics gives the stopping distance d_{stop}, and stopping time t_{stop} for the vehicle. This you can obtain using the vehicle kinematics Simulink model ($M_{vehicle}$) provided to you.

If $d_{stop} \geq d_{AI}$ then the autonomous vehicle fails, and the cars collide, else there is no collision. If cars collide, then stopping time t_{stop} is same as collision time t_c.

In this example, we will fix the distance between cars A and I, d_{AI}. However, changing initial velocity v_A, controller gain G, and deceleration limit a_{max}, can change the stopping distance and stopping time resulting in potential collisions.

Human driver model

The human driver is assumed to be alert all the time to take over control. However, we have to account for the reaction time t_r of the human driver after a switching signal is sent through the dashboard controls. In addition to the reaction time the human action also takes some time to execute.

In this example, we assume that the human provides a deceleration that is 10% higher than the deceleration limit a_{max}, i.e. $h_a = 1.1a_{max}$. The action time t_a can be obtained by removing the braking controller from the Simulink model and applying a static deceleration of h_a into the vehicle kinematics model. After simulation the action time can be obtained as the stopping time reported in the Simulink model.

The total time taken by the human to completely stop the car is $h_{stop} = t_r + t_a$.

We are now in a position to define a basic logic for the advisory control. The advisory control uses the following algorithm to decide if it wants to switch:

SWITCH = ADVISORY CONTROL (v_A,G,a_{max}, t_r)

Step 1: Predict if autonomous braking control collides or not using $v_A, G, and\ a_{max}$.

Step 2: If no collision then

Step 3: Do not switch to manual control

Step 4: Else

Step 5: **Collision time t_c = predicted stopping time from Step 1**

Step 6: **Predict action time t_a, using v_A, remove controller, and constant deceleration** $1.1a_{max}$.

Step 7: **Predict reaction time** t_r

Step 8: **Compute** $h_{stop} = t_r + t_a$

Step 9: **if** $\boldsymbol{h_{stop} < t_c}$

Step 10: **Switch to human**

Step 11: **Else**

Step 12: **Do not switch.**

Driving Context Changes

Road conditions depend on several factors including:

a) User mobility patterns
b) Traffic conditions
c) Weather patterns

Each context can be modeled using stochastic modeling techniques such as Markov chains. In this assignment you are given a composite model of driving context that has two states: a) normal with deceleration limit -200 $\frac{m}{sec^2}$ and b) poor road conditions with deceleration limit -150 $\frac{m}{\sec^2}$.

The prior probabilities of each state $p(normal) = 0.6, and\ p(poor) = 0.4$. The transition probabilities are given by the matrix:

$$q = \begin{array}{ccc} & Normal & Poor \\ Normal & 0.6 & 0.4 \\ Poor & 0.85 & 0.15 \end{array}$$

Human Reaction time model

The cognitive workload of a human changes based on driving context. We will assume that normal driving conditions have low workload, and poor driving conditions require high workload.

Consider Table 1 in https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6338053/pdf/fnhum-12-00525.pdf

The heart rate and respiratory rate is provided for single task under low cognitive workload (LCW) and high cognitive workload (HCW) tasks. Use the mean and standard deviation to create a gaussian model of heart rate and respiratory rate. For each driving condition, the heart rate Hr and respiratory rate Rr is sampled from the gaussian model. The respiratory quotient can be computed as $Rq = \frac{Hr}{Rr}$. The human reaction time t_r can be computed using the following equation:

$$t_r = 0.2 * Rq$$

Project Tasks

The project has three specific tasks as shown in Fig 2.

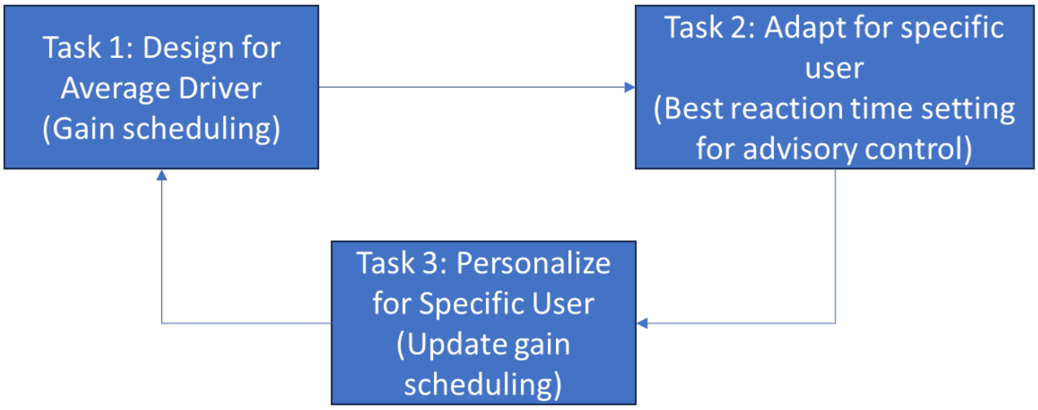

Fig 2: Project tasks

In this project, we will first attempt to design the advisory control for an average user. Then choose a specific user and modify the advisory control design at deployment. Finally, after deployment revert back to the design phase to change the advisory control and personalize it for the specific user.

This mimics the standard context aware application development workflow. Here the manufacturer initially designs a system to be used for a large set of users. After deployment, a specific user can make some limited configuration changes to the system. The data is collected during deployment. This data is then used to redesign the system to personalize for specific user profiles.

Task 1: Controller design

Given:

1. **A set of initial speeds, 20 kmph to 40 kmph**
2. **Deceleration limits of normal (-200 m/s^2) and poor (-150 m/s^2)**
3. **User profiles**

User	**LCW**		**HCW**	
	HR	**RR**	**HR**	**RR**
1	**80 (± 14)**	**16 (± 6)**	**95 (± 26)**	**21 (± 14)**

2	65 (± 15)	13 (± 4)	71 (± 21)	14 (± 5)
3	61 (± 14)	17 (± 8)	92 (± 23)	26 (± 16)

To do:

1: Create an average user profile

2. Determine reaction time model

3. Determine controller gain for each road condition such that collision is minimized for the given initial speed range

4. Determine the number of switches necessary to avoid collision for your design

Task 2: Reaction time update in advisory control

Given:

1. **User 3 as the actual user of the system**
2. **Initial speed range 20 kmph to 60 kmph**
3. **Fix the gains for Normal and Poor road condition from Task 1.**
4. **Markov chain model for road condition generation**

To do:

1. **Using the monte Carlo Markov chain simulation method generate a sequence of length 100 of normal and poor road conditions**
2. **For each road condition in the sequence, sample the initial speed range uniformly to get one initial speed**
3. **For each road condition in the sequence, sample the HR and RR models of User 3 to obtain one HR, one RR and consequently one reaction time for the user.**
4. **Evaluate collision status for each road condition in the sequence.**
5. **Evaluate the number of switches in each road condition in the sequence.**
6. **Determine the reaction time setting in the advisory control that gives the least number of collisions with the least number of switches.**

Note here that the reaction time of the human user is independent of the reaction time setting of the controller.

So there will be scenarios where even if you decide to switch according to the current reaction time of the human driver it is not feasible for the user to stop the car and as a result there is a collision.

Task 3: Gain personalization

Given:

1. **User 3 as the actual user of the system**
2. **Fixed reaction time setting of advisory control**
3. **Initial speed range of 20 to 60 kmph**

To do:

Update the gains for each road condition such that the number of collisions and number of switches are both reduced.