CONTROL SYSTEM DESIGN OF CONFIGURABLE HYBRID ELECTRIC MODULES

Shyam Sundar Pradhan

Department of Mechanical Engineering, Michigan Technological University, 1400 Townsend Dr, Houghton, MI 49931, USA

Abstract - *This paper presents a controller design of "Configurable Hybrid Electric Learning Module" using Stateflow with the help of model-based embedded control design approach including six sub-models and a stepper motor driving model. This design provides control logic and input/output signals of each sub-model.*

***Key Words*: Hybrid Electric Learning Module, Stateflow, Model based embedded control design.**

1. INTRODUCTION

The control of Hybrid Electric Vehicle is an elaborate task that involves lots of parameters and variables. It is particularly difficult to control HEVs since two types of power sources are involved and for the efficient and robust operation, right amount of power has to be drawn from each one of them at any instant.

The controller module that I am going to design has to handle the engine and the motor appropriately and also has to optimally distribute demanded power among them, so that the vehicle can operate efficiently. The controller should also consider the limitations of CHELM, so that it could be smoothly implemented on it.

The control Module will take inputs from the Sensor Module and process it to generate appropriate signals through Actuator Module, for controlling the engine and the e-motor. The inputs and outputs of the Control Module and its interface with the Sensor and the Actuator Modules has been shown in figure 1.

Stateflow [1] is the most popular control logic tool within a Simulink model and it is used for model reactive systems with state machines and flow charts. Generally, it is used in discrete hybrid control system.

2. CONTROL MODEL DESIGN

In my control system, input signals are Motor On Switch, Engine On Switch, Forward Switch, Reverse Switch, Crank Switch, Engine Average RPM, Vehicle Speed, APP (Accelerator Pedal Position), TPS (Throttle Position Sensor) and the output signals are E-Motor Torque Request, Forward, Reverse, E-Motor On, Engine Crank, Engine Kill, Coils (4).

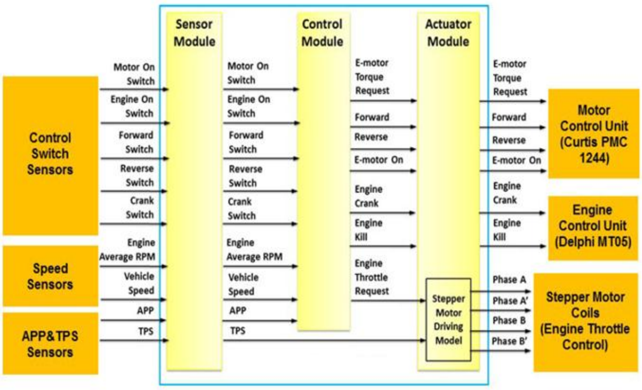

Fig. 1: Overview of the control system

The control module has six subsystems to perform different tasks such as changing driving mode, engine state, computing engine and e-motor blending factors, appropriately distributing requested torque etc. These subsystems communicate with each other to execute the control logic. The signal flow among these subsystems are shown in Stateflow in figure 2. To differentiate different kind of signals, input signals are colored green, output signals are colored red and intermediate signals are colored blue.

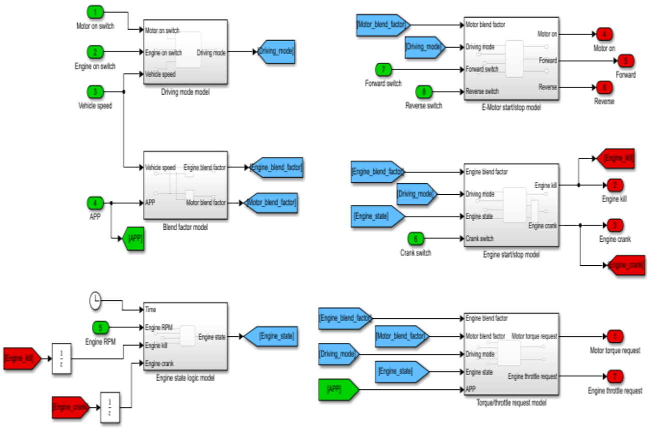

Fig. 2: Sub-systems and signal inside the control Module

Each of the sub-systems are discussed in details below-

2.1 Driving Mode Model

This model has three driving modes: Electric Solo Mode, Blending Mode and Engine Solo Mode. In the electric solo mode, the E-motor is the only source providing driving force for the vehicle. In the blending mode, both engine and motor provide driving force for the vehicle. In the engine solo mode, the Engine is the only source providing driving force for the vehicle. It has three inputs: Motor on Switch, Engine on Switch and Vehicle speed. The vehicle can switch from one driving mode to another only when the vehicle speed is less than 1 mph.

The logic between the output signal and input signal listed in Table 1 below

Table 1: Determine driving mode based on switch signal

Inputs			**Driving Mode Output**		
Engine on Switch	Motor on Switch	Vehicle Speed	Electric Solo Mode (1)	Blending Mode (2)	Engine Solo Mode (3)
True	False	v<1 mph	False	False	True
False	True		True	False	False
False	False		False	True	False
True	True		Driving Mode will stay unchanged		
		v>1 mph	Driving Mode will stay unchanged		

Driving Mode is an important intermediate signal which is used in Engine start/stop logic model, E-Motor start/stop logic model and torque/throttle request logic model.

The above logic for driving mode model is implemented in State Flow diagram (figure 3). First, I design two exclusive (OR) states that represent mutually exclusive operation modes. One Exclusive state (Fixed Mode) will be active when V>1 mph, in that moment driving mode will not change. Other Exclusive state (changing mode) will be active when V<1 mph. Inside this changing mode state, it consists three Exclusive states to represent electric solo mode (1), blending mode (2) and engine solo mode (3). In changing mode state, I start the vehicle in electric solo mode that's why I am using it as a default state. Connective junction is used as a decision point between alternate transition paths for a single transition.

2.2 Blend Factor Model

Blend factor is a very important variable when the vehicle is operating in 'Blending Mode'. It helps in efficiently distributing the desired power between engine and motor. Two types of blending factors are calculated inside this block, first one is 'Engine Blend Factor' and the second is 'E-motor Blend factor'. These two blend factors are further used in Torque/Throttle Request model to calculate the torque/throttle request from the respective power sources. Both the factors depend on vehicle speed and accelerator pedal position.

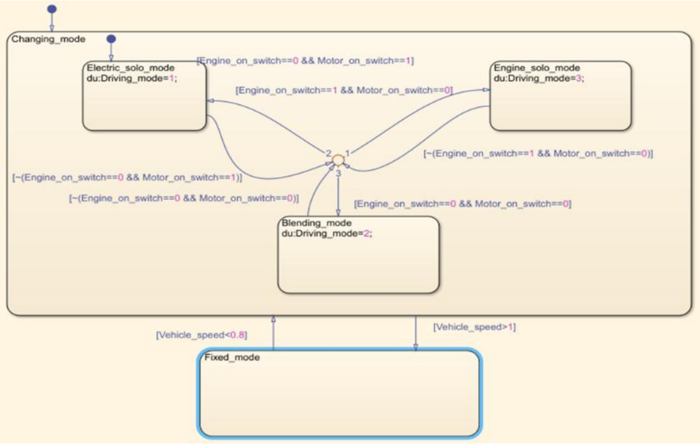

Fig. 3: State flow model of driving mode

In my model, I have used two 2-D look-up table for Engine blend factor and Motor blend factor. For each 2-D look-up table every factor depends on vehicle speed and Accelerator Pedal Position. Figure 4 shows the inside view of Engine and Motor blend factor model.

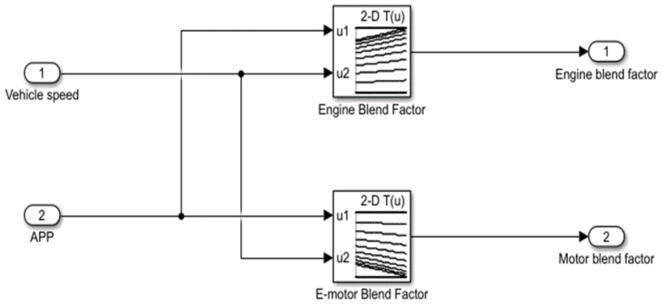

Fig. 4: Blend factor look-up table model

2.3 Engine State Logic Model

This model determines the state of the engine based on engine RPM, a system clock, the engine crank signal and the engine kill signal. There are five engine states: Engine off State (1), Engine Crank State (2), Engine Warm-up State (3), Engine on State (4), and Engine Start Fail State (5). The values of the Engine State output signal corresponding to these five states are 1, 2, 3, 4, and 5, respectively. The conditions of the state transitions are listed in Table 2. Engine Crank Time and Engine Warm-up Time are the time duration of the Engine Crank State and the Engine Warm-up State. A "T (sec)" block can be used to count the time duration of these two states. The output of Engine State is an intermediate signal, which will be used in engine start/stop logic model and throttle request logic model.

Table 2: Conditions of engine state transition

Current state	Condition	Next state (Output Value)
Engine Off State (1) (default state)	Engine Crank=True Engine Kill=False Delay 1 sec	Engine Crank State (2)
Engine Crank State (2)	Engine RPM>800 and Engine Crank Time>1.6 sec.	Engine Warm-up State (3)
	Engine Crank time>2 sec. and Engine RPM≤800	Engine Start Fail State (5)
Engine Warm-up State (3)	Engine Warm-up time>3 sec. and Engine RPM≥500	Engine On State (4)
	Engine Warm-up time>3 sec. and Engine RPM<500	Engine Start Fail State (5)
Engine On State (4)	Engine RPM<50 or Engine Kill=True	Engine Off State (1)
Engine Start Fail State (5)	Delay 1 sec	Engine Off State (1)

Base on the engine state logic, state flow diagram is shown in figure 5.

2.4 Torque/Throttle Request Model

Torque/throttle request model determines the percentage of the output power is requested from engine or motor based on blend factors, driving mode, engine state, and APP as shown in figure 2.

When a vehicle is in electric solo mode (driving mode = 1), the engine throttle request is equal to zero. When the vehicle is in engine solo mode (driving mode = 3), the engine throttle request is equal to APP. When the vehicle is in blending mode (driving mode = 2), the engine throttle request is determined by engine blend factor and APP (Engine Throttle Request = APP * Engine Blend Factor).

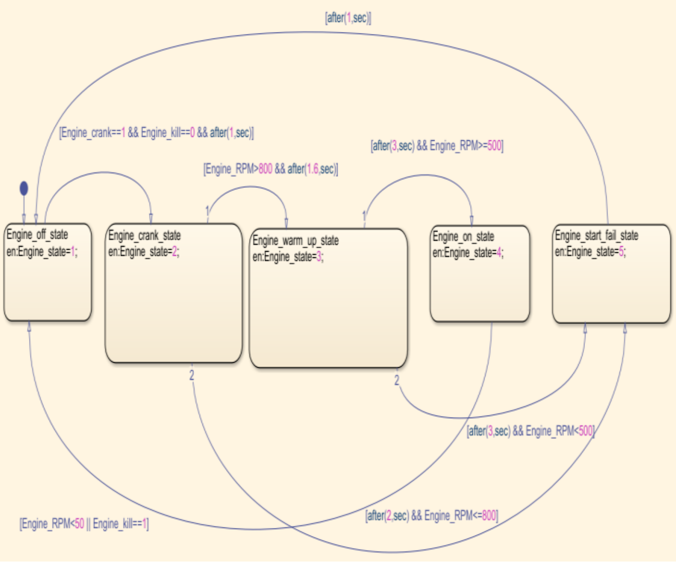

Fig. 5: State flow model of Engine State logic

When the engine is being started, the engine throttle request is a pre-defined value. The impact of engine state on engine throttle request is listed in Table 3 and the control logic for motor torque request is shown in Table 4.

Table 3: Imapact of engine state on engine throttle request

Conditions	Driving Mode	Electric solo mode (1)	Blending mode (2)			Engine solo mode (3)
	Engine State	No	== 2 Or ==3	== 1 Or ==5	Engine On State	No
Engine throttle request		0	7.5	0	APP*(Engine Blend Factor)	APP

Table 4: Imapact of driving mode on motor torque request

Driving Mode	Electric solo mode (1)	Blending mode (2)	Engine solo mode (3)
Motor torque request	APP	APP*(Motor Blend Factor)	0

Based on the above logic, the state flow diagram is drawn in figure 6 and figure 7.

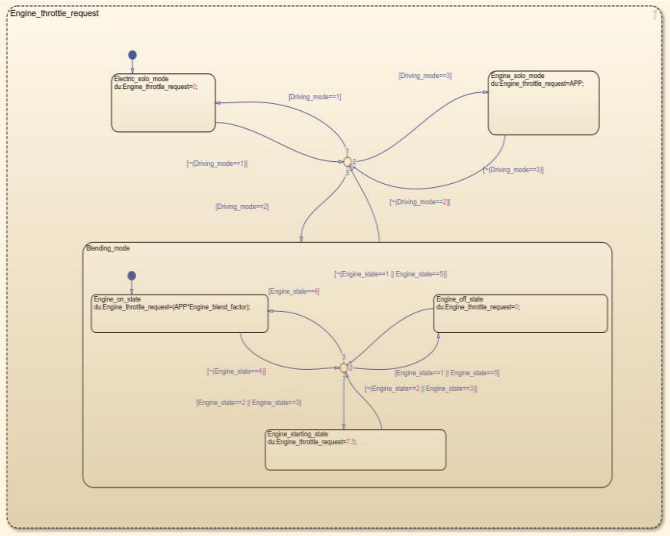

Fig. 6 : State flow model of Engine throttle request

In this diagram (figure 6 and figure 7), I use two Parallel (AND) states that represent graphically by a dashed rectangle with a number indicating execution order. So, the entry state will be engine throttle request according to my diagram and then it will enter motor torque request. Inside each AND state I use three exclusive (OR) states that will execute according to the given logic.

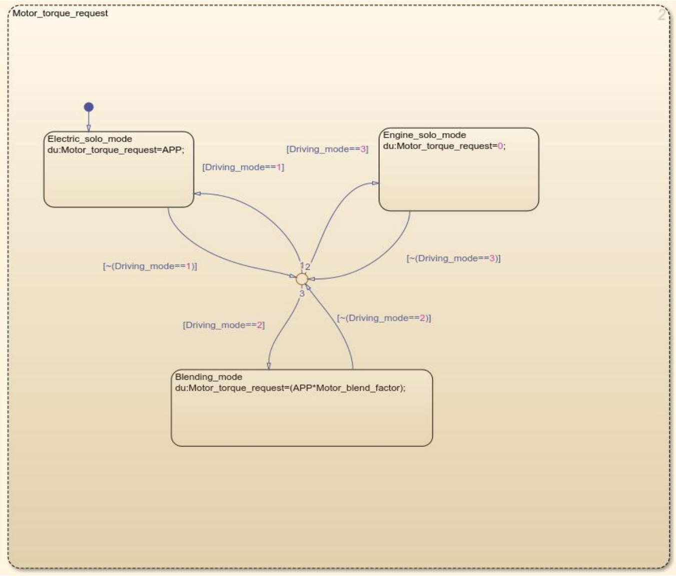

Fig. 7 : State flow model of Motor torque request

2.5 Engine Start /Stop Model

Engine start/stop signals generates whether to start or stop the engine according to the inputs. It has for inputs Engine blend factor, Driving mode, Engine state and Crank switch and has two outputs Engine kill and Engine crank. 'Engine Kill' becomes True when the engine needs to be turned off as it is not required. This signal remains True during the electric operation and remain False during the engine solo mode. While during the blended operation, it becomes True and turns off the engine only when the engine blend factor goes below 0.01 i.e. power request from the engine is negligible. Below table shows input/output relation for this signal.

Table 5: Imapact of driving mode and engine blend factor on engine kill signal

Conditions	Driving Mode	Electric solo mode (1)	Blending mode (2)		Engine solo mode (3)
	Engine Blend Factor	No impact	>0.01	<0.01	No impact
Engine Kill		True	False	True	False

Next output is Engine Crank, and it depends on mainly two signals, Manual Crank and Automatic Crank signal. Logic for manual crank signal is simple, in engine solo mode, it becomes true when the crank switch signal is true and in rest of the modes, it remains false irrespective of other inputs, the logic table for manual crank is given in table VI. Automatic crank signal depends on three inputs and it becomes True only during the blended mode and when the engine blend factor goes above 0.01 while the engine state is still '1' i.e. engine is off. Logic for this signal is given in table VII. Engine crank signal is determined by applying 'OR' operation between manual crank and automatic crank signals.

Table 6: The logic of manual crank signal

Driving Mode	Electric solo mode (1)	Blending mode (2)	Engine solo mode (3)
Manual Crank	False	False	True when the Crank Switch is True

Table 7: The logic of Automatic crank signal

Conditions	Driving Mode	Electric solo mode (1)	Blending mode (2)			Engine solo mode (3)
	Engine Blend Factor	Irrelevant	>0.01		<0.01	Irrelevant
	Engine State	Irrelevant	== 1	==4	Irrelevant	Irrelevant

Automatic Crank	False	True	False	False	False

The overall logic with state flow diagram is illustrated in below figure 8.

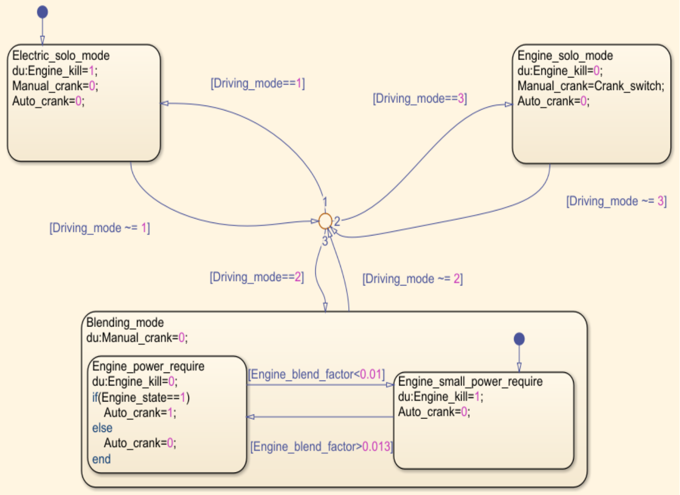

Fig. 8 : State flow model of Engine start/stop

2.6 E-Motor Start /Stop Model

This model determines when to switch the motor on or off and in which direction the motor should move. This model consists with four inputs: E-motor blend factor, Driving mode, Forward switch and Reverse switch and it has three outputs E-motor on, Forward and Reverse.

Table 8: The logic of Motor on signal

Conditions	**Driving Mode**	**Electric solo mode (1)**	**Blending mode (2)**		**Engine solo mode (3)**
	Motor Blend Factor	Irrelevant	>0.01	<0.01	Irrelevant
Motor On		True	True	False	False

The 'Motor On' signal remains True in electric solo operation and becomes False in engine solo operation and in blending mode it turns off the motor whenever the e-motor blend factor goes below 0.01. The logic is shown in Table VIII.

The spinning direction of the motor is determined by the Driving Mode, Forward Switch, and Reverse Switch as shown in Table IX.

Based on the logics which are discussed in table 8 and 9, state flow diagram is depicted in figure 9. In this diagram, I use three exclusive states and keep electric solo mode as a default state. Inside the bending mode, again I use two exclusive states where one state is used for small motor power required and I select it as a default state because during motor start, it requires small amount of torque and it will be active when motor blend factor is less than 0.01.

Table 9: The Forward and Reverse logic

Driving Mode	**Electric solo mode (1)**	**Blending mode (2)**	**Engine solo mode (3)**
Forward Signal	Forward Switch status	True	False
Reverse Signal	Reverse Switch status	False	False

Other state it will use for high motor power require and will be active when motor blend factor is greater than 0.01.

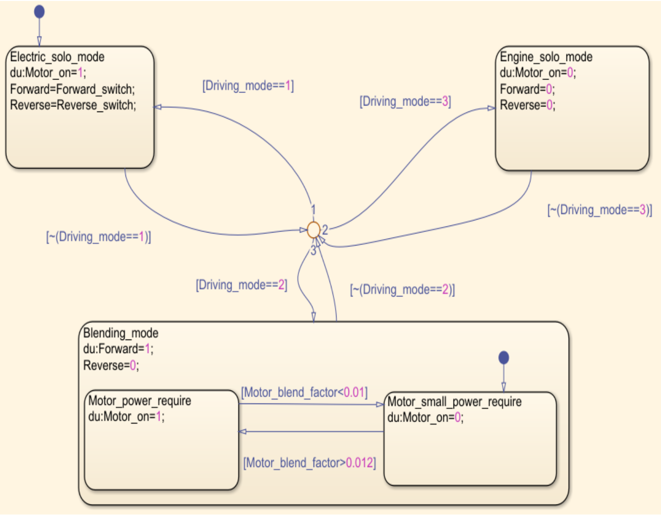

Fig. 9 : State flow diagram of E-motor start/stop model

2.7 Stepper Motor Driving Model

The main function of the 'Stepper Motor Driving Model' is to generate signals to drive the stepper motor. Stepper motor [2] contains four coils which have to be excited in a set sequence in order to move the motor in the desired direction. The stepping scheme, I am using for this project is 'half stepping' and the motor that I am using is bifilar wound, unipolar drive. The signal sequence for the stepper coils for clockwise and anti-clockwise rotation is shown in table 10 below.

This model also takes the 'Engine Throttle Request (ETR)' and 'Throttle Position Sensor (TPS)' value as input and determines when to actuate the stepper motor and in which direction in order to reduce the difference between ETR and TPS. The difference is calculated as,

$Difference$ (D) = $ETR - TPS$

The logic for actuating the motor is shown below Table 11.

Table 10 : Driving circuit and signals for half-step driving

Clock Wise	**Step**	**Coil 1 (Q1)**	**Coil 2 (Q2)**	**Coil 3 (Q3)**	**Coil 4 (Q4)**	
↓	1	ON	OFF	ON	OFF	↑ Counter Clock Wise
	2	ON	OFF	OFF	OFF	
	3	ON	OFF	OFF	ON	
	4	OFF	OFF	OFF	ON	
	5	OFF	ON	OFF	ON	
	6	OFF	ON	OFF	OFF	
	7	OFF	ON	ON	OFF	
	8	OFF	OFF	ON	OFF	
	1	ON	OFF	ON	OFF	

Table 11 : Logic for stepper motor actuation signal

Difference (D)	|D| < 1	D > 0	D < 0
Motor Rotation	Remain Stationary	Clockwise Rotation	Anti-Clockwise Rotation

In this model, the input signals are ETR and TPS. Using the above logic, I improve the model in state flow.

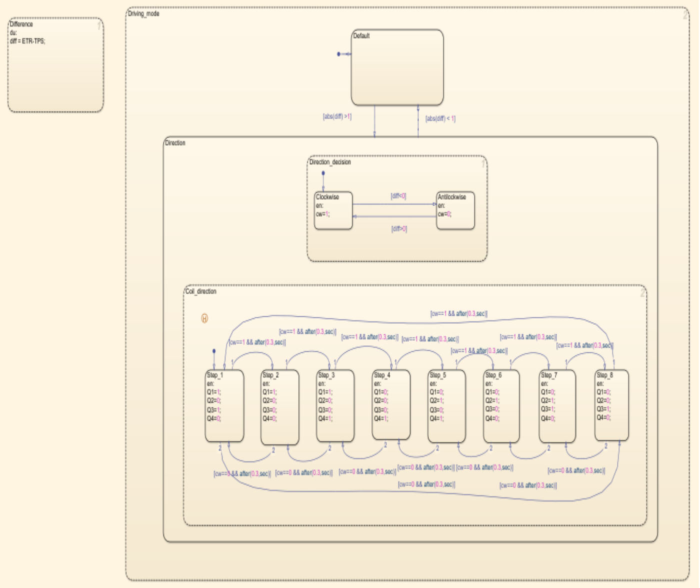

Fig. 10 : Stepper motor driving model in state flow

First, in this model I use two AND states. According to its executive order, first it will calculate the difference between ETR and TPS then it will proceed to the next state. In driving mode state, I design two OR state. The default state that I sate is default state. So, according to the logic, if the absolute difference value is greater than 1 then it will enter the second state. Again, in this second state, I design two AND states. So, first AND state will calculate the difference value is greater than zero or not to determine the coil direction. After that, it will go to the next state. In this state, I design 8 OR states for eight step that is shown in Table X. Here, I use history junction because it helps to record previously active state in which it resided. Also, I use 0.3 second time delay, as the two running coils cannot overlap and use step 1 as a default state. Besides, I use the notation CW for clockwise direction, it will be 1 when CW is true, and it will be 0 when CW is false.

3. APPLICATIONS

This Configurable Hybrid Electric Module is designed by using model-based design methodology [3] Hybride Electric Vehicle are powered by engine and elctric motor. When the car is moving on elctric Solo mode then only elctric motor will work and take the power from battery. When the car is moving on blending mode, then both engine and electric motor will work accoring to the torque demand. In the engine solo mode, only engine will operate to provide the torque. Together, these features result in better fuel economy.

4. CONCLUSION

This paper discussed about vehicles modeling, simulation and analysis using Stateflow diagram. The nine inputs are using for six different control modules and a stepper motor. These six different control modules control the driving mode, engine and motor torque requirement, engine start and stop, forward and backward motion. Whereas Stepper motor controls the motor rotation direction. Every controller is related to each other and working according to the logics and given inputs.

ACKNOWLEDGEMENT

I would like to thank Dr. Bo Chen for giving me the opportunity to research in Embedded Control System Lab from Michigan Technological University and help me to get all the experimental data and logics for this model.

REFERENCES

[1] Introduction to State flow with Applications, Steven T. Karris, Orchard Publications, 2007, ISBN No. 1934404071.

[2] Introduction to Mechatronic Design, J. Edward Carryer, R. Matthew Ohline, Thomas W. Kenny, ISBN-10: 0131433563.

[3] P. Struss and C. Price, "Model-based systems in the automotive industry," AI Mag., vol. 24, no. 4, pp. 17–34, Winter 2004.