Ford Motor Company™

SmartVehicle Baseline Report

Dynamic Analysis of Hybrid System Models for Design Validation

In support of the University California
Open Experimental Platform for DARPA - MoBIES,
Contract F33615-00-C-1698.

April 30, 2002

Alongkrit Chutinan
Emmeskay, Inc.
Plymouth, MI 48170
alongkrit@emmeskay.com

Kenneth R. Butts
Ford Research Laboratory
Dearborn, MI 48121
kbutts1@ford.com

1 Introduction

This report describes an effort by the automotive OEP participants in the DARPA MoBIES program to baseline the capabilities of the current state-of-the-art computational tools for analyzing hybrid system models. Recent efforts by the hybrid system research community [10, 18, 12] have produced a number of promising tools. This report sets forth some of the requirements and provides an assessment of these tools against such requirements. The objective is to provide guidance for future development of these tools and facilitate their adoption by industry. Since a similar effort has already been undertaken in [15] to assess the current state of the hybrid system research, we seek to complement this previous effort by focusing mainly on the tools that are most relevant to the MoBIES powertrain challenge problem [6]. Consequently, our assessment of each tool will vary significantly in depth and scope, depending on the applicability of each tool to the powertrain challenge problem. The reader is referred to [15] for introductory information on various hybrid system tools that may receive inadequate treatment in this report.

This report is organized as follows. Section 2 states the industrial requirements for the analysis tools and justification for such requirements. Section 3 describes the hybrid system model for which the hybrid system analysis challenge problem is stated. In Section 4, we apply the criteria in Section 2 to narrow down the list of tools that will be the candidates for our baseline evaluation using the challenge problem model. As a result of this preliminary assessment, we have selected CheckMate as the first candidate tool for analyzing the challenge problem model because it appears to be the most readily applicable. The conversion of the challenge problem model into an equivalent CheckMate model is then described in Section 5. The final CheckMate model and its validation are described in Section 6. Finally, Section 7 summarizes our experiences thus far and provides our assessments in modeling with CheckMate. The actual analysis of the model using CheckMate is ongoing and will be presented in a subsequent report.

2 Tool Requirements

In order for hybrid system analysis tools to be suitable for industrial use, they must possess the following qualities that enhance the productivity of the analyst:

1. *Simulation Capability*

   It is well known that the analysis of a hybrid system is extremely difficult and computationally expensive in general. Thus, before attempting the analysis, the user would initially want to explore the behaviors of the system via numerical simulations to gain a preliminary understanding of the model behaviors as well as to debug the model. Most of the current hybrid system analysis tools lack this capability since the developers often assume that the user can always use other simulation tools, which are available in abundance, for this purpose. In general, translating the model between tools requires a significant amount of time and effort and is error prone. Thus, the tool should provide a simulation capability, or at least a direct and automated connection to another simulation tool to eliminate the need for manual translation.

2. *Direct Applicability*

The tool should be based on a theory or a framework that directly accommodates the types of models that are commonly encountered in the industry. Many of the current hybrid system tools only admit models with severely restricted classes of continuous dynamics such as timers and polyhedral differential inclusions. To apply these tools to a general hybrid system model, the original model must be approximated by another model that contains only these restricted dynamics. The approximation generally involves a discretization of the continuous state space into several regions. Each region is chosen so that the continuous dynamics of the original model can be closely approximated by a single set of timer dynamics or a polyhedral differential inclusion for the entire region. This type of approximation is in general a manual and time consuming process. Furthermore, the approximating model often suffers from state explosion, making it difficult for the engineer to verify whether the approximate model is correct with respect to the original model. Thus, it is desirable for the hybrid system tool to admit general hybrid system models or at least provide a mechanism to construct the approximating model automatically.

3. *User Friendliness*

   The tool should be user friendly and require only minimal understanding of the underlying analysis methodology. Graphical user interfaces (GUIs) for entering the models and analysis queries as well as for retrieving and interpreting the analysis results are desirable to enhance the productivity of the engineer who uses the tool.

4. *Complete Automation of Analysis*

   The analysis procedures employed by the tool should be completely automatic. User intervention should be kept to the minimum to avoid any potential mistakes that can be made by the user.

5. *Computational Efficiency*

   The tool should be as computationally efficient as possible to reduce the system design validation time and improve time to market.

3 Challenge Problem Model

The analysis query studied in this baseline report asks whether the proposed powertrain system will oscillate between first and second gears given constant throttle and road grade inputs. More specifically, over the space of constant throttle position [0 to 100] and constant road grade [0 to 0.5], does there exist a gear transition "second-to-first-to-second" as the vehicle accelerates from rest (zero vehicle speed) to 100 km/hr. The constant values for throttle position and grade must be chosen so that the initial vehicle acceleration is greater than zero. The gear is initialized to first at the beginning of the simulation.

The model used in this report is simplified and modified relative to the model described in [6]. The engine dynamics have been eliminated and the vehicle dynamics have been reduced to a single state. The transmission and transmission controller models only capture the behavior of first and second gear operation. Drive shaft damping and the second clutch pressure state variable have been added to reduce the original model's response to excitation at the system's driveline natural frequency.

3.1 Nomenclature

The variables used throughout the model are listed below.

T_t : engine (turbine) torque $(N \cdot m)$
T_s : transmission output shaft torque $(N \cdot m)$
veh_speed : vehicle velocity (m/s)

ω_t : turbine rotational velocity (rad/s)
ω_{si} : input sun gear rotational velocity (rad/s)
ω_{cr} : reaction carrier gear rotational velocity (rad/s)
ω_{ci} : input carrier gear rotational velocity (rad/s)

RT_{sp_1} : first clutch sprag reaction torque $(N \cdot m)$
$RT_{c2,up}$: second clutch upstream reaction torque $(N \cdot m)$
$RT_{c2,down}$: second clutch downstream reaction torque $(N \cdot m)$

pc_1 : first clutch (C_1)pressure (kPa)
pc_2 : second clutch (C_2)pressure (kPa)
$pc_{2,target}$: second clutch target pressure (kPa)
$pc_{2,filter}$: filtered second clutch target pressure (kPa)
Tc_1 : first clutch torque $(N \cdot m)$
Tc_2 : second clutch torque $(N \cdot m)$
$c_{1,slip}$: first clutch slip (rad/s)
$c_{2,slip}$: second clutch slip (rad/s)

$grade$: road grade (rad)
tps : throttle position $(\%)$

$gear$: actual transmission gear
to_gear : desired transmission gear
$shift\_speed_{12}$: vehicle speed to shift from first to second (m/s)
$shift\_speed_{21}$: vehicle speed to shift from second to first (m/s)

The parameters used throughout the model, along with their values, are listed below.

$$
\begin{aligned}
AR_1 &= 2.912 : \text{clutch gain } (m^3) \\
c_1\mu_1 &= 0.1316 : \text{static coefficient of friction } (dimensionless) \\
c_1\mu_2 &= 0.0001748 : \text{dynamic coefficient of friction } (rad^{-1}) \\
\\
AR_2 &= 4.125 : \text{clutch gain } (m^3) \\
c_2\mu_1 &= 0.1316 : \text{static coefficient of friction } (dimensionless) \\
c_2\mu_2 &= 0.0001748 : \text{dynamic coefficient of friction } (rad^{-1}) \\
\\
pc_{1,max} &= 1000.0 : (kPa) \\
pc_{2,max} &= 400.0 : (kPa) \\
pc_{2,torque\_phase} &= 0.4 : \text{normalized pressure offset } (dimensionless) \\
\\
I_t &= 0.05623 : \text{turbine inertia } (kg \cdot m^2) \\
I_{si} &= 0.001020 : \text{input sun gear inertia } (kg \cdot m^2) \\
I_{ci} &= 0.009020 : \text{input carrier gear inertia } (kg \cdot m^2) \\
I_{cr} &= 0.005806 : \text{reaction carrier gear inertia } (kg \cdot m^2) \\
I_{t_1} &= I_t + I_{si} + R_1^2 I_{cr} + \frac{R_1^2}{R_2^2} I_{ci} \\
I_{t_2} &= I_t + I_{ci} + R_2^2 I_{cr} + \frac{R_2^2}{R_1^2} I_{si} \\
I_{si_{12}} &= I_{t_1} - I_t \\
I_{ci_{12}} &= I_{t_2} - I_t \\
I_{cr_{12}} &= I_{cr} + \frac{I_{si}}{R_1^2} + \frac{I_{ci}}{R_2^2} \\
\\
R_{si} &= 0.2955 : \text{input sun gear ratio} \\
R_{ci} &= 0.6379 : \text{input carrier gear ratio} \\
R_{cr} &= 0.7045 : \text{reaction carrier gear ratio} \\
R_d &= 0.3521 : \text{final drive gear ratio} \\
\\
R_1 &= \frac{R_{ci} R_{si}}{1 - R_{ci} R_{cr}} : \text{1st gear ratio} \\
R_2 &= R_{ci} : \text{2nd gear ratio} \\
\\
M &= 1644.0 + 125.0 : \text{vehicle mass } (kg) \\
g &= 9.81 : \text{gravity constant } (m/s^2) \\
H_f &= 0.310 : \text{static ground-to-axle height of front wheel } (m) \\
I_{wf} &= 2.8 : \text{front wheel inertia (both sides) } (kg \cdot m^2)
\end{aligned}
$$

$$K_s = 6742.0 : \text{drive shaft spring constant } (N \cdot m/rad)$$

3.2 Model Inputs

The exogenous inputs to the model are the throttle position tps in % and the road $grade$ in rad.

3.3 Engine Torque

A simple affine function of throttle position is used to model engine torque production. This simplification is reasonable since the problem statement restricts the throttle position to a constant. The torque generation model is shown below.

$$T_t = 1.7tps + 30 \tag{1}$$

The units for engine torque T_t are $N \cdot m$.

3.4 Clutches

The actuators in this system are the first and second clutches C_1 and C_2. Their torque capacities are managed by controlling the pressures pc_1 and pc_2 respectively.

$$c_{1,slip} = \begin{cases} \omega_t - \omega_{si}, & |\omega_t - \omega_{si}| \geq 0.5 \\ 0, & |\omega_t - \omega_{si}| < 0.5 \end{cases} \tag{2}$$

$$Tc_1 = sgn(c_{1,slip})\,(c_1\mu_2|\omega_t - \omega_{si}| + c_1\mu_1)\,AR_1pc_1 \tag{3}$$

$$c_{2,slip} = \begin{cases} \omega_t - \omega_{ci}, & |\omega_t - \omega_{ci}| \geq 0.5 \\ 0, & |\omega_t - \omega_{ci}| < 0.5 \end{cases} \tag{4}$$

$$Tc_2 = sgn(c_{2,slip})\,(c_2\mu_2|\omega_t - \omega_{ci}| + c_2\mu_1)\,AR_2pc_2 \tag{5}$$

The units for clutch torques Tc_1 and Tc_2 are $N \cdot m$. Clutch slips $c_{1,slip}$ and $c_{2,slip}$ are measured in rad/s. The units for clutch pressures pc_1 and pc_2 are in kPa.

3.5 Controller

The transmission controller decides when the transmission should shift, and modulates the clutch pressures during shifts to control shift quality. The transmission controller is modeled for first and second gear operation in this study. Clutch pressures pc_1 and pc_2 are given by

$$pc_1 = \begin{cases} pc_{1,max}, & to\_gear \in \{1, 2\} \\ 0, & otherwise \end{cases} \tag{6}$$

$$pc_{2,target} = \begin{cases} pc_{2,max}, & to\_gear = 2 \\ 0, & otherwise \end{cases} \tag{7}$$

$$\dot{pc}_{2,filter} = -pc_{2,filter} + (1 - pc_{2,torque\_phase})pc_{2,target} \tag{8}$$

$$pc_2 = pc_{2,filter} + pc_{2,torque\_phase} \cdot pc_{2,target} \tag{9}$$

where to_gear is determined by the *shift scheduler* state machine shown in Figure 1. Again, the units for clutch pressures pc_1 and pc_2 are in kPa.

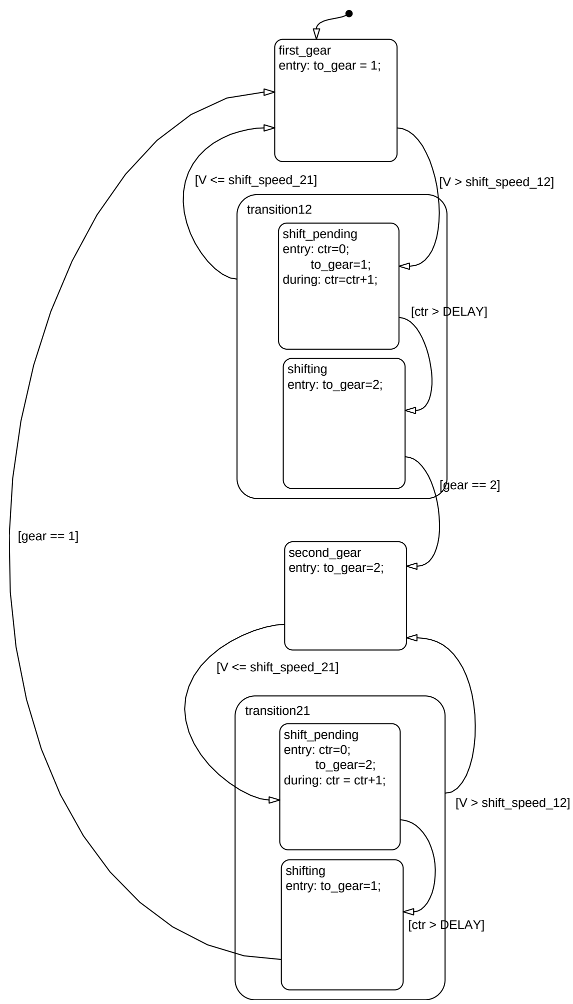

Figure 1: Shift Scheduler

The up-shift and down-shift speeds in km/h for the above state machine are determined by

$$shift\_speed_{12} = \begin{cases} 20, & tps \leq 30 \\ 1.7(tps - 30) + 20, & 30 < tps < 80 \\ 55, & tps \geq 80 \end{cases} \tag{10}$$

and

$$shift\_speed_{21} = \begin{cases} 14, & tps \leq 80 \\ 364(tps - 80) + 14, & 80 < tps < 80.1 \\ 50.4, & tps \geq 80.1 \end{cases} . \tag{11}$$

The variable $gear$ comes from another state machine shown in Figure 2. Note that $gear$ is actually written as a_gear in the state machine. The parameter $DELAY$ is set to 1.

3.6 Transmission

The transmission operates in multiple dynamic modes. The particular mode that the transmission is in is determined by the state machine shown in Figure 2. Note that the transmission's operation is only described for first and second gears. The model is always initialized to first gear.

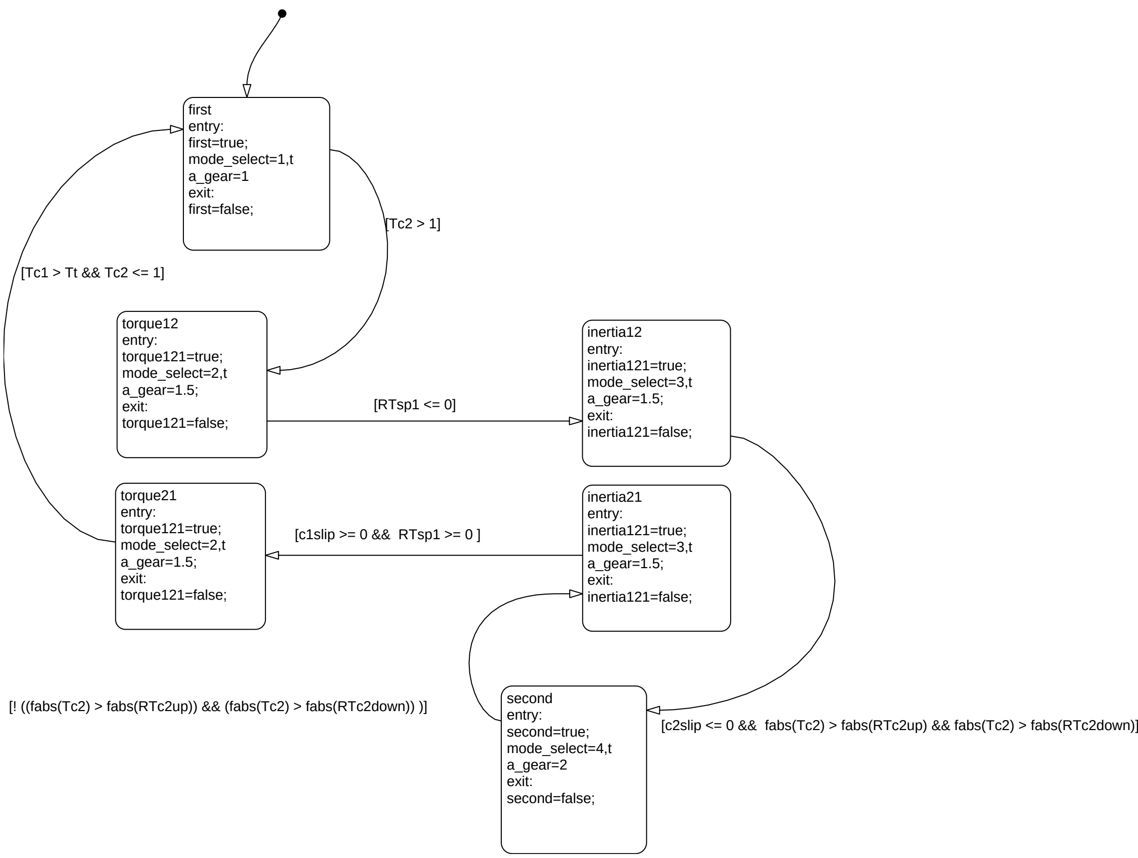

Figure 2: Transmission Dynamic Mode Selection

The dynamics for each of the transmission modes are governed by the following equations.

$$RT_{sp_1} = I_{si_{12}}\dot{\omega}_{si} + R_1 R_d T_s - \frac{R_1}{R_2} Tc_2 \tag{12}$$

$$RT_{c_2,up} = \begin{cases} 0, & gear = 1 \\ -I_t\dot{\omega}_t + T_t, & gear \in \{1.5, 2\} \end{cases} \tag{13}$$

$$RT_{c_2,down} = \begin{cases} 0, & gear = 1 \\ I_{ci_{12}}\dot{\omega}_{ci} + R_2 R_d T_s, & gear \in \{1.5, 2\} \end{cases} \tag{14}$$

$$\dot{\omega}_t = \begin{cases} \frac{1}{I_{t_1}}\left(T_t - R_1 R_d T_s\right), & dynamic\_mode = first \\ \frac{1}{I_{t_1}}\left(T_t - R_1 R_d T_s - (1 - \frac{R_1}{R_2}) T_{c_2}\right), & dynamic\_mode \in \{torque_{12}, torque_{21}\} \\ \frac{1}{I_t}\left(T_t - T_{c_2}\right), & dynamic\_mode \in \{inertia_{12}, inertia_{21}\} \\ \frac{1}{I_{t_2}}\left(T_t - R_2 R_d T_s\right), & dynamic\_mode = second \end{cases} \tag{15}$$

$$\dot{\omega}_{cr} = \begin{cases} R_1 \dot{\omega}_t, & dynamic\_mode \in \{first, torque_{12}, torque_{21}\} \\ \frac{1}{I_{cr_{12}}}\left(\frac{T_{c_2}}{R_2} - R_d T_s\right), & dynamic\_mode \in \{inertia_{12}, inertia_{21}\} \\ R_2 \dot{\omega}_t, & dynamic\_mode = second \end{cases} \tag{16}$$

$$\omega_{si} = \begin{cases} \omega_t, & dynamic\_mode \in \{first, torque_{12}, torque_{21}\} \\ \frac{1}{R_1}\omega_{cr}, & dynamic\_mode \in \{inertia_{12}, inertia_{21}\} \\ \frac{R_2}{R_1}\omega_t, & dynamic\_mode = second \end{cases} \tag{17}$$

$$\omega_{ci} = \begin{cases} \frac{R_1}{R_2}\omega_t, & dynamic\_mode \in \{first, torque_{12}, torque_{21}\} \\ \frac{1}{R_2}\omega_{cr}, & dynamic\_mode \in \{inertia_{12}, inertia_{21}\} \\ \omega_t, & dynamic\_mode = second \end{cases} \tag{18}$$

3.7 Drive Shaft Torque and Vehicle Dynamics

For this study, no-slip wheels and a single state for the vehicle speed have replaced the original system's wheel and vehicle dynamics. Also, the aerodynamic drag and rolling resistance terms have been eliminated from the vehicle dynamics. The brake input has been removed since it is not used in the problem statement. An extra shaft damping term has been introduced in the shaft torque equation to reduce the propensity for driveline oscillations. The resulting simplified drive shaft torque and vehicle dynamics are given by

$$\dot{T}_s = K_s \left(R_d \omega_{cr} - \frac{1}{H_f} veh\_speed - 0.001 T_s \right) \tag{19}$$

and

$$veh\_\dot{s}peed = \frac{\left(\frac{T_s}{H_f} - Mg \cdot sin(grade)\right)}{\left(M + \frac{2I_{wf}}{H_f^2}\right)}. \tag{20}$$

The units for the drive shaft torque T_s and vehicle speed veh_speed are $N \cdot m$ and m/s, respectively.

4 Selecting Tools for Baseline Evaluation

Due to limited resources, it is not possible for us to evaluate every hybrid system tool available. Thus, we need to carefully select a subset of tools for our baseline evaluation using the challenge problem model described in the previous section. As a starting point for our tool selection process, we first compile the comparison information from the currently available hybrid system tools in Table 1. The table was based on a similar table appearing in [15], and augmented with information from other sources as indicated in the table itself. The comparison criteria in the table reflect the tool requirements stated in Section 2. Based on Table 1, we select the tools by process of elimination.

Considering Requirement 2 in Section 2, the tools that are most directly applicable to the challenge problem model are

- VERDICT
- CHECKMATE
- d/dt
- VERISHIFT

The tools SM2SMV, HYTECH, KRONOS, TAXYS, and UPPAAL, are not chosen because the continuous dynamics in the challenge problem model will need to be discretized in order to use these tools. VERDICT remains a candidate even though it uses these tools because it does provide an automatic model conversion. Finally, although HYSDEL handles a fairly general dynamics, it is eliminated because it only deals with discrete-time models.

Next, we consider Requirement 3. For this requirement, the tools that appear to be the most user friendly are

- VERDICT
- CHECKMATE

The tools d/dt and VERISHIFT are eliminated because they seem to rely heavily on textual programming for modeling. The same is true, but to a lesser extent, in both VERDICT and CHECKMATE where the continuous dynamics must be entered textually.

Finally, considering Requirement 1, we find that the only tool equipped with a simulation capability is

- CHECKMATE

Even though VERDICT does provide a connection to UPPAAL, which has a simulation capability, it may be difficult to relate the simulation results for the approximating model in UPPAAL to the original model for reasons discussed under Requirement 2 in Section 2. Since we are down to only one candidate tool, we stop our elimination process and select CHECKMATE as the first hybrid system tool for our baseline evaluation. Time permitting, we will also evaluate d/dt and VERISHIFT since they cover fairly general continuous dynamics and have features such as control synthesis that are not available in CHECKMATE. The next two sections describe the CHECKMATE conversion of the challenge problem model in detail.

5 CheckMate Model Conversion

The challenge problem model presented in Section 3 cannot be used directly as an input to CHECKMATE. The following subsections describe the steps we undergo to convert the challenge problem model into a model admissible by CHECKMATE.

Tool	Model Class	Graphical Interface	Analysis Capabilities
SM2SMV [20]	Finite state machines	Yes (Stateflow).	Simulation (Stateflow). Verification (SMV).
HyTech [8, 9]	Linear hybrid automata	No.	Verification.
Kronos [7, 16]	Timed automata		Verification.
Taxys [19]	Timed automata		Verification (Kronos).
Uppaal [4, 13]	Timed automata	Yes.	Simulation. Verification.
Verdict [11]	Hybrid systems w/ nonlinear ODEs and rectangular guards	Yes for modeling, but continuous dynamics must be entered textually.	Verification (translation to HyTech, Kronos, SMV, and Uppaal).
CheckMate [14]	Hybrid systems w/ nonlinear ODEs and linear guards	Yes for modeling (Simulink/Stateflow), but continuous dynamics must be entered textually. No for analysis.	Simulation (Simulink/ Stateflow). Verification.
d/dt [2, 1]	Hybrid systems w/ linear differential inclusions	No for modeling. Yes for analysis.	Verification. Control synthesis.
VeriSHIFT [5]	Hybrid systems w/ linear differential inclusions		Reachability analysis.
HYSDEL [3, 17]	Discrete-time hybrid systems w/ linear difference equations	No.	Verification.

Table 1: Hybrid System Tools Comparison

5.1 Identifying Continuous and Discrete States

CheckMate requires a complete separation of continuous and discrete dynamics in the model. Thus, the first step in the model conversion is to identify and separate the continuous and discrete states in the challenge problem model. For the challenge problem model, this task is straightforward since the discrete states in the state machines in Figures 1 and 2 are already separated from the continuous states. The continuous states can be found by simply searching the model for differential equations. Table 2 summarizes the results for this step.

Continuous States	Discrete States
T_s	$shift\_schedule$
veh_speed	$dynamic\_mode$
$pc_{2,filter}$	
ω_t	
ω_{cr}	

Table 2: Continuous and Discrete States

The two discrete states, $shift\_schedule$ and $dynamic\_mode$, represent the states for the state machines in Figures 1 and 2, respectively. Note that the counter variable ctr in the shift scheduler in Figure 1 is also a discrete state. It has been omitted from Table 2, however, because we have concluded that the dynamics of this counter in the shift scheduler are negligible. The update period of 0.5 msec for the counter, which is the same as the model sampling period, is very fast compared to the dynamics of the continuous states in the model. Neglecting the counter dynamics, the two $shift\_pending$ states in Figure 1 are eliminated, resulting in the state machine shown in Figure 4 (see Section 6).

5.2 Replacing Continuous Guard Conditions

CheckMate requires that all the guard conditions that control the discrete transitions in the state machines be expressed as linear combinations of continuous states. Not all guard conditions in the challenge problem model satisfy this requirement, however. Most guard conditions are expressed in terms of intermediate variables. These intermediate variables depend on both continuous and discrete states.

For the guard conditions that depend only on continuous intermediate variables, we recursively substitute the intermediate variables in the guard conditions by their expressions until we have only the continuous states in the guard expressions. The following subsections describe the conversion of the continuous guard conditions in each state machine. The conversion for the guard conditions that depend only on discrete intermediate variables is deferred to Section 5.3.

5.2.1 Dynamic Mode Selection

The conversion procedure described above is applied to the guard condition in each state of the dynamic mode selection state machine in Figure 2 as follows. All the derivations are made with reference the model equations in Section 3.

1. *first*

From Figure 2, the guard condition for this state is

$$Tc_2 > 1.$$

From (18), we have that

$$\omega_{ci} = \frac{R_1}{R_2}\omega_t.$$

Here we must resort to analyst intervention and thus violate Requirement 4 in Section 2 as follows. We note that the transmission should not transition out of the state *first* if the shift scheduler has not issued a shift command. For the transition to occur, it must be the case that

$$to\_gear = 2.$$

Consequently, we have from (7) and (9) that

$$pc_{2,target} = pc_{2,max}$$

and

$$pc_2 = pc_{2,filter} + pc_{2,torque\_phase} pc_{2,max}. \tag{21}$$

We now consider 3 cases for the quantity $\omega_t - \omega_{ci}$ which is equivalent to $(1 - \frac{R_1}{R_2})\omega_t$ for the current state.

(a) $\omega_t - \omega_{ci} \geq 0 \iff \left(1 - \frac{R_1}{R_2}\right)\omega_t \geq 0.$

In this case, we have $sgn(c_{2,slip}) = sgn(\omega_t - \omega_{ci}) = 1$ and $|\omega_t - \omega_{ci}| = \omega_t - \omega_{ci} = \left(1 - \frac{R_1}{R_2}\right)\omega_t$. Thus we have from (5) that

$$\begin{aligned} Tc_2 &= \left(c_2\mu_2\left(1 - \frac{R_1}{R_2}\right)\omega_t + c_2\mu_1\right) AR_2 pc_2 \\ &= c_2\mu_2\left(1 - \frac{R_1}{R_2}\right) AR_2 pc_2 \omega_t + c_2\mu_1 AR_2 pc_2 \end{aligned}$$

Substituting (21) and expanding all the terms, we have that

$$\begin{aligned} Tc_2 = \; & c_2\mu_2\left(1 - \frac{R_1}{R_2}\right) AR_2 pc_{2,filter}\omega_t \\ & + c_2\mu_2\left(1 - \frac{R_1}{R_2}\right) AR_2 pc_{2,torque\_phase} pc_{2,max}\omega_t \\ & + c_2\mu_1 AR_2 pc_{2,filter} \\ & + c_2\mu_1 AR_2 pc_{2,torque\_phase} pc_{2,max} \end{aligned} \tag{22}$$

(b) $-0.5 < \omega_t - \omega_{ci} < 0 \iff -0.5 < \left(1 - \frac{R_1}{R_2}\right)\omega_t < 0.$

In this case, we have $sgn(c_{2,slip}) = sgn(0) = 1$ and $|\omega_t - \omega_{ci}| = -(\omega_t - \omega_{ci}) = -\left(1 - \frac{R_1}{R_2}\right)\omega_t$. Thus we have from (5) that

$$\begin{aligned} Tc_2 &= \left(-c_2\mu_2\left(1 - \frac{R_1}{R_2}\right)\omega_t + c_2\mu_1\right) AR_2 pc_2 \\ &= -c_2\mu_2\left(1 - \frac{R_1}{R_2}\right) AR_2 pc_2 \omega_t + c_2\mu_1 AR_2 pc_2 \end{aligned}$$

Substituting (21) and expanding all the terms, we have that

$$\begin{aligned}
Tc_2 &= -c_2\mu_2\left(1-\frac{R_1}{R_2}\right)AR_2pc_{2,filter}\omega_t \\
&\quad -c_2\mu_2\left(1-\frac{R_1}{R_2}\right)AR_2pc_{2,torque_phase}pc_{2,max}\omega_t \\
&\quad +c_2\mu_1AR_2pc_{2,filter} \\
&\quad +c_2\mu_1AR_2pc_{2,torque_phase}pc_{2,max} \qquad (23)
\end{aligned}$$

(c) $\omega_t-\omega_{ci}\leq -0.5 \iff \left(1-\frac{R_1}{R_2}\right)\omega_t \leq -0.5$.
In this case, we have $sgn(c_{2,slip}) = sgn(\omega_t-\omega_{ci}) = -1$ and $|\omega_t-\omega_{ci}| = -(\omega_t-\omega_{ci}) = -\left(1-\frac{R_1}{R_2}\right)\omega_t$. Thus we have from (5) that

$$\begin{aligned}
Tc_2 &= -\left(-c_2\mu_2\left(1-\frac{R_1}{R_2}\right)\omega_t + c_2\mu_1\right)AR_2pc_2 \\
&= c_2\mu_2\left(1-\frac{R_1}{R_2}\right)AR_2pc_2\omega_t - c_2\mu_1AR_2pc_2
\end{aligned}$$

Substituting (21) and expanding all the terms, we have that

$$\begin{aligned}
Tc_2 &= c_2\mu_2\left(1-\frac{R_1}{R_2}\right)AR_2pc_{2,filter}\omega_t \\
&\quad +c_2\mu_2\left(1-\frac{R_1}{R_2}\right)AR_2pc_{2,torque_phase}pc_{2,max}\omega_t \\
&\quad -c_2\mu_1AR_2pc_{2,filter} \\
&\quad -c_2\mu_1AR_2pc_{2,torque_phase}pc_{2,max} \qquad (24)
\end{aligned}$$

From the 3 cases above, we can rewrite the guard condition $Tc_2 > 1$ in terms of the continuous states as

$$\begin{aligned}
Tc_2 > 1 \iff &\left[\left(1-\frac{R_1}{R_2}\right)\omega_t \geq 0 \wedge (22) > 1\right] \\
&\vee\left[-0.5 < \left(1-\frac{R_1}{R_2}\right)\omega_t < 0 \wedge (23) > 1\right] \\
&\vee\left[\left(1-\frac{R_1}{R_2}\right)\omega_t \leq -0.5 \wedge (24) > 1\right] \qquad (25)
\end{aligned}$$

At initialization or prior to entering the state *first*, to_gear must have been 1. Thus, for to_gear to switch to 2, we must also have that the vehicle speed is above $shift\_speed_{12}$. Thus, we can rewrite the final guard condition for the state *first* as

$$Tc_2 > 1 \wedge veh\_speed > shift\_speed_{12} \iff (25) \wedge veh\_speed > shift\_speed_{12} \qquad (26)$$

Note that the expressions in (22), (23), and (24) are not linear in the continuous states. Thus, the above guard expression is still not admissible in CheckMate. We address this problem in Section 5.2.3.

2. *torque12*

From Figure 2, the guard condition for this state is

$$RT_{sp_1} \leq 0.$$

From (18), (17), and (15), we have that

$$\omega_{ci} = \frac{R_1}{R_2}\omega_t,\ \omega_{si} = \omega_t,\ \text{and}\ \dot{\omega}_t = \frac{1}{I_{t_1}}\left(T_t - R_1 R_d T_s - (1 - \frac{R_1}{R_2})T_{c_2}\right).$$

Substituting the above results into (12), we have

$$RT_{sp_1} = \frac{I_{si_{12}}}{I_{t_1}}T_t + \left(1 - \frac{I_{si_{12}}}{I_{t_1}}\right) R_1 R_d T_s - \left(\frac{I_{si_{12}}}{I_{t_1}}\left(1 - \frac{R_1}{R_2}\right) + \frac{R_1}{R_2}\right) T_{c_2} \tag{27}$$

Using the results for the state *first* above, we consider 3 cases for the quantity $\omega_t - \omega_{ci}$ which is equivalent to $(1 - \frac{R_1}{R_2})\omega_t$ for the current state.

(a) $\omega_t - \omega_{ci} \geq 0 \iff \left(1 - \frac{R_1}{R_2}\right)\omega_t \geq 0.$

Substituting (22) into (27), and expanding the terms we have that

$$\begin{aligned} RT_{sp_1} \quad = \quad & \frac{I_{si_{12}}}{I_{t_1}}T_t + \left(1 - \frac{I_{si_{12}}}{I_{t_1}}\right) R_1 R_d T_s \\ & - \left(\frac{I_{si_{12}}}{I_{t_1}}\left(1 - \frac{R_1}{R_2}\right) + \frac{R_1}{R_2}\right) c_2 \mu_2 \left(1 - \frac{R_1}{R_2}\right) A R_2 pc_{2,filter}\omega_t \\ & - \left(\frac{I_{si_{12}}}{I_{t_1}}\left(1 - \frac{R_1}{R_2}\right) + \frac{R_1}{R_2}\right) c_2 \mu_2 \left(1 - \frac{R_1}{R_2}\right) A R_2 pc_{2,torque\_phase}pc_{2,max}\omega_t \\ & - \left(\frac{I_{si_{12}}}{I_{t_1}}\left(1 - \frac{R_1}{R_2}\right) + \frac{R_1}{R_2}\right) c_2 \mu_1 A R_2 pc_{2,filter} \\ & - \left(\frac{I_{si_{12}}}{I_{t_1}}\left(1 - \frac{R_1}{R_2}\right) + \frac{R_1}{R_2}\right) c_2 \mu_1 A R_2 pc_{2,torque\_phase}pc_{2,max} \end{aligned} \tag{28}$$

(b) $-0.5 < \omega_t - \omega_{ci} < 0 \iff -0.5 < \left(1 - \frac{R_1}{R_2}\right)\omega_t < 0.$

Substituting (23) into (27), and expanding the terms we have that

$$\begin{aligned} RT_{sp_1} \quad = \quad & \frac{I_{si_{12}}}{I_{t_1}}T_t + \left(1 - \frac{I_{si_{12}}}{I_{t_1}}\right) R_1 R_d T_s \\ & + \left(\frac{I_{si_{12}}}{I_{t_1}}\left(1 - \frac{R_1}{R_2}\right) + \frac{R_1}{R_2}\right) c_2 \mu_2 \left(1 - \frac{R_1}{R_2}\right) A R_2 pc_{2,filter}\omega_t \\ & + \left(\frac{I_{si_{12}}}{I_{t_1}}\left(1 - \frac{R_1}{R_2}\right) + \frac{R_1}{R_2}\right) c_2 \mu_2 \left(1 - \frac{R_1}{R_2}\right) A R_2 pc_{2,torque\_phase}pc_{2,max}\omega_t \\ & - \left(\frac{I_{si_{12}}}{I_{t_1}}\left(1 - \frac{R_1}{R_2}\right) + \frac{R_1}{R_2}\right) c_2 \mu_1 A R_2 pc_{2,filter} \\ & - \left(\frac{I_{si_{12}}}{I_{t_1}}\left(1 - \frac{R_1}{R_2}\right) + \frac{R_1}{R_2}\right) c_2 \mu_1 A R_2 pc_{2,torque\_phase}pc_{2,max} \end{aligned} \tag{29}$$

(c) $\omega_t - \omega_{ci} \leq -0.5 \iff \left(1 - \frac{R_1}{R_2}\right)\omega_t \leq -0.5.$

Substituting (24) into (27), and expanding the terms we have that

$$\begin{aligned} RT_{sp_1} &= \frac{I_{si_{12}}}{I_{t_1}} T_t + \left(1 - \frac{I_{si_{12}}}{I_{t_1}}\right) R_1 R_d T_s \\ &- \left(\frac{I_{si_{12}}}{I_{t_1}}\left(1 - \frac{R_1}{R_2}\right) + \frac{R_1}{R_2}\right) c_2 \mu_2 \left(1 - \frac{R_1}{R_2}\right) A R_2 pc_{2,filter} \omega_t \\ &- \left(\frac{I_{si_{12}}}{I_{t_1}}\left(1 - \frac{R_1}{R_2}\right) + \frac{R_1}{R_2}\right) c_2 \mu_2 \left(1 - \frac{R_1}{R_2}\right) A R_2 pc_{2,torque\_phase} pc_{2,max} \omega_t \\ &+ \left(\frac{I_{si_{12}}}{I_{t_1}}\left(1 - \frac{R_1}{R_2}\right) + \frac{R_1}{R_2}\right) c_2 \mu_1 A R_2 pc_{2,filter} \\ &+ \left(\frac{I_{si_{12}}}{I_{t_1}}\left(1 - \frac{R_1}{R_2}\right) + \frac{R_1}{R_2}\right) c_2 \mu_1 A R_2 pc_{2,torque\_phase} pc_{2,max} \end{aligned} \quad (30)$$

From the 3 cases above, we can rewrite the guard condition $RT_{sp_1} \leq 0$ in terms of the continuous states as

$$\begin{aligned} RT_{sp_1} \leq 0 \quad \iff \quad & \left[\left(1 - \frac{R_1}{R_2}\right)\omega_t \geq 0 \wedge (28) \leq 0\right] \\ & \vee \left[-0.5 < \left(1 - \frac{R_1}{R_2}\right)\omega_t < 0 \wedge (29) \leq 0\right] \\ & \vee \left[\left(1 - \frac{R_1}{R_2}\right)\omega_t \leq -0.5 \wedge (30) \leq 0\right] \end{aligned} \quad (31)$$

Note that the expressions in (28), (29), and (30) are not linear in the continuous states. We address this problem in Section 5.2.3.

3. *inertia12*

From Figure 2, the guard condition for this state is

$$c_{2,slip} \leq 0 \wedge |Tc_2| > |RT_{c_2,up}| \wedge |Tc2| > |RT_{c_2,down}|.$$

From (18), (16), and (15), we have that

$$\omega_{ci} = \frac{1}{R_2}\omega_{cr},\ \dot{\omega}_{cr} = \frac{1}{I_{cr_{12}}}\left(\frac{T_{c_2}}{R_2} - R_d T_s\right),\ \text{and } \dot{\omega}_t = \frac{1}{I_t}\left(T_t - T_{c_2}\right).$$

From Section 3.1,

$$I_{ci_{12}} = I_{t_2} - I_t = I_{ci} + R_2^2 I_{cr} + \frac{R_2^2}{R_1^2} I_{si} = R_2^2\left(\frac{I_{ci}}{R_2^2} + I_{cr} + \frac{I_{si}}{R_1^2}\right) = R_2^2 I_{cr_{12}}$$

Since $gear = 1.5$ for the current state, we have from the above results, (13), and (14) that

$$RT_{c_2,up} = -\frac{I_t}{I_t}\left(T_t - T_{c_2}\right) + T_t = T_{c_2} - T_t + T_t = T_{c_2}$$

and

$$RT_{c_2,down} = \frac{I_{ci_{12}}}{R_2^2 I_{cr_{12}}} \left(T_{c_2} - R_2 R_d T_s\right) + R_2 R_d T_s = T_{c_2}.$$

The above results imply that the conditions $|Tc_2| > |RT_{c_2,up}|$ and $|Tc_2| > |RT_{c_2,down}|$ are always *false*. The transition, however, does take place during the simulation in Simulink. This can possibly be attributed to numerical errors incurred during the simulation. To allow the transition to be taken, we omit these two conditions from the guard condition for the current state. We now examine the remaining condition $c_{2,slip} \leq 0$ in the guard condition. From (4), we have that

$$c_{2,slip} \leq 0 \iff \omega_t - \omega_{ci} < 0.5$$

Thus, we can rewrite the guard condition $c_{2,slip} \leq 0$ in terms of the continuous states as

$$c_{2,slip} \leq 0 \iff \omega_t - \frac{1}{R_2}\omega_{cr} < 0.5. \tag{32}$$

The above guard condition is linear in the continuous states.

4. *second*

   From Figure 2, the guard condition for this state is

$$\neg \left(|Tc_2| > |RT_{c_2,up}| \wedge |Tc_2| > |RT_{c_2,down}|\right)$$

   From (18), we have that $\omega_{ci} = \omega_t$. Thus, we have from (4) that $c_{2,slip} = 0$ and $sgn(c_{2,slip}) = 1$. Consequently, we have from (5) that

$$Tc_2 = c_2 \mu_1 A R_2 p c_2. \tag{33}$$

   From (15), we have that

$$\dot{\omega}_t = \frac{1}{I_{t_2}} \left(T_t - R_2 R_d T_s\right).$$

   From Figure 2, we have that $gear = 2$ for the current state. Using the results we have so far in conjunction with (13) and (14), we have that

$$\begin{aligned} RT_{c_2,up} &= -I_t \dot{\omega}_t + T_t = -\frac{I_t}{I_{t_2}} \left(T_t - R_2 R_d T_s\right) + T_t \\ &= \left(1 - \frac{I_t}{I_{t_2}}\right) T_t + \frac{I_t}{I_{t_2}} R_2 R_d T_s \end{aligned}$$

   and

$$\begin{aligned} RT_{c_2,down} &= I_{ci_{12}} \dot{\omega}_{ci} + R_2 R_d T_s = \frac{I_{ci_{12}}}{I_{t_2}} \left(T_t - R_2 R_d T_s\right) + R_2 R_d T_s \\ &= \frac{I_{ci_{12}}}{I_{t_2}} T_t + \left(1 - \frac{I_{ci_{12}}}{I_{t_2}}\right) R_2 R_d T_s. \end{aligned}$$

   From Section 3.1,

$$I_{t_2} = I_{ci_{12}} + I_t \Rightarrow \frac{I_{ci_{12}}}{I_{t_2}} = 1 - \frac{I_t}{I_{t_2}} \text{ and } \frac{I_t}{I_{t_2}} = 1 - \frac{I_{ci_{12}}}{I_{t_2}}$$

Thus, we have that

$$RT_{c_2,up} = RT_{c_2,down}$$

and consequently, the guard condition reduces to

$$\neg\left(|Tc_2| > |RT_{c_2,up}|\right).$$

Here, again, we must resort to analyst intervention and thus violate Requirement 4 in Section 2 as follows. We note that the transmission should not transition out of the state *second* if the shift scheduler has not issued a shift command. For the transition to occur, it must be the case that

$$to\_gear = 1.$$

Consequently, we have from (7) and (9) that $pc_{2,target} = 0$ and that $pc_2 = pc_{2,filter}$. Thus, from (33),

$$Tc_2 = c_2 \mu_1 A R_2 pc_{2,filter}.$$

We conclude from the above expression that $Tc_2 \geq 0$ because all quantities on the right-hand side of the above equation are non-negative. Therefore,

$$\begin{aligned} |RT_{c_2,up}| < |Tc_2| \quad &\Longleftrightarrow \quad RT_{c_2,up} > -Tc_2 \wedge RT_{c_2,up} < Tc_2 \\ &\Longleftrightarrow \quad \left(1 - \frac{I_t}{I_{t_2}}\right) T_t + \frac{I_t}{I_{t_2}} R_2 R_d T_s > -c_2 \mu_1 A R_2 pc_{2,filter} \\ &\qquad \wedge \left(1 - \frac{I_t}{I_{t_2}}\right) T_t + \frac{I_t}{I_{t_2}} R_2 R_d T_s < c_2 \mu_1 A R_2 pc_{2,filter} \qquad (34) \end{aligned}$$

Prior to entering the state *second*, to_gear must have been 2. Thus, for to_gear to switch back to 1 again, we must also have that the vehicle speed is below $shift\_speed_{21}$. Thus, we can rewrite the final guard condition for the state *second* as

$$\begin{aligned} &\neg\left(|Tc_2| > |RT_{c_2,up}|\right) \wedge (veh\_speed \leq shift\_speed_{21}) \\ &\Longleftrightarrow \neg(34) \wedge (veh\_speed \leq shift\_speed_{21}) \qquad (35) \end{aligned}$$

The above guard condition is linear in the continuous states.

5. *inertia21*

From Figure 2, the the guard condition for this state is

$$c_{1,slip} \geq 0 \wedge RT_{sp_1} \geq 0.$$

From (17) and (16), we have that

$$\omega_{si} = \frac{1}{R_1}\omega_{cr} \text{ and } \dot{\omega}_{cr} = \frac{1}{I_{cr_{12}}}\left(\frac{T_{c_2}}{R_2} - R_d T_s\right).$$

Substituting the above results into (12), we have that

$$\begin{aligned} RT_{sp_1} &= \frac{I_{si_{12}}}{R_1}\dot{\omega}_{cr} + R_1 R_d T_s - \frac{R_1}{R_2} Tc_2 \\ &= \frac{I_{si_{12}}}{I_{cr_{12}} R_1}\left(\frac{T_{c_2}}{R_2} - R_d T_s\right) + R_1 R_d T_s - \frac{R_1}{R_2} Tc_2 \\ &= \left(\frac{I_{si_{12}}}{I_{cr_{12}} R_1^2} - 1\right)\frac{R_1}{R_2} T_{c_2} + \left(1 - \frac{I_{si_{12}}}{I_{cr_{12}} R_1^2}\right) R_1 R_d T_s. \end{aligned}$$

But from Section 3.1, we know that

$$I_{si_{12}} = I_{t_1} - I_t = I_{si} + R_1^2 I_{cr} + \frac{R_1^2}{R_2^2} I_{ci} \quad \text{and} \quad I_{cr_{12}} = I_{cr} + \frac{I_{si}}{R_1^2} + \frac{I_{ci}}{R_2^2}.$$

It follows that $I_{si_{12}} = R_1^2 I_{cr_{12}}$ and consequently,

$$RT_{sp_1} \equiv 0.$$

Thus, the condition $RT_{sp_1} \geq 0$ is always true and the guard condition for the current state reduces to $c_{1,slip} \geq 0$. From (2), it can be shown that the guard condition for the current state is equivalent to

$$c_{1,slip} \geq 0 \iff \omega_t - \omega_{si} > -0.5 \iff \omega_t - \frac{1}{R_1}\omega_{cr} > -0.5 \tag{36}$$

The above guard condition is linear in the continuous states.

6. *torque21*

From Figure 2, the guard condition for this state is

$$Tc_1 > T_t \wedge Tc_2 \leq 1.$$

We first analyze the condition $Tc_1 > T_t$. From (17) we have that $\omega_{si} = \omega_t$. It follows from (2) that $c_{1,slip} = 0$ and $sgn(c_{1,slip}) = 1$. Since $pc_1 = pc_{1,max}$ for both possible values of to_gear, we have from the above results and (3) that

$$Tc_1 = c_1 \mu_1 A R_1 pc_{1,max}.$$

The above expression for Tc_1 and the expression for T_t in (1) do not depend on any of the states of the model. Therefore, we try to evaluate the truth value of the expression $Tc_1 > T_t$ with the given parameter values and the range of input variable tps. Using the parameter values given in Section 3.1,

$$Tc_1 = 0.1316 \times 2.912 \times 1000 = 383.2192.$$

The range for tps is $[0, 100]$, which implies that the maximum value of T_t is $1.7 \times 100 + 30 = 20$. This maximum value is well below the value of Tc_1 shown above. Therefore, the truth value for the condition $Tc_1 > T_t$ is always *true* and we may omit this condition from the guard condition.

From the state machine, the destination state for the state *torque21* is the state *first*. This implies that $to\_gear = 1$. Consequently, we have from (7) and (9) that $pc_{2,target} = 0$ and

$$pc_2 = pc_{2,filter}. \tag{37}$$

We now consider 3 cases for the quantity $\omega_t - \omega_{ci}$ which is equivalent to $(1 - \frac{R_1}{R_2})\omega_t$ for the current state.

(a) $\omega_t - \omega_{ci} \geq 0 \iff \left(1 - \frac{R_1}{R_2}\right)\omega_t \geq 0.$

In this case, we have that $sgn(c_{2,slip}) = sgn(\omega_t - \omega_{ci}) = 1$ and $|\omega_t - \omega_{ci}| = \omega_t - \omega_{ci} = \left(1 - \frac{R_1}{R_2}\right)\omega_t$. Thus, we have from (5) and (37) that

$$\begin{aligned} Tc_2 &= \left(c_2\mu_2\left(1 - \frac{R_1}{R_2}\right)\omega_t + c_2\mu_1\right) AR_2 pc_{2,filter} \\ &= c_2\mu_2\left(1 - \frac{R_1}{R_2}\right) AR_2 pc_{2,filter}\omega_t + c_2\mu_1 AR_2 pc_{2,filter} \end{aligned} \tag{38}$$

(b) $-0.5 < \omega_t - \omega_{ci} < 0 \iff -0.5 < \left(1 - \frac{R_1}{R_2}\right)\omega_t < 0.$

In this case, we have $sgn(c_{2,slip}) = sgn(0) = 1$ and $|\omega_t - \omega_{ci}| = -(\omega_t - \omega_{ci}) = -\left(1 - \frac{R_1}{R_2}\right)\omega_t$. Thus, we have from (5) and (37) that

$$\begin{aligned} Tc_2 &= \left(-c_2\mu_2\left(1 - \frac{R_1}{R_2}\right)\omega_t + c_2\mu_1\right) AR_2 pc_{2,filter} \\ &= -c_2\mu_2\left(1 - \frac{R_1}{R_2}\right) AR_2 pc_{2,filter}\omega_t + c_2\mu_1 AR_2 pc_{2,filter} \end{aligned} \tag{39}$$

(c) $\omega_t - \omega_{ci} \leq -0.5 \iff \left(1 - \frac{R_1}{R_2}\right)\omega_t \leq -0.5.$

In this case, we have $sgn(c_{2,slip}) = sgn(\omega_t - \omega_{ci}) = -1$ and $|\omega_t - \omega_{ci}| = -(\omega_t - \omega_{ci}) = -\left(1 - \frac{R_1}{R_2}\right)\omega_t$. Thus we have from (5) and (37) that

$$\begin{aligned} Tc_2 &= -\left(-c_2\mu_2\left(1 - \frac{R_1}{R_2}\right)\omega_t + c_2\mu_1\right) AR_2 pc_{2,filter} \\ &= c_2\mu_2\left(1 - \frac{R_1}{R_2}\right) AR_2 pc_{2,filter}\omega_t - c_2\mu_1 AR_2 pc_{2,filter} \end{aligned} \tag{40}$$

From the 3 cases above, we can rewrite the guard condition $Tc_2 \leq 1$ in terms of the continuous states as

$$\begin{aligned} Tc_2 \leq 1 \quad \iff \quad & \left[\left(1 - \frac{R_1}{R_2}\right)\omega_t \geq 0 \wedge (38) \leq 1\right] \\ & \vee \left[-0.5 < \left(1 - \frac{R_1}{R_2}\right)\omega_t < 0 \wedge (39) \leq 1\right] \\ & \vee \left[\left(1 - \frac{R_1}{R_2}\right)\omega_t \leq -0.5 \wedge (40) \leq 1\right] \end{aligned} \tag{41}$$

Note that the expressions in (38), (39), and (40) are not linear in the continuous states. We address this problem in Section 5.2.3.

5.2.2 Shift Scheduler

As discussed in Section 5.1, all the *shift_pending* states in the shift scheduler in Figure 1 are to be eliminated. This leaves two guard conditions that depend on continuous variables, namely the guard conditions for the states *first_gear* and *second_gear*. Since these guard conditions are already functions of the continuous state *veh_speed*, there is no need to perform any continuous guard conversion for this state machine.

5.2.3 Replacing Nonlinear Guard Conditions

As noted earlier in Section 5.2, the guard conditions (26), (31), and (41) are not linear with respect to the continuous states. In particular, all of the above guard conditions involve the product $pc_{2,\mathit{filter}}\omega_t$. To eliminate this nonlinearity from the guard conditions, we augment the system with an extra state z as shown below.

$$z = pc_{2,\mathit{filter}}\omega_t$$

The derivative of this new state variable is given by

$$\dot{z} = \dot{pc}_{2,\mathit{filter}}\omega_t + pc_{2,\mathit{filter}}\dot{\omega}_t.$$

Replacing all occurrences of the product $pc_{2,\mathit{filter}}\omega_t$ with the state variable z, we have that the guard conditions (26), (31), and (41) are now linear in the continuous state variables.

Note that the above procedure for converting the nonlinear guard conditions is a general procedure. Any differentiable nonlinear guard condition can be converted into an equivalent linear guard condition by defining an extra state variable

$$v = g(x)$$

for each unique occurrence of the nonlinearity $g(x)$ where x is the original continuous state vector. The derivative of v is given by

$$\dot{v} = \frac{\partial g}{\partial x}\dot{x}.$$

Although any differentiable nonlinearity in the guard conditions can be handled by the above procedure, the procedure should be used sparingly since augmenting the system with extra state variables can significantly increase the computation time for the simulation and verification of the system.

5.3 Replacing Discrete Guard Conditions

The guard conditions that depend on discrete variables in the challenge problem model are used as a synchronization mechanism between the state machines in Figures 1 and 2. To eliminate this type of guard condition while preserving the semantics of the model, we do the following. For each synchronization, we find a continuous guard condition that triggers a transition in one of the synchronized states and replace the discrete guard condition in each of the other synchronized states by this continuous guard condition.

The above procedure is applied to the guard conditions that depend on discrete variables in each state of the shift scheduler state machine in Figure 1 as follows.

1. *transition12/shifting*

   For this state, the guard condition is $gear = 2$. This guard condition changes from *false* to *true* when the dynamic mode selection machine in Figure 2 makes a transition from state *inertia12* to *second*. Thus, we replace the guard condition $gear = 2$ by the guard condition that triggers the transition *inertia12* $\rightarrow$ *second* in Figure 2. In particular, we replace the condition $gear = 2$ by condition (32) on the transition out of this state.

2. *transition21/shifting*

   For this state, the guard condition is $gear = 1$. This guard condition changes from *false* to *true* when the dynamic mode selection machine in Figure 2 makes a transition from state *torque21* to *first*. Thus, we replace the guard condition $gear = 1$ by the guard condition that triggers the transition *torque21* $\rightarrow$ *first* in Figure 2. In particular, we replace the condition $gear = 1$ by condition (41) on the transition out of this state.

6 Final CheckMate Model

Figure 3 shows the CheckMate model resulting from the conversion procedure described in Section 5. The model has 3 main components, namely the *switched continuous system*, the *threshold blocks*, and the *finite state machines*. The switched continuous system produces the continuous state variables as its output. The dynamics of the continuous states depend on the values of the discrete states in the finite state machines. The continuous states can also be reset when a finite state machine enters a new discrete state through the *reset* signals. The continuous states are passed to the threshold blocks, which produce Boolean signals indicating whether or not certain thresholds are exceeded. These Boolean signals are provided as the event inputs that drive all the finite state machines in the model. The outputs of the state machines are integer signals indicating the current discrete states. These signals are then fed back to the switched continuous system block to drive the dynamics of the continuous states. The following sections describe each of three components in more detail.

6.1 Switched Continuous Dynamics

Table 3 shows the complete set of states in the final CheckMate model, which includes the augmented continuous state z described in Section 5.2.3. The continuous dynamics for all 6 continuous states is implemented in the block *Switched Continuous System with Resets* in Figure 3. This block is one of the fundamental blocks in CheckMate. The switching function that describes the continuous dynamics and reset condition for each combination of the discrete states is implemented in the MATLAB m-file shown in the Appendix. It is straight-forward, albeit tedious, to see the direct connection between the MATLAB code and the equations in Sections 3 and 5.2.3.

Continuous States	Discrete States
T_s	$shift\_schedule$
veh_speed	$dynamic\_mode$
$pc_{2,filter}$	
ω_t	
ω_{cr}	
z	

Table 3: Continuous and Discrete States in CheckMate Model

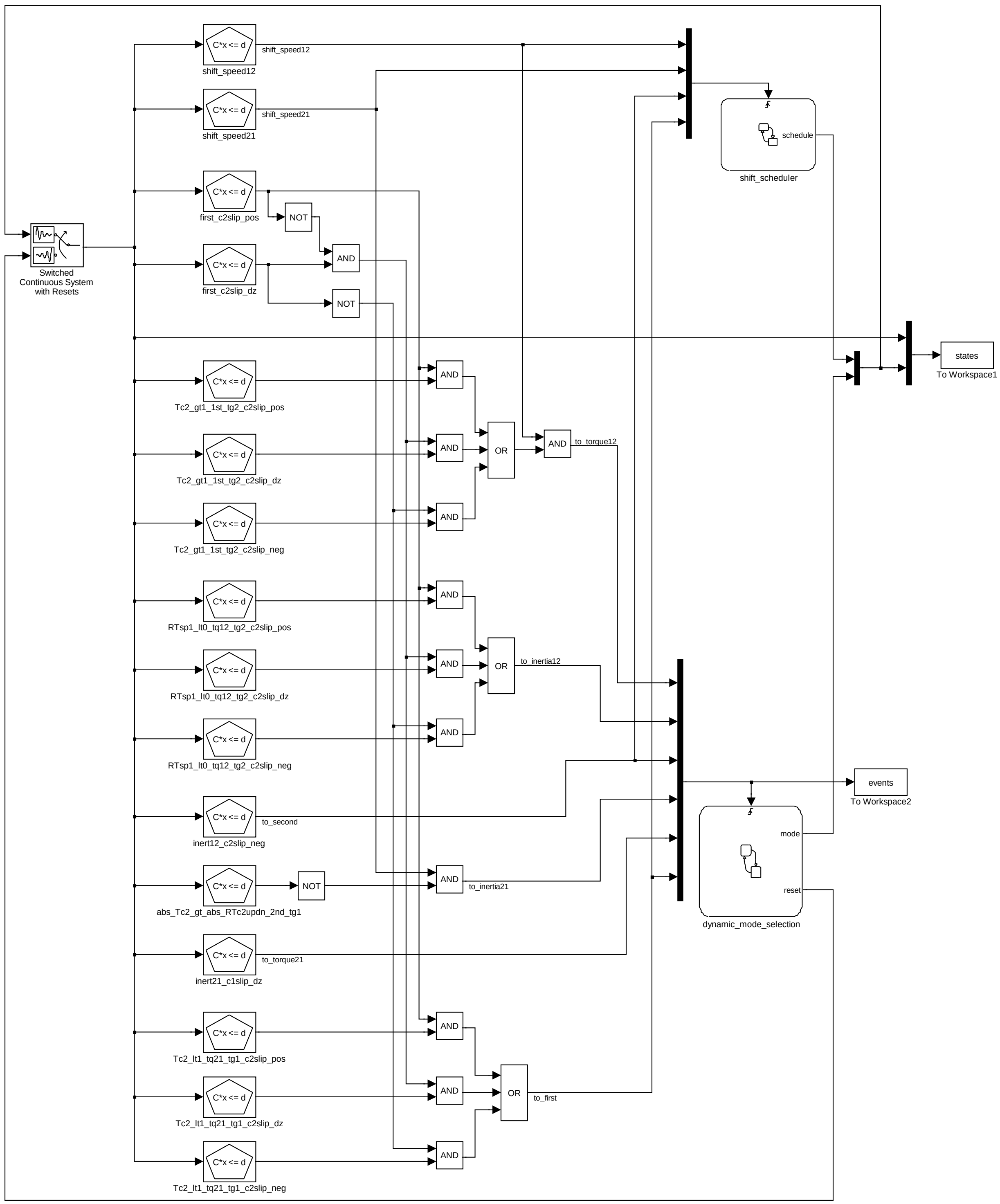

Figure 3: Final Model in CheckMate

6.2 Threshold Blocks

The CheckMate block library contains a fundamental block called the *polyhedral threshold block* that can be used to generate a Boolean signal indicating whether the continuous states lie within a polyhedral region. A change in this Boolean signal indicates an event that the continuous state trajectory is entering or leaving a region. Such an event can be used to drive the transition in the finite state machines connected to the threshold block.

Table 4 shows the correspondence between each of the partial guard conditions in Section 5.2 and a block in Figure 3. Since CheckMate does not handle strict inequality in the guard conditions, all strict inequalities (<,>) have been replaced by non-strict inequalities (≤,≥). The Boolean signals for these partial guard condition are passed through the logical operator blocks to form the Boolean signal for the final guard conditions that drive the finite state machines.

Condition	Threshold Block
$veh\_speed \geq shift\_speed12$	shift_speed12
$veh\_speed \leq shift\_speed21$	shift_speed21
$\left(1 - \frac{R_1}{R_2}\right)\omega_t \geq 0$	first_c2slip_pos
$\left(1 - \frac{R_1}{R_2}\right)\omega_t \geq -0.5$	first_c2slip_dz
$(22) > 1$	Tc2_gt1_1st_tg2_c2slip_pos
$(23) > 1$	Tc2_gt1_1st_tg2_dz
$(24) > 1$	Tc2_gt1_1st_tg2_neg
$(28) \leq 0$	RTsp1_lt0_tq12_tg2_c2slip_pos
$(29) \leq 0$	RTsp1_lt0_tq12_tg2_c2slip_dz
$(30) \leq 0$	RTsp1_lt0_tq12_tg2_c2slip_neg
(32)	inert12_c2slip_neg
(34)	abs_Tc2_gt_abs_RTc2updn_2nd_tg1
(36)	inert21_c1slip_dz
$(38) \leq 1$	Tc2_lt1_tq21_tg1_c2slip_pos
$(39) \leq 1$	Tc2_lt1_tq21_tg1_c2slip_dz
$(40) \leq 1$	Tc2_lt1_tq21_tg1_c2slip_neg

Table 4: Partial Guard Conditions and Corresponding Threshold Blocks

6.3 Finite State Machines

The finite state machines in Figures 1 and 2 in the original model have been converted into the finite state machines in Figures 4 and 5, respectively. As seen from the figures, the structure of the original state machines are largely preserved. The two *shift_pending* states have been eliminated from the shift scheduler machine for the reasons discussed in Section 5.1. Only one discrete (integer) output variable is allowed in each state machine by CheckMate to indicate the current state of the state machine. All other discrete variables in the original state machines have been eliminated. These variables are reconstructed as needed elsewhere in the model. For example, the variable to_gear has been reconstructed in the switching function found in the Appendix based on the value of the current state of the shift scheduler machine.

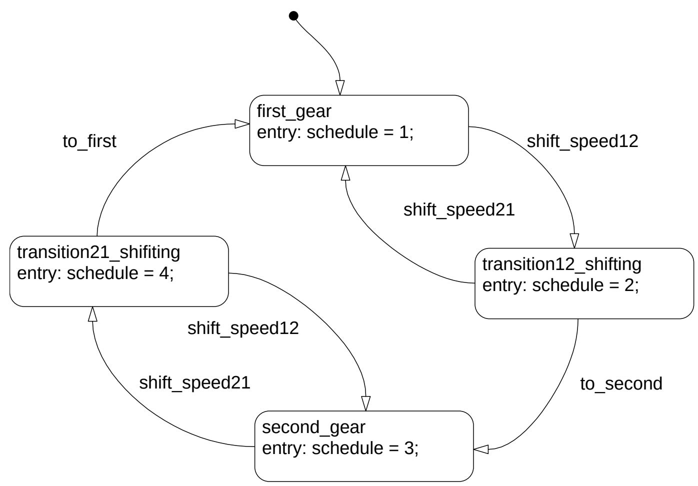

Figure 4: Shift Scheduler in CheckMate

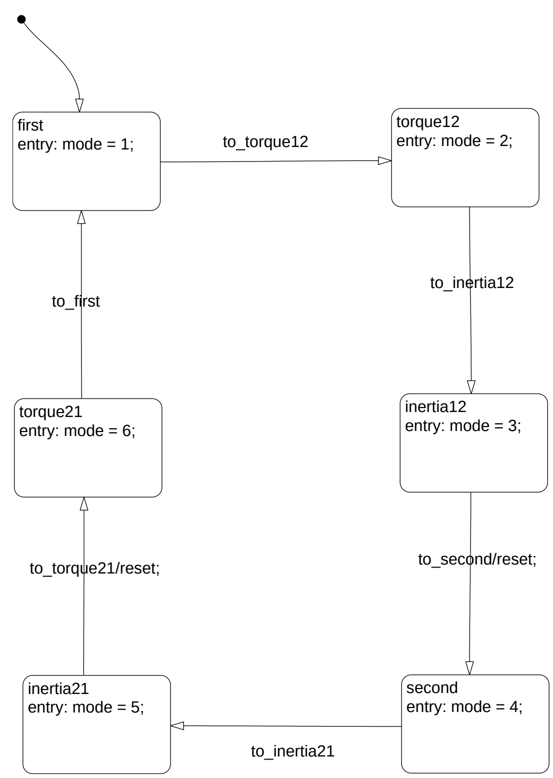

Figure 5: Transmission Dynamic Mode Selection in CheckMate

All finite state machines in CheckMate must be strictly event driven. In other words, a data signal cannot be used to trigger a transition in a state machine. Thus, all guard conditions in the original model, which are implemented as data signals, have been converted into event signals. Specifically, the Boolean signal for each guard condition is supplied as a rising-edged event signal to the state machines. Each Boolean signal carries the same name as the corresponding event signal in Figure 3. Since each event signal is rising-edged, the transition is taken when the corresponding guard condition changes from *false* to *true*. Table 5 shows the correspondence between the guard conditions and the event signals in the state machines.

Guard Condition	Event Name
$veh\_speed \geq shift\_speed12$	shift_speed12
$veh\_speed \leq shift\_speed21$	shift_speed21
(26)	to_torque12
(31)	to_inertia12
(32)	to_second
(35)	to_inertia21
(36)	to_torque21
(41)	to_first

Table 5: Guard Conditions and Corresponding Events

6.4 Model Validation

With the inputs tps and $grade$ held constant at 70% and 0.1745 rad, respectively, the simulation results from the CheckMate model are plotted against the results from the original model in Figure 6. The simulation results are from 0 to 60 seconds. As seen from the plots, the responses of the CheckMate model agree well with those of the original model.

7 Discussion

7.1 Modeling Difficulties in CheckMate

As seen from Section 5, entering the challenge problem model into CheckMate has not been an easy exercise even though we do not need to approximate the continuous dynamics of the model. In particular, the following are the main difficulties encountered during the model conversion process.

1. *Complexity of Guard Conditions.*

   Requiring the guard conditions to be functions of only the continuous state variables results in the guard conditions that are large and difficult to understand. To obtain linear guard conditions, nonlinearities such as the absolute value operations require further analyses that result in even more complex guard conditions. In some instances we were forced to rely on the analyst to recognize and exploit relationships between model variables to admit reformulation of the guard conditions.

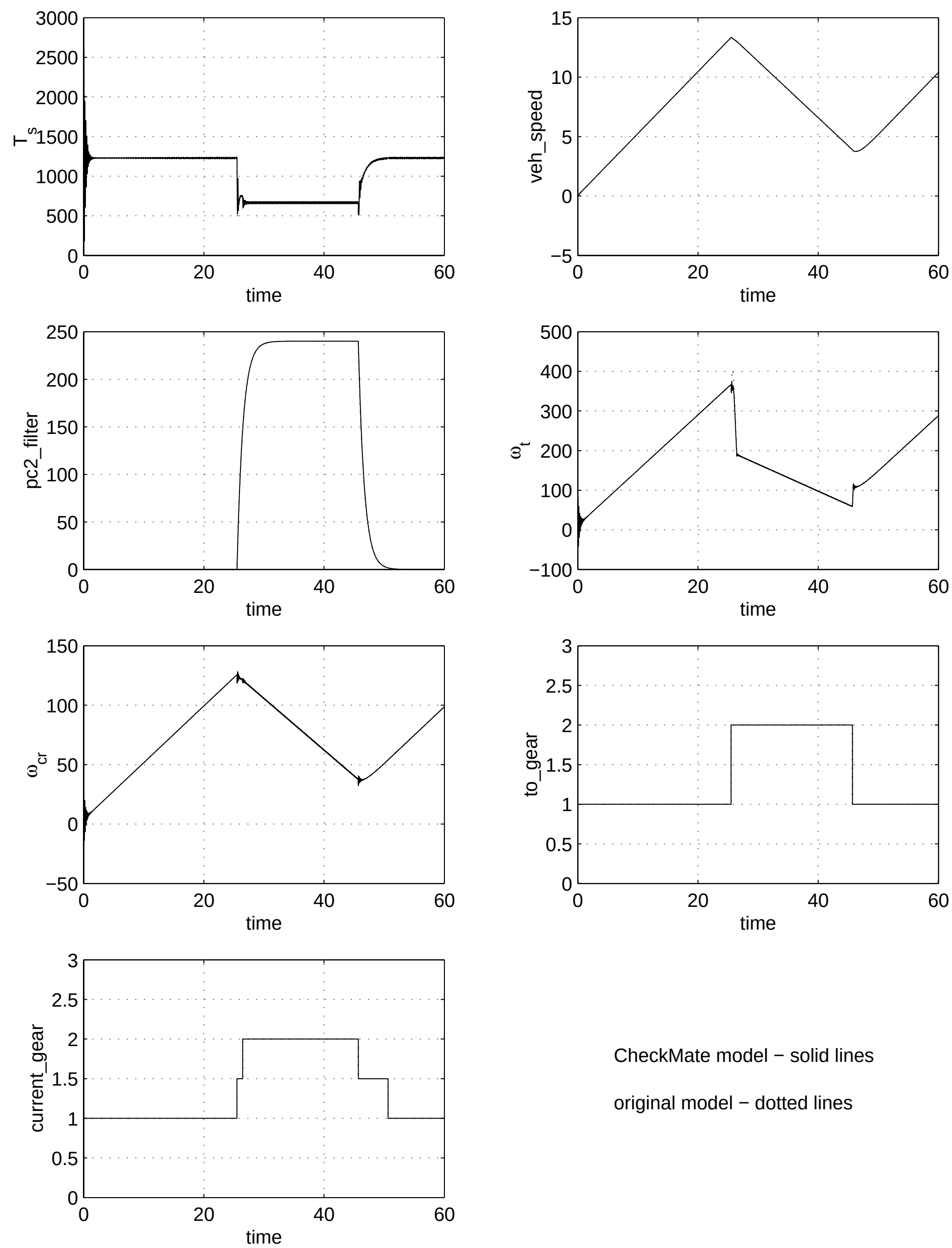

Figure 6: CheckMate Model Validation

2. *Increased System Order.*

   Although any differentiable nonlinearities in the guard conditions may be eliminated by augmenting the system with extra continuous states as discussed in Section 5.2.3, adding extra continuous states is not desirable. It is well known that the computational cost of the system analysis increases rapidly with the number of continuous states.

3. *Loss of Model Hierarchy.*

   As described in Section 6, CheckMate requires that the model be reorganized into the three main components, namely the continuous dynamics, the threshold event generators, and the discrete dynamics. Since CheckMate does not support hierarchical modeling, all of these components must reside at the root level of the Simulink model. Thus, the model hierarchy information in the challenge problem model is lost as a result of this reorganization. The loss of hierarchical information combined with the increased complexity of the guard conditions makes it difficult to relate the CheckMate model to the original model without the help of this report.

7.2 Assessment of CheckMate

Having identified the difficulties encountered during the model conversion process, we now give our assessment of CheckMate with respect to the requirements stated in Section 2.

First, all of the above difficulties can be seen as a direct result of not satisfying Requirement 2. None of the above problems would have arisen if we can simply use the original challenge problem model as the starting point for our analysis. Given the current state of the hybrid system research, it is unlikely that we will have a tool that can handle the challenge problem model directly any time soon. Nevertheless, we would like to encourage the researchers to keep this requirement in mind when developing their respective tools.

To emphasize the importance of Requirement 1, we note that the validation of the CheckMate model in Section 6.4 would have been much more difficult without the simulation capability.

As for Requirement 3, CheckMate has nicely leveraged the GUI capability in Simulink/Stateflow for the most part. Nevertheless, the continuous dynamics must be entered textually as an m-file. While this may not be a big problem if one already has the model equations, the difficulty arises if one has to reverse engineer an existing Simulink model to obtain the equations. In fact, this is precisely what has been done for our challenge problem model. The user friendliness of CheckMate would be significantly enhanced if the user could "cut and paste" portions of block diagrams from existing Simulink models to specify the continuous dynamics.

As mentioned in the introduction, the actual analysis of the challenge problem model is still an ongoing task. Thus, we cannot give any assessment of CheckMate with respect to Requirements 4 and 5 in this report. The assessment of CheckMate with respect to these requirements as well as its user friendliness in performing the analysis will be presented in our next baseline report.

References

[1] E. Asarin, O. Bournez, T. Dang, and O. Maler. Approximate reachability analysis of piecewise linear dynamical systems. In N. Lynch and B.H. Krogh, editors, *Hybrid Systems: Computation and Control (HSCC'00),* LNCS 1790, pages 20–31. Springer, 2000.

[2] E. Asarin, O. Bournez, T. Dang, O. Maler, and A. Pnueli. Effective synthesis of switching controllers for linear systems. In *Proceedings of the IEEE: Special Issue on Hybrid Systems*, July 2000.

[3] A. Bemporad, F.D. Torrisi, and M. Morari. Optimization-based verification and stability characterization of piecewise affine and hybrid systems. In N. Lynch and B.H. Krogh, editors, *Hybrid Systems: Computation and Control (HSCC'00)*, LNCS 1790, pages 45–58. Springer, 2000.

[4] Johan Bengtsson, Kim G. Larsen, Fredrik Larsson, Paul Pettersson, Yi Wang, and Carsten Weise. New Generation of UPPAAL. In *Int. Workshop on Software Tools for Technology Transfer*, June 1998.

[5] O. Botchkarev and S. Tripakis. Verification of hybrid systems with linear differential inclusions using ellipsoidal approximations. In N. Lynch and B.H. Krogh, editors, *Hybrid Systems: Computation and Control (HSCC'00)*, LNCS 1790, pages 73–88. Springer, 2000.

[6] Kenneth R. Butts. *SmartVehicle* challenge problems. In *http://vehicle.me.berkeley.edu/mobies/*, 2001.

[7] C.Daws, A.Olivero, S.Tripakis, and S.Yovine. The tool KRONOS. In *Hybrid Systems III,* LNCS 1066, pages 208–219. Springer, 1996.

[8] T.A. Henzinger, P.-H. Ho, and H. Wong-Toi. HYTECH: A model checker for hybrid systems. *Springer International Journal of Software Tools for Technology Transfer*, 1(1/2):110–122, October 1997.

[9] T.A. Henzinger, B. Horowitz, R. Majumdar, and H. Wong-Toi. Beyond HYTECH: Hybrid System Analysis Using Interval Numerical Methods. In N. Lynch and B.H. Krogh, editors, *Hybrid Systems: Computation and Control (HSCC'00)*, LNCS 1790, pages 130–144. Springer, 2000.

[10] T.A. Henzinger and S. Sastry, editors. *Hybrid Systems: Computation and Control (HSCC'98).* LNCS 1386. Springer, 1998.

[11] S. Kowalewski and H. Treseler. VERDICT - A tool for model-based verification of real-time logic process controllers. In 5^{th} *Int. Workshop on Parallel and Distributed Real-Time Systems (WPDRTS'97)*. IEEE Comp. Society Press, 1997.

[12] N. Lynch and B.H. Krogh, editors. *Hybrid Systems: Computation and Control (HSCC'00).* LNCS 1790. Springer, 2000.

[13] Paul Pettersson and Kim G. Larsen. UPPAAL2k. *Bulletin of the European Association for Theoretical Computer Science*, 70:40–44, February 2000.

[14] B. Silva, K. Richeson, B.H. Krogh, and A. Chutinan. Modeling and verifying hybrid dynamic systems using CHECKMATE. In *Proc. 4^{th} Int. Conf. On Automation of Mixed Processes*, pages 323–328, 2000.

[15] B.I. Silva, O. Stursberf, B.H. Krogh, and S. Engell. An assessment of the current status of algorithmic approaches to the verification of hybrid systems. In *Proceedings of the 40th IEEE Conference on Decision and Control*, December 2001.

[16] S.Yovine. KRONOS: A verification tool for real-time systems. *Springer International Journal of Software Tools for Technology Transfer*, 1(1/2), October 1997.

[17] F.D. Torrisi, A. Bemporad, and D. Mignone. HYSDEL — A tool for generating hybrid models. Technical Report AUT00-03, Automatic Control Lab, ETH, Zurich, Switzerland, 2000.

[18] F.W. Vaandrager and J.H. Van Schuppen, editors. *Hybrid Systems: Computation and Control (HSCC'99).* LNCS 1596. Springer, 1999.

[19] D. Weil, E. Closse, M. Poize, P. Venier, J. Pulou, S. Yovine, and J. Sifakis. TAXYS : A tool for developing and verifying real-time properties of embedded systems. In *The 13^{th} Conference on Computer-Aided Verification (CAV01)*, 2001.

[20] Q. Zhao and B.H. Krogh. Formal verification of statecharts using finite-state model checkers. In *Proceedings of the 2001 American Control Conference*, 2001.

Appendix

A Switched Continuous Dynamics

```
function [sys,type,reset] = hsprb_dynamics(x,u)

% Switched continuous dynamics function for the MoBIES hybrid system
% analysis challenge problem.
%
% Syntax:
%   "[sys,type,reset] = hsprb_dynamics(x,u)"
%
% Description:
%   Returns the continuous state derivatives, the type of system dynamics,
%   and the reset vector for the continuous states as function of "x", the
%   current continuous state vector, and "u", the discrete input vector to
%   the switched continuous system.

type = 'nonlinear';
if any(u == 0)
  reset = x; sys = [0 0 0 0 0 0]';
  return
end

% ================
```

```
% Model Parameters
% ================

M  = 1644.0+125.0; % Vehicle mass (kg)
Hf = 0.310;        % Static ground-to-axle height of front wheel (m)
Iwf = 2.8;         % Front wheel inertia(both sides) (kg-m^2)
Ks = 6742.0;       % Driveshaft spring constant (Nm/rad)

Rsi = 0.2955; % Input sun gear ratio
Rci = 0.6379; % Input carrier gear ratio
Rcr = 0.7045; % Reaction carrier gear ratio
Rd = 0.3521;  % Final drive gear ratio

R1 = Rci*Rsi/(1-Rci*Rcr); % 1st gear ratio
R2 = Rci; % 2nd gear ratio

AR2 = 4.125;
c2_mu1 = 0.1316;         % mu2 = c2_mu1 + c2_mu2*fabs(c2slip)
c2_mu2 = 0.0001748;

It = 0.05623;         % Turbine inertia (kg-m^2)
Isi = 0.001020;       % Input sun gear inertia (kg-m^2)
Ici = 0.009020;       % Input carrier gear inertia (kg-m^2)
Icr = 0.005806;       % Reaction carrier gear inertia (kg-m^2)

It1 = It + Isi + R1*R1*Icr + R1*R1/R2/R2*Ici;
It2 = It + Ici + R2*R2*Icr + R2*R2/R1/R1*Isi;

Icr12 = Icr + Isi/R1/R1 + Ici/R2/R2;

c2_table.y = [0 1 1 1];

Pc2max = 400.0;     % kPa

m_s_to_km_h = 3.6;  % conversion factor between m/s and km/h; unitless
pc2_torque_phase = .4; % ratio of Pc2max to be applied initially as pressure
                       % offset

% ===============================
% Model Inputs (Assumed Constant)
% ===============================
tps = evalin('base','tps');
grade = evalin('base','grade');

% =================
% Continuous States
% =================
Ts = x(1);
veh_speed = x(2);
pc2_filter = x(3);
wt = x(4);
wcr = x(5);
z = x(6);

% ===============
% Discrete Inputs
% ===============
gear_schedule = u(1);
dynamic_mode = u(2);

% Gear schedule state enumerations.
FIRST_GEAR = 1;
TRANSITION12_SHIFTING = 2;
SECOND_GEAR = 3;
TRANSITION21_SHIFTING = 4;

% Dynamic mode enumerations.
FIRST = 1;
```

```
TORQUE12 = 2;
INERTIA12 = 3;
SECOND = 4;
INERTIA21 = 5;
TORQUE21 = 6;

% ====================================================
% Compute variables that depends on continuous states.
% ====================================================
if ismember(dynamic_mode,[FIRST,TORQUE12,TORQUE21])
  wci = R1/R2*wt;
elseif ismember(dynamic_mode,[INERTIA12,INERTIA21])
  wci = 1/R2*wcr;
elseif ismember(dynamic_mode,[SECOND])
  wci = wt;
end

% ==================================================
% Compute variables that depends on discrete states.
% ==================================================
if ismember(gear_schedule,[FIRST_GEAR,TRANSITION21_SHIFTING])
  to_gear = 1;
elseif ismember(gear_schedule,[SECOND_GEAR,TRANSITION12_SHIFTING])
  to_gear = 2;
end

% =====================================
% Compute other intermediate variables.
% =====================================

% Engine torque.
Tt = 1.7*tps+30;

% Clutch pressures.
pc2_target = Pc2max*c2_table.y(to_gear);
pc2 = pc2_filter + pc2_torque_phase*pc2_target;

% Torques.
c2slip = wt-wci;
if (c2slip > -0.5) & (c2slip < 0.5)
  sgn2 = 1;
else
  sgn2 = sign(c2slip);
end
Tc2 = sgn2*(c2_mu2*abs(c2slip)+c2_mu1)*AR2*pc2;

% ==========================
% Compute State Derivatives.
% ==========================

Ts_dot = Ks*(Rd*wcr-veh_speed/Hf-0.001*Ts);
veh_speed_dot = (Ts/Hf-M*9.81*sin(grade))/(M+2*Iwf/(Hf*Hf));
pc2_filter_dot = -pc2_filter + (1-pc2_torque_phase)*pc2_target;

if ismember(dynamic_mode,[FIRST])
  wt_dot = 1/It1*(Tt-R1*Rd*Ts);
elseif ismember(dynamic_mode,[TORQUE12,TORQUE21])
  wt_dot = 1/It1*(Tt-R1*Rd*Ts-(1-R1/R2)*Tc2);
elseif ismember(dynamic_mode,[INERTIA12,INERTIA21])
  wt_dot = 1/It*(Tt-Tc2);
elseif ismember(dynamic_mode,[SECOND])
  wt_dot = 1/It2*(Tt-R2*Rd*Ts);
end

if ismember(dynamic_mode,[FIRST,TORQUE12,TORQUE21])
  wcr_dot = R1*wt_dot;
elseif ismember(dynamic_mode,[INERTIA12,INERTIA21])
  wcr_dot = 1/Icr12*(Tc2/R2-Rd*Ts);
```

```
elseif ismember(dynamic_mode,[SECOND])
  wcr_dot = R2*wt_dot;
end

z_dot = pc2_filter_dot*wt + wt_dot*pc2_filter;

sys = [Ts_dot
       veh_speed_dot
       pc2_filter_dot
       wt_dot
       wcr_dot
       z_dot];

% =====================
% Compute State Resets.
% =====================

% The reset is applied upon entering the discrete state.
if ismember(dynamic_mode,[FIRST,TORQUE12,TORQUE21])
  wcr_reset = R1*wt;
elseif ismember(dynamic_mode,[INERTIA12,INERTIA21])
  wcr_reset = wcr;
elseif ismember(dynamic_mode,[SECOND])
  wcr_reset = R2*wt;
end

reset = [Ts
         veh_speed
         pc2_filter
         wt
         wcr_reset
         z];

return
```