Modeling and Verification of an Automatic Transmission System

Aninda Manocha and Filip Mazurek

May 4, 2018

1 Introduction

Automobiles have had a significant and remarkable impact on society ever since they were first invented. They are a convenient and widely used method of transportation, and have it made travel throughout the world much easier. Given their ubiquity, it is useful to understand how the many constituents inside them operate. The transmission is a vital component of a vehicle in that it is composed of hundreds of pieces that work together to allow the vehicle to move. There are many gears that ensure that the appropriate amount of torque is delivered to the vehicle wheels so that the vehicle moves at the desired speed. Some vehicles operate with a manual transmission system, where the driver engages with a clutch that controls the connection between the engine and the transmission. A majority of vehicles in the United States have an automatic transmission system, where gears are shifted automatically. These transmission systems are critical to model properly and ensure their safety because if a vehicle malfunctions, it can be dangerous or even fatal for the driver or passengers. Due to the popularity of automatic transmissions in addition to the fact that manual transmissions are more mechanical, we decided that an automatic transmission is more interesting to model and focused solely on this type of transmission system.

Over the past two decades, researchers have been using timed (finite) automata to model the behavior of real-time systems because it is a useful and sophisticated method of modeling and verifying systems that are event-triggered. An automatic transmission system contains multiple components that each have individual functions and depend on one another to carry out the functions of the system as a whole. Each completion of a function signifies an event that triggers another event. For example, a gear shift follows the opening and closing of clutch packs. The opening and closing of clutch packs constitute events and these events trigger the shift of a gear, so a state in which a gear has been shifted is an example of an event-trigger state. Since there are many event-triggered states of a transmission system, the logic behind its functionality can be appropriately modeled with a timed automata model.

In this paper, we propose a timed automata representation of a 4-speed automatic transmission system. Our model is a generic example of a four forward gear, one reverse gear automatic transmission. We construct and verify the model using Uppaal, an integrated tool environment for modeling, validating, and verifying real-time systems that have been modeled as timed automata. Due to the nature of the software and its inability to capture the dynamics of a physical system, the model is an approximation of the behavior of the physical components. Therefore, we also describe a simulation of the dynamic physical system in Simulink, a graphical programming environment for modeling, simulating, and analyzing multidomain dynamical systems. Our simulation corresponds directly to our timed automata model. While it captures the physical dynamics of an automatic transmission system, our timed automata model illustrates all possible states within the entire

system and allows us to verify that the system functions properly. Together, our models represent all aspects of interest in the entire system.

We have organized our paper as follows: In Section 2, we introduce the purpose, components, and functions of the 4-speed automatic transmission system. In Section 3, we describe the collection of Uppaal models of the components of an automatic transmission system used to model the logic of the system as a whole. In Section 4, we define the safety constraints placed on the automatic transmission system model that allow it to function well. In Section 5, we note the edge case behaviors that can occur within an automatic transmission system and how our model handles them. In Section 6, we explain recoverable errors, when they could occur in our system, and how we have designed our system to respond to them. In Section 7, we verify that our model functions properly and well by performing a series of queries about the system and ensuring that they are met. In Section 8, we discuss the difficulty with modeling a physical system in Uppaal and the adjustments we made to make our approximation as accurate and representative as possible. Lastly in Section 9, we detail how we simulate our physical model in Simulink in order to demonstrate the dynamics of the automatic transmission system we have modeled.

2 The Automatic Transmission System

Gears in a vehicle allow it to adjust power delivery from the engine to the wheels, therefore adjusting the ratio between the engine speed and the wheel speed. Consequently, they make more effective use of the engine's torque while the engine continues to operate at an appropriate speed. The transmission allows cars to have more than one gear ratio. A manual transmission system locks and unlocks different sets of gears to the output shaft in order to achieve different gear ratios, so the primary purpose of an automatic transmission system is to change the gear ratios in a vehicle without the need for a human driver to interact with the gear shift. It has the same set of gears as a manual transmission in order to produce different gear ratios and the best gear is determined by the gear controller, which will place the transmission in the selected gear. Several safety and performance checks are in place in order to ensure good shifting for the driver and vehicle.

The transmission also has user-selected modes. The one reverse gear of the transmission may be selected by the driver in order to make the transmission spin in the reverse direction, therefore making the engine power the car into reverse movement. The transmission may be placed into neutral, which will essentially decouple the engine from the transmission since no clutch packs will be closed which will allow the engine to deliver power to the rest of the transmission.

Placing the vehicle in Park is a slightly different move altogether. Rather than a specific combination of closed clutch packs, it is a physical pin which slips into place on a part of the transmission, preventing it from moving at all.

There are other forward drive modes which may be included on some models of cars. Some cars have the option to put the transmission into a gear where the transmission will not be allowed to shift into the overdrive gear. This is designed for cases where the car needs more power delivery, such as when it is going uphill. Other modes may include a pseudo manual mode, where the user is allowed to shift up or down through the gears manually–but the TCU still protects the engine and transmission by not allowing the user to upshift if doing so woudl stall the engine or downshift if doing so would force the engine to jump to very high RPMs.

2.1 Engine

The engine provides the power necessary to move the car. In this case for modeling and simulating, we assume that it is an internal combustion engine which delivers different amounts of power based on the demand by the driver. The driver may demand power by depressing the throttle in the vehicle, which increases fuel delivery to the engine. In our study, we keep the engine as a general entity.

We maintain some assumptions in order to make the modeling and the simulation simple. We are assuming an internal combustion engine which idles when not in use (i.e. there is no auto shut off when the vehicle is stopped). Furthermore, we assume that the engine has some minimum idle speed and has some maximum rotation speed. These values will be within those for small consumer vehicles.

2.2 Torque Converter

The torque converter replaces the clutch, which exists in manual transmission cars and separates the engine from the transmission. It instead allows power to be transferred from the engine to the transmission by connecting the two. The engine can still turn even when the wheels and gears in the transmission are stopped because the converter is fluid coupling and allows the engine to spin independently. When the engine turns slowly, there is a small resulting amount of torque and only light brake pressure is needed to keep the vehicle still. As more pressure is applied to the throttle, more torque is transmitted to the wheels, and the engine speeds up and pumps more fluid into the converter.

The torque converter itself is made up of three parts: the impeller, the turbine, and the stator. There is also transmission fluid that aids in the mechanics of the converter. In general, the converter functions by having the engine spin the impeller, which is a centrifugal pump. As the impeller spins, its fins cause the fluid to be flung to the outside of the pump and towards the turbine. Meanwhile, a vacuum is created and draws fluid in at the center of the impeller so that fluid continues to flow through the pump.

The fluid hits the curved fins of the turbine and changes direction before exiting to the center of the turbine. This directional change allows the turbine to spin, although the impeller always spins slightly faster. The fluid repeatedly exits the turbine at its center and moves in a different direction than when it entered. The turbine's turning energy is applied to the input of the transmission, where the turbine's output shaft is connected. When the transmission spins, the car moves.

Lastly, the stator sits at the center of the torque converter and is responsible for redirecting fluid exiting from the turbine before it reaches the pump again in order to increase efficiency. A one-way clutch connects it to a fixed shaft in the transmission, which causes fluid to change direction when it hits the stator blades and the stator then spins in the direction opposite to the fluid.

This setup using a fluid coupling has several advantages. The most convenient of which is the ability for the transmission to be in gear with the car stopped without the engine stalling. The transmission is, for our purposes, directly connected to the differential and then to the wheels of the car. This means that if the car's brakes are pressed and the car isn't moving, then the stopping force is transferred to the transmission. If the engine were to be directly connected to the transmission, the stopping force would win, therefore stopping the engine. The fluid coupling allows the engine to continue spinning the impeller without moving the turbine.

The major disadvantage of the torque converter is that it always results in energy losses due to the transfer of energy through a fluid medium. This means that even when the engine operates at high speeds, the turbine will always turn a little slower since some energy is lost as waste heat. In order to combat this, most modern torque converters have a lock-up clutch. When the turbine spins at 90% speed of the impeller, a clutch pack

may be closed in order to form a physical connection between the two, resulting in direct energy transfer between the engine and the transmission.

Our model does not include the torque converter itself. Instead, the converter works as a passive system which reacts to input, without active reactions.

2.3 Transmission

The transmission is a set of gears which multiply the torque produced by the engine before putting it to the wheels. The torque may be increased so that it is easier for the car to begin moving, and then later the torque produced may be decreased (while conserving power) so that the engine may stay within its revolution limits as the wheel speed increases.

We are specifically modeling a four forward speed, one reverse gear transmission. Gears 1 and 2 multiply torque so that there is more torque at the wheels than the engine produces. Gear 3 achieves a one to one ratio of engine torque to transmission output. Gear 4 is the overdrive gear, where the transmission output spins faster than the engine, at the cost of torque. The overdrive gear is used mainly to keep engine revolutions and fuel consumption low during high speed driving, such as interstate cruising. The reverse gear reverses the direction of transmission output so that the car may apply power to move backwards.

Contrary to older cars' gears being changed by pressure-gated hydraulic systems, newer cars gears' may be changed by the electronic Transmission Control Unit (TCU) based on some algorithm. The simplest algorithms depend on graphs which map vehicle speed and throttle position, and instruct the transmission to shift when a gear boundary is crossed. More advanced TCU systems may use artificial intelligence to improve shifting–such as preventing upshifts when the car turns while moving uphill so that the driver maintains maximal control, but many of these systems are not yet on the market.

We are modeling a four forward speed transmission with an electronic TCU which changes gears based on throttle position and vehicle speed. This model was decided upon thanks to its relative simplicity, understandability, and ease to extend with more gears or different TCU gear change schemes.

2.3.1 Planetary Gears and the Compound Planetary Gearset

The different gearings are achieved by two connected planetary gearsets. Planetary gears consist of a central sun gear, planetary gears around it connected by a planet carrier, and a ring on the outside. A single planet carrier may obtain two forward gears and one reverse gear. We achieve four forward gears by using a compound planetary gearset, where it appears like a single planetary gearset, but behaves like two planetary gearsets combined. There is one ring gear that serves as the output of the transmission, two sun gears and two sets of planets, and a planet carrier inside. The first planetary gearset's planet carrier is connected to the second, output planet carrier's outside ring. The gear ratios are based on the ratio of the number of teeth on the ring gear to the number of teeth on the sun gear.

Different transmission systems are possible. In our model and simulation, only the transmission implementation would need to be replaced in order to create other types of systems. The other components may remain in place.

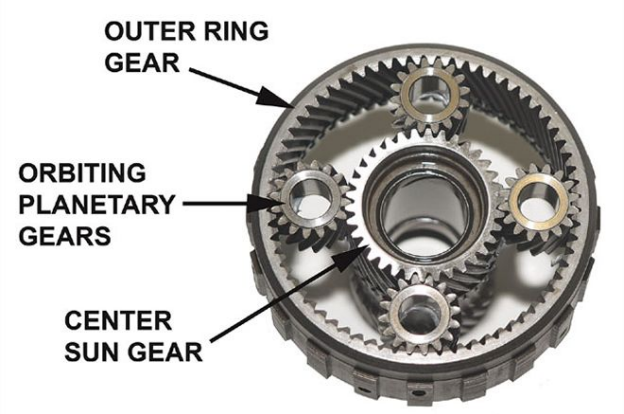

Figure 1: A planetary gearset.

2.3.2 Clutch Packs

Gears changes are actually achieved by changing the input location of the power to the compound planetary gearset and changing whether some components are allowed to move relative to one another. Clutches may be closed (put into contact) or opened (allowing the two components to rotate independently of each other. Signals are sent out to the clutches instructing them to close or open. This is taken into account both in our model of physical clutches and in our Simulink simulation of the transmission. A clutch can also be used for the lock-up clutch in the torque converter, as described above.

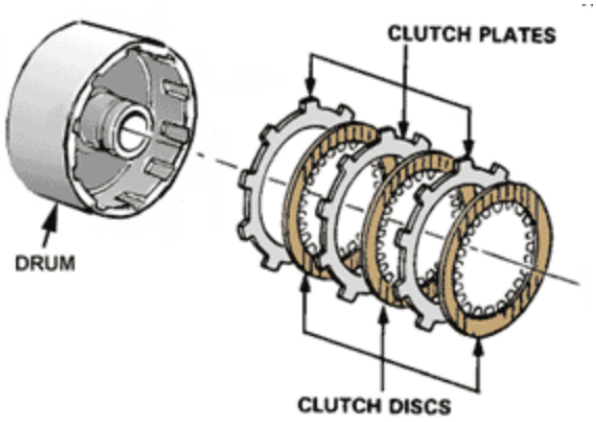

Figure 2: Clutch packs in an automatic transmission system.

2.4 Transmission Control Unit (TCU)

The Transmission Control Unit (TCU) is the most important logical part of the system. This controller monitors a variety of factors around the vehicle. It could be as simple as only reading the vehicle speed and the throttle position to apply the most simple shift mapping, or it may also sense data of how far the car is turning, whether it is driving at an incline, and other more specific data, and then use all that and artificial intelligence in order to decide on a more optimal shifting schedule.

We are modeling the TCU which decides when the vehicle should shift based on a map of shifts. The TCU will give a signal to shift whenever the combined information of the throttle and vehicle speed move over the gearing boundary. The following is the shift map we use in our simulation.

The shift map which we use is one of the included examples given by Mathworks. It was determined to be an effective shift map for one of the engine and transmission profiles that were analyzed. Our model is more concerned with being a general model rather than an optimal one, so we opted for a shift map that was proven correct for about the same engine and transmission profile connected to about the same vehicle body.

It is easy for anyone using the model or simulation to change the shift map. Alternately, this general map allows for the TCU logic to be swapped for any other sort of logic. Our model and simulation have a number of available parameters, but others, such as turning angle, may be added.

3 System Modeling (Uppaal)

Our collection of Uppaal models seeks to best represent the digital logic used in the TCU and how that data is read. At the same time, our models represent the environment (i.e. the driver and all user control over the car) as almost totally random variables. This randomness allows us to verify the model for every possible state that it may be in since user input is not predictable from the viewpoint of a timed automata model.

This model comes with pre-set values for certain parameters. The values are in terms of relative time units, but we define them to be in terms of milliseconds. This way, we can set parameters such as the minimum time requirement for switching gears and the time for how long the TCU should check that the transmission may shift. These parameters should be set to the same values in the simulation. Anytime these parameters are changed (e.g. to better shifting performance by reducing the maximum time in between shifts) the model must be re-verified before it is updated in the simulation.

3.1 Physical System

Modeling the physical side of the transmission system is much trickier to work with in Uppaal. User inputs are values or components that the user can control. In our model, we are concerned with the throttle position and the brake position. Both the throttle and brake are modeled as variables in this system. Vehicle speed was also chosen to be a standalone variable. This was due to the fact that the relationship between throttle, brake, and vehicle speed is very complicated and depends on many factors such as engine power, wheel slippage, and incline of the road on which the vehicle is driving. Instead, by allowing speed to be modeled as a more independent variable, we are allowing the system to react to whatever speed may be set at any time. Furthermore, our shift maps are based on vehicle speed and throttle position. Because speed is a separate variable, we may independently test all combination possibilities for shifting.

There are two possibilities for running the model:

1. We may string together a series of states with certain actions occurring at specific time points. This would allow us to ensure that the behavior of the system is as expected based on our assumptions and the given exterior conditions. This sort of model is better for a sort of manual sanity check, if one is needed.
2. We may allow the throttle, brake, and speed to randomly change. This is accomplished by single state models for each of the variables, where paths taken will either increase or decrease the variable. This is

a better method for verification of all properties because the values chosen are random and the verifier is able to find any and all possible state combinations.

In order to make the physical system more realistic, we made the physical components take time to occur. The clutch packs which control the connections in the transmission close due to pressure of a fluid. Therefore, we added parameter requirements. One parameter is the minimum time that a clutch takes to open. This is necessary so that we are sure that the system we model doesn't occur any more quickly than is physically possible. We also include a maximum time constraint parameter: this parameter is meant to check whether clutch closing takes longer than is anticipated. In this case, we force the model into an error state. These timing parameters are all available to be changed. It is necessary that if performance enhancements are considered (such as quicker shift times) that firstly the physical clutches themselves are verified to have a shorter closing time and that secondly the model is verified with the newly decided times.

3.1.1 Clutch Packs

In our system, there are five different clutch packs: A, B, C, D, and R. For each of the five possible gear states we have modeled (R, 1, 2, 3, and 4), two clutch packs are closed and three are open. The clutch packs that close depend on the current gear. Below is the clutch schedule based on the gear:

Clutch Schedule

Gear	A	B	C	D	R	Ratio
R	0	0	0	1	1	-g1
1	1	0	0	1	0	g2
2	1	0	1	0	0	(g1+g2)/(1+g1)
3	1	1	0	0	0	1
4	0	1	1	0	0	g1 /(1+g1)

0 - Disengaged, 1 - Engaged

g1: Input planetary ring/sun gear ratio
g2: Output planetary ring/sun gear ratio

Figure 3: The schedule in our modeled automatic transmission system.

We decided to treat each clutch pack as an individual model. Since a clutch pack can either be closed (engaged) or open (disengaged), these were our two main states of interest. However, a clutch pack is a physical component that cannot change its state instantaneously. It is a process to open or close a clutch pack so in order to illustrate this process, we added two additional states "closing" and "opening". The clutch pack can then follow a cycle of "closed" to "opening" to "open" to "closing" to "closed". If the clutch pack is in either the "closed" or "closing" state and it receives a signal that the TCU wants to open that clutch, then its state changes to "opening". The same idea applies when the clutch pack is either "open" or "opening" and receives a signal to close.

In order to model the minimum amount of time required for a clutch pack to open or close, we created two time constants CLOSE_MIN and OPEN_MIN to represent the minimum amount of time each operation takes to complete. Additionally for each state we created two time constants CLOSE_MAX and OPEN_MAX to represent the maximum amount of time each operation should take. If the operation takes longer than the maximum expected time, it must be malfunctioning and the clutch pack would fall into an error state. The maximum time constants serve as invariants on the "closing" and "opening" states because the clutch pack should no longer be trying to close or open after the defined maximum amount of time has passed. On the other hand, once the minimum amount of time has passed that the clutch has spent opening or closing, the clutch can exit from "opening" or "closing" state. The clutch model then transitions into an urgent state, where no time can pass. This state is used to evaluate how long the clutch spent opening or closing by comparing the current time to a constant CLOSE_SPEC or OPEN_SPEC (time constants that have been listed in the transmission specifications as amounts that flag an error). If the time is less than the specification amount, then the clutch model can proceed to the "closed" or "open state. Otherwise, it falls into an error state. The time is reset to be 0 every time the clutch receives a signal to change into the opposite state of engagement.

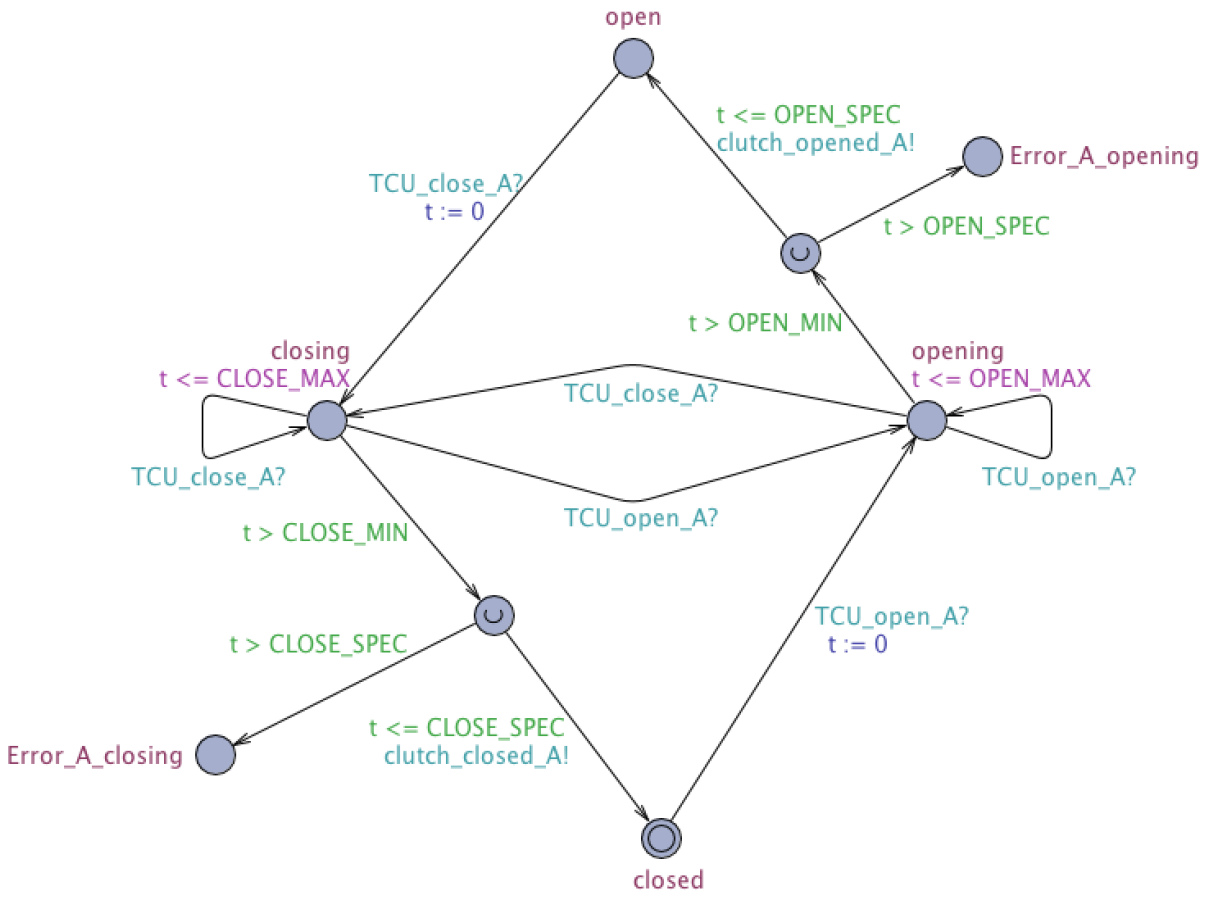

Figure 4: A model of one of our clutchpacks (A) in Uppaal. There are four others that look identical to this one in structure, but have different state, signal, and guard names based on the clutch.

We used constants to represent the times used for dynamics modeling and error checking because we had set

their values arbitrarily with some guidance from the models in [1]. By using constants, we create placeholders for the values that are changeable and tunable so that this model can be applied to specific transmission systems.

Another clutch that we would have liked to include in our system is the lockup clutch. However, our system model did not include the engine and torque converter, so it would be difficult to model the signals that trigger the engagement and disengagement of this clutch. The lockup clutch would have a similar appearance to the rest of the clutch models because it operates like a normal clutch that opens and closes. However, there would not be signals coming from the TCU, but from some other microcontroller.

3.2 Digital System

Uppaal best models digital systems and software interaction. The discrete states in Uppaal models correspond to the discrete possibilities of values in a computer program. The TCU is the software part which observes all variables and then gives the proper shift signal based on what it detects.

Our TCU decides when to shift between the forward gears based on how depressed the throttle is (percentage) and the vehicle speed. The values are compared to a matricized version of the shift map. Additionally, our TCU is split among several models, but purely for the sake of compartmentalizing the models. The whole TCU was much too large to fit in a single model and still be manageable.

3.2.1 TCU Mode

Our TCU model is made to function exclusively while the vehicle is placed in the default forward driving mode which uses all gears ("Drive"). This is the upshift and downshift logic part. The initial state is kept under control of its invariants, meaning that if there is a possibility for the transmission to shift gears based on its shift map, then it definitely should shift. Once the TCU detects the possibility of a shift, it enters a separate state which continually makes sure that the vehicle meets the condition to shift over a period of specified time CHANGE_CHECK. This is a check made in order to filter out some noise of the throttle being momentarily pressed or the speed briefly changing. This greatly increases the comfort of the driver, since the car is not continually and unnecessarily changing gears.

If the TCU decides it should shift, it immediately starts a timer in order to make sure that the shift will take place within SHIFT_TIME. Otherwise it will enter an error state. The *upshift!* or *downshift!* essentially calls a subroutine which carries out the call.

The initial state in this model is made to only function while the vehicle is placed in a forward driving mode by checking that the vehicle is at least placed in a forward moving gear.

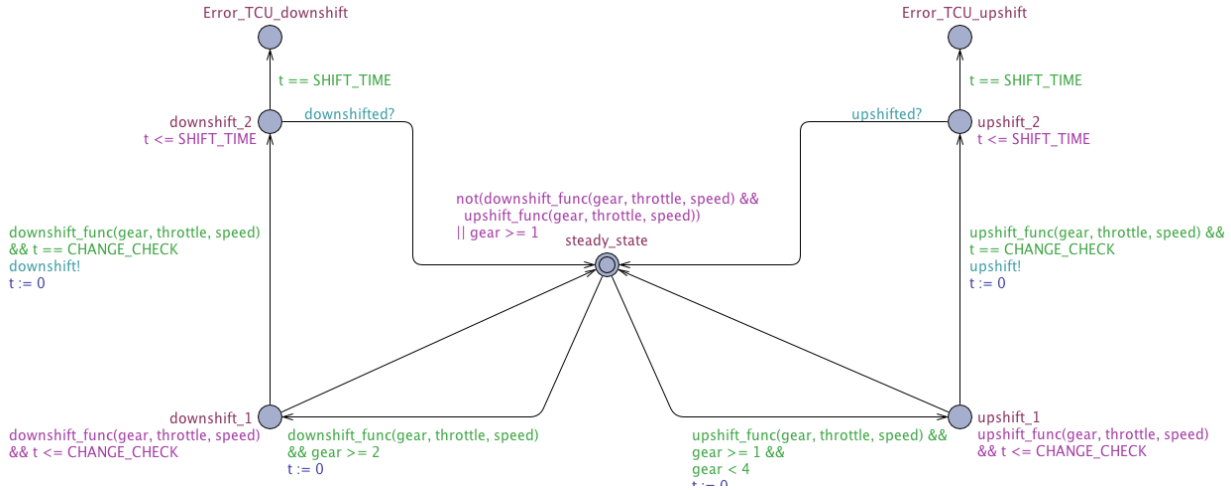

Figure 5: A model of the TCU mode in Uppaal.

3.2.2 Gear Shift Controller

The next part of the TCU is where it keeps track of each of the individual gears into which the vehicle may be placed. It is necessary to keep track of all the individual gears because each specific gear requires its own clutch pack mapping in the transmission system. This model serves to keep track of the gear and to call more specific clutch changes. It has its own timing checks with CONTROLLER_WAIT. This check is here in case of any failure to communicate with the subroutines which will be calling the clutches themselves.

It is important to note here that this part of the TCU assumes that the vehicle begins in gear and possibly driving. The Committed state is purely for setup purposes.

Here the important interface for user-mediated gear changes is also located. The TCU lets the physical system know whenever it is ready to switch gears from forward driving to reverse. The conditions for doing so are described in the safety conditions section.

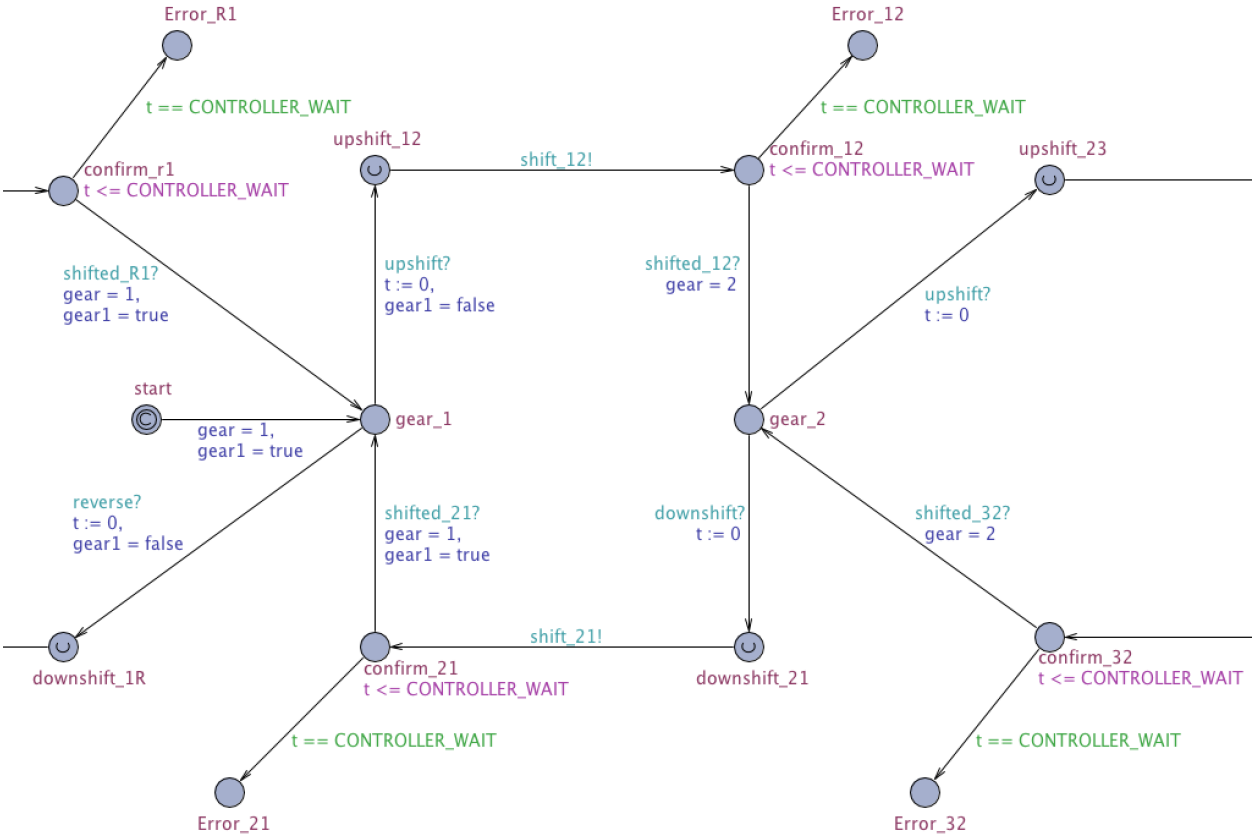

Figure 6: A part of the model of the TCU controller in Uppaal that highlights the logic for the specific gear shifts between first and second gear. The other gear shifts are carried out in the same fashion.

3.2.3 Gear Shifts

The part of the TCU which actually signals the physical clutches to change is made up of a model for every possible gear change combination in our system. We modeled each possible gear shift with an individual model in order to show the specific clutch actions that must take place for the gear shift to successfully occur.

Every change requires a specific clutch to be opened and another clutch to be closed. The states during which the clutches are signaled to begin making their changes are urgent, in order to model the fact that properly working code will send out messages to both of the clutches at essentially the same time. The timer then begins as soon as both instructions to change are sent out. The CLUTCH_RESPONSE_TIME waits is the maximal time the TCU can wait for the clutches to send out success signals. Otherwise if the time is reached, then the TCU will throw an error since the physical clutches wouldn't change when they were told to change. In our model, we have an error state to signify the error occurring.

The normal state for all of these models is a "ready" state where the model waits before receiving a signal to shift. Once it receives this signal, then it transitions to the "open" and "close" states, which we have created to be practically simultaneous.

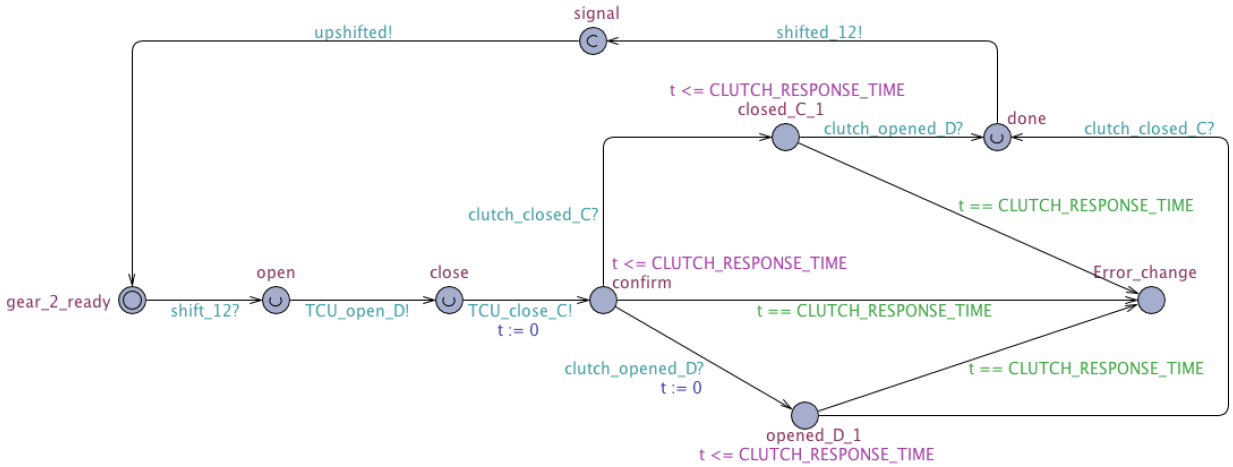

Figure 7: A model of one of our gear shifts (first to second) in Uppaal. There are many other possible shifts in our system that look identical to this one in structure, but have different state, signal, and guard names based on the gear shift.

4 Safety Constraints

There are several constraints in order to make the automatic transmission function well. Below is a comprehensive list of safety conditions:

1. The vehicle speed may not exceed x under acceleration.
   - This protects the driver from unsafe conditions. Vehicle tires lose grip at high speeds. Driver reflexes are not enough to react at these speeds.
   - To prevent this, the system will ignore throttle input if vehicle speed exceeds x.
2. Engine RPM cannot exceed x
   - The engine must be protected against damage. Additionally, if this were to happen when the vehicle is in motion, then the user could lose control of the vehicle.
   - To prevent this, the system cannot downshift if the shift would put the engine above its threshold. Additionally, in cases when the system cannot upshift, the throttle will be ignored until RPM's fall.
3. Engine RPM cannot fall below x
   - If the revolutions fall below a certain threshold, the engine will stall and turn off. This would be a dangerous condition, especially if the vehicle was in motion and the driver lost control.
   - To prevent this, the system will prevent premature upshifts. Additionally, if the engine were to fall below its threshold, the idle mechanism will turn on to keep the revolutions above stall level.
4. Shifting into Reverse while moving forward
   - Shifting into a reverse gear while the vehicle is moving forward, and therefore spinning the transmission into a forward going direction would greatly damage the transmission.

- To prevent this, the system should prevent shifting into a reverse gear from a forward gear or into a forward gear while in a reverse gear while the vehicle is moving more than x mph and with the brake pressed in order to minimize any damage.

5 Edge Case Behavior

There are certain behaviors of the system that if not addressed, can affect the system in an unpredictable manner. Excluding user input, the functionality of our system model should be deterministic in that every behavior has a corresponding expected result. Therefore, we must address all behaviors of the system that could lead to nondeterministic output. Below we list these behaviors and how we designed our system to respond:

1. Throttle and brake are pressed at the same time
   - To prevent this, if the vehicle is in a driving gear (i.e. one of the forward gears or the reverse gear) then the throttle will be ignored. Otherwise, the vehicle will be allowed to rev the engine.
2. Shifting into Drive from Neutral
   - There is the unexpected case of shifting into a forward moving gear while the transmission is in neutral. It should be possible to shift into gear from neutral while the vehicle is in motion–there is nothing dangerous about it. However, in order to prevent any unexpected transmission damage, the TCU will keep track of what gear the vehicle "should be in" based on the current vehicle speed, and then shift into that gear from neutral.

6 Recoverable Errors

A number of scenarios could occur where a part of the system is not functioning properly, but there is a backup function in the event that this is the case. Below we state these possible scenarios and name them as recoverable errors because the malfunction indicates an error, but the backup functionality makes the error recoverable. A recoverable error occurs when the system is:

1. In a certain gear and want to upshift or downshift, but the next gear is broken.
   - In these cases, the TCU should recognize a gear is broken and let the user know. However, the vehicle should remain functional for the remainder of the drive. In doing so, the TCU should begin to correctly open and close clutches so that the transmission may shift over a gear. For example, the transmission could shift from gear 1 to gear 3 at the gear 2-3 shift in the shift map if gear 2 was broken.

7 Model Verification

Uppaal allows for advanced model verification. In order to ensure that our model would work correctly, we made sure to run the verifier. We ensured safety of the model by checking that the model will never deadlock, and it will never enter any of the error states which we specified above, given the current assumptions of the

model. There are error states for the clutch packs, gear changes, and different states of the TCU controller. We also ensure liveness and functionality of the model by ensuring that every gear can be reached, and also that it is possible to get to a gear from any other adjacent gear. Since it is possible to get to any gear from any other gear, the model is guaranteed to function.

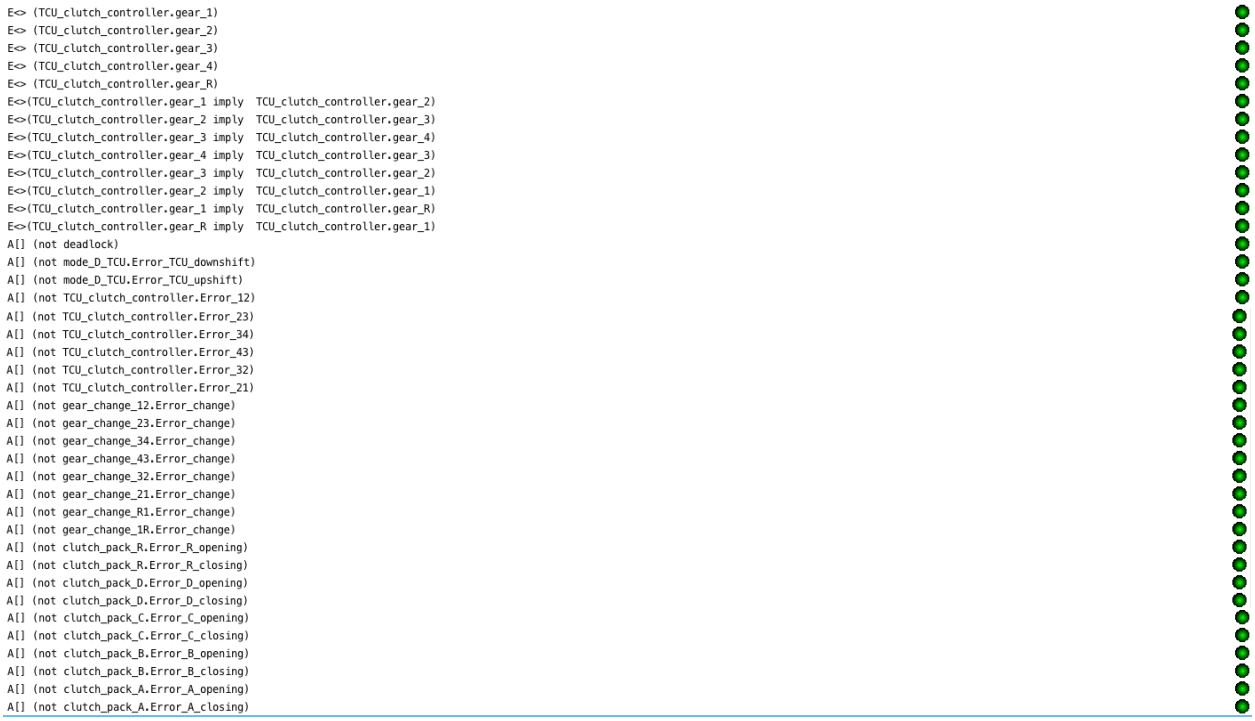

Figure 8: The series of queries we entered into the verifier in Uppaal to ensure that our model was live and safe, and that it could never deadlock.

8 Difficulty with Modeling in Uppaal

We chose to model the transmission and the TCU using model verification software Uppaal. Uppaal provides verification that surpasses that of Simulink and Stateflow, so we may ensure that our models are correct before we build simulations of our system.

Uppaal provides accurate model checking capability for the digital system. We built in the logic for our TCU using Uppaal. However, it is more difficult to model physical systems. We made basic models with time checks for the clutch packs in the transmission. Whichever packs are engaged at any one time correspond to the gear that is selected in the transmission.

Therefore, in order to integrate our model with a physical system, we chose to model a car in Mathworks Simulink. We used the Driveline package in order to model the vehicle and Stateflow logic in order to model the TCU. Simulink is able to model the physical system and therefore show that the verified transmission model will work with a specific vehicle simulation.

9 Dynamic System Simulation (Simulink)

Simulink contains the Driveline package, which allows for advanced modeling of lots of rotational forces, especially those found in a vehicle engine, drive shaft, and wheels. Therefore Driveline is perfect for modeling the reaction of a car to different forces and to different inputs.

Simulink is best used for actual simulation test cases, not for general model verification. But here we can check all our general models and edge cases in order to ensure that the model is definitely behaving as expected and that the software was formed exactly to the specifications of the models.

9.1 Input

Everything in Simulink may be used and reused as an input. For user input, we use the Signal Builder block in order to simulate a driver. The driver is free to press the throttle, press the gas, and change gearing from forward driving to reverse. The reactions of the system (shown as the speed of the car) is also fed back into the TCU.

9.2 Transmission

We found that Driveline offers an accurate four forward speed transmission which works using a compound planetary gearset. The model works exactly as described above: the transmission also takes a digital signal of gearing in order to engage the correct clutches.

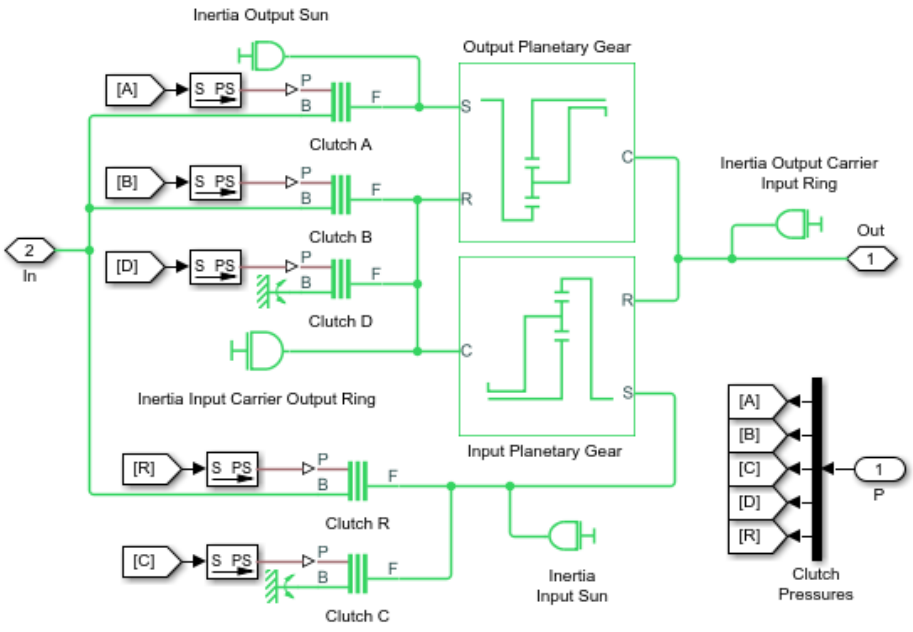

Figure 9: Diagram of a transmission system, where the clutch packs are illustrated with three bold lines.

9.3 Transmission Control Unit

The best way to simulate the TCU logic in Simulink is using Stateflow. Stateflow allows us to model using logical states, similarly to Uppaal. There are several significant differences between Uppaal and Stateflow, but that discussion will not take place here.

9.4 Simulink Results

We unfortunately could not get Simulink to work correctly for our use. Placing the transmission in reverse gear resulted in unexpected results which we could not remedy during the duration of this study. It will need to be investigated in depth at a later point in time.

10 References

1. Magnus Lindahl, Paul Pettersson and Wang Yi. Formal Design and Analysis of a Gear Controller. *In Springer International Journal of Software Tools for Technology Transfer (STTT)*, volume 3, issue 3, pages 353-368, 2001
2. King, Graham & Peter Jones, R & D. Bailey, Andrew. (2003). Application of Systems Modeling and Simulation in the Discrete Ratio Automatic Transmission Calibration Process for an Automobile. 72. 10.1115/IMECE2003-41119.
3. Jiang, Yu & Yang, Yixiao & Liu, Han & Kong, Hui & Gu, Ming & Sun, Jiaguang & Sha, Lui. (2016). From Stateflow Simulation to Verified Implementation: A Verification Approach and A Real-Time Train Controller Design. 1-11. 10.1109/RTAS.2016.7461337.
4. Jiang, Zhihao, et al. Modeling and Verification of a Dual Chamber Implantable Pacemaker. SpringerLink, Springer, Dordrecht, 24 Mar. 2012, link.springer.com/chapter/10.1007/978-3-642-28756-5_14.
5. Nice, Karim. "How Automatic Transmissions Work." HowStuffWorks, HowStuffWorks, 29 Nov. 2000, auto.howstuffworks.com/automatic-transmission.htm.
6. "Vehicle with Four-Speed Transmission." MATLAB & Simulink, Mathworks, www.mathworks.com/help/physmod/sdl/e with-four-speed-transmission.html.
7. "4-Speed CR-CR." MATLAB & Simulink, www.mathworks.com/help/physmod/sdl/ref/4speedcrcr.html.
8. Alur, Rajeev, and David L Dill. "A Theory of Timed Automata." Studies in Computational Intelligence Advances in Verification of Time Petri Nets and Timed Automata, 2006, pp. 29–49., doi:10.1007/978-3-540-32870-4_2.

11 Images

1. `http://image.fourwheeler.com/f/31033630+w660+h440+q80+re0+cr1+ar0/0706or_19_z%2bautomatic_transmission%2btypical_planetary_gearset.jpg`
2. `http://www.gottransmissions.com/blog/wp-content/uploads/2009/05/trans_clutchpack.gif`
3. `https://www.mathworks.com/help/examples/sdl_product/win64/sdl_transmission_4spd_crcr_02.png`