Formal Verification of a MEMS Based Adaptive Cruise Control System

Jairam S* Kusum Lata** Subir K Roy* Navakanta Bhat†

* SDTC, TI India, (sjairam,subir)@ti.com
** CEDT, IISc India, lkusum@cedt.iisc.ernet.in
†ECE IISc India, navakant@ece.iisc.ernet.in

ABSTRACT

A formal verification approach is presented for MEMS based adaptive cruise control (ACC) system. The system consists of a MEMS based gyroscope for measuring speed. The ACC system and the MEMS component are first modeled as a hybrid system, and then validated using a discrete time domain dynamic simulation approach in the Simulink/Stateflow (SS) framework from Mathworks. For its validation using a formal approach, CheckMate [1] a public domain formal verification tool for hybrid systems is used. In this paper we outline our experiences and highlight several issues faced in using CheckMate to carry out a formal analysis. The key contributions of the paper include 1) Formulation of realistic properties to enable formal analysis 2) Techniques to model an open hybrid system in CheckMate (it accepts only closed hybrid systems for formal analysis). 4) Transformations in SS models of the ACC and MEMS gyroscope needed to conform to the CheckMate model. 5) Description of changes necessary in the CheckMate methodology to enable formal analysis. 6) Optimization of the ACC system parameters using formal runs in CheckMate to identify fail-safe regions of operation. 7) Selection of MEMS gyroscope topologies based on optimized ACC system parameters.

Keywords: Hybrid Systems, MEMS Gyroscope, Simulation, Formal and Semi-Formal Verification

1 Introduction

With the design of hybrid systems becoming increasingly complex their validation to ensure fail safe (or safety-critical) behavior is becoming a challenging task. Automated methods based on formal analysis are the only route by which safety criticality can be guaranteed in such systems [2]. This is specially true of embedded hybrid system controllers targeting the automotive domain. Hybrid systems are characterized by continuous time differential equations which work concurrently with discrete time digital systems. Modeling of such hybrid system involves modeling both the discrete behavior, as well as, the continuous time or dynamic behavior. Most approaches to modeling hybrid systems is based on extending finite automata used for modeling discrete behavior to include simple continuous behavior. Validation for safety critical behavior implicitly involves determining the reachable set of states of the hybrid system on this model, and ensuring that it never reaches a state space representing unsafe operation.

In this paper paper we propose a formal verification approach for validation of MEMS based hybrid systems. The methodology is demonstrated on an adaptive cruise control (ACC) system consisting of a MEMS based gyroscope for measuring speed. The organization of the paper is as follows. In Section 2 the ACC system and the MEMS component are first modeled as a hybrid system in the SS framework. The safety critical behavior of the ACC system is validated in this framework through time domain simulation. In Section 3, we introduce the formal analysis approach based on CheckMate [1]. The safety critical behavior of ACC is captured by a set of properties which are then formally verified in CheckMate.CheckMate imposes the following restriction for performing formal analysis. It assumes a hybrid system to be closed. Our ACC system model is open. While the SS framework easily allows modeling of an open hybrid system, it needs some effort to model this in CheckMate. The MEMS gyroscope model in SS uses several continuous time domain dynamic components which do not belong to the set of dynamic components allowed by CheckMate. To enable formal analysis of the MEMS based gyroscope ACC system in CheckMate, we circumvent this problem through the use of a Look Up Table (LUT) to macromodel the gyroscope. In Section 4 we describe the generation and integration of this LUT in CheckMate, as well as, show how an open system can be modeled in it. In Section 5, we present our results, discuss several modeling issues faced in the formal validation process, extensions to the proposed work related to optimization of the ACC system parameters and selection of MEMS gyroscope topologies for these parameters.

2 ACC System Description, Modeling & Simulation Setup

Figure 1 shows the state transition diagram of the MEMS based ACC system. The system behavior consists of four states, *viz*, '$HALT$', '$ACCELARATE$',

'$CRUISE$' and '$RETARD$'. The variables x_p (Proximity of the tracking vehicle to the leading vehicle) and v (velocity of the tracking vehicle) govern the assignments to different states and the transitions between these states. Our ACC system model can easily be seen to be open with respect to the velocity of the leading vehicle. In the ACC system, the control actions depends on the behavior of the leading vehicle resulting from changes in its velocity V_L [3]. As can also be noted in the ACC system model, though x_p is a derived system variable, it is nevertheless treated as an independent system variable, as it too causes control actions to be initiated. Thus, x_p is a system level input to the controller, while the velocity of the tracking vehicle is an intra-system input, sensed by the MEMS based gyroscope whose output is an input to the ACC system. The differential equations corresponding to the $ACCELARATE$ and $RETARD$ states are $\dot{v} = A$ and $\dot{v} = R$, respectively.

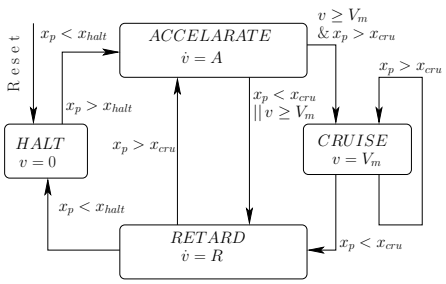

Figure. 1 State Transistion Graph for ACC

We give below, a brief summary of the MEMS gyroscope model. More details can found in [4] and in [7]. Figure 2 illustrates a first order implementation of a gyroscope. The output is modeled as a capacitance, which is seen to be a function of input motion [5]. Thus, the analytical model gives us the value of the capacitance as a function of vehicular angular velocity. The model transfer function $H(s)$ can be computed by abstracting out the system parameters and retaining only the input and output variables. This results in the following expressions:

$$h_1(s) = \mathcal{K}\frac{A\omega^2}{s^2+\omega^2}\,\frac{s}{(Ms^2+B_ds+K_d)} \quad (1)$$

$$h_2(s) = \frac{1}{(Ms^2+B_ss+K_s)} \quad (2)$$

where $\mathcal{K}$ is a constant, M is the proof-mass, $B_{s|d}$ and $K_{s|d}$ are the respective damping and stiffness co-efficients for drive and sense modes respectively.The output response for an input angular rate in time and frequency domain can be written down as [7]:

$$O(s) = [\Omega(s) \star h_1(s)]h_2(s) \quad (3)$$

$$O(t) = [\Omega(t)\, h_1(t)] \star h_2(t) \quad (4)$$

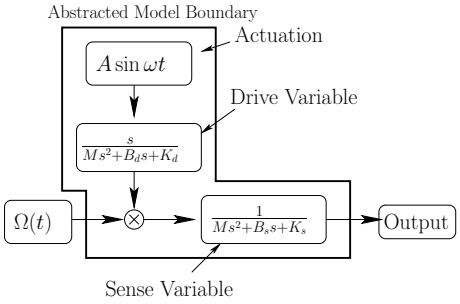

Figure. 2 Mathematical Model of a Gyroscope

The above mathematical model of the MEMS gyroscope and the complete ACC system is implemented in the SS framework as shown below in Figure 3. This setup is used to validate the specifications by real time analysis. The safety critical properties (described in Section 7) are initially analysed using time domain simulation in this platform.

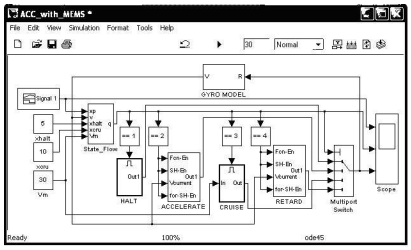

Figure.3 SS Model for ACC

3 Formal Analysis in CheckMate

We give a brief overview of the proposed formal verification approach carried out in CheckMate. For more details on CheckMate we refer to [6]. We then explain the implementation of the ACC system in CheckMate, with our proposed modifications.

For the formal analysis, CheckMate identifies the continuous set of reachable state space for a given set of different dynamics associated with a hybrid system. This is achieved by constructing a flow-pipe using different trajectories over time of the hybrid system originating from a well defined minimal set of initial states chosen from a given initial continuous state space set. This flow pipe represents the set of reachable points in the vector space of state variables of the hybrid system. It then approximates this flow-pipe with overlapping linear polyhedrons [6]. Formal analysis of the ACC system with respect to each property is carried out by using the approximate flow pipe region and the region defined by a property being validated. The formal analysis in CheckMate is performed in two stages; *viz. Explore* and *Verify*. The *Explore* phase performs time domain simulation in SS to store the different trajectories needed to construct a flow pipe. The construction of the flow pipe and its approximation along with property verfication computational geometry based algorithms is carried out in the *Verify* phase.

NSTI-Nanotech 2008, www.nsti.org, ISBN 978-1-4200-8505-1 Vol. 3

4 ACC Model in CheckMate system

CheckMate accepts a restrictive hybrid automata model, known as polyhedral invariant hybrid automata (PIHA) [6]. This requires transformation of the general SS model into a restrictive Simulink/Stateflow model by using a subset of its models/blocks allowed by CheckMate to create the equivalent PIHA model for formal analysis. The MEMS gyroscope model in SS uses several continuous time domain dynamic components which do not belong to the set of dynamic components allowed by CheckMate.

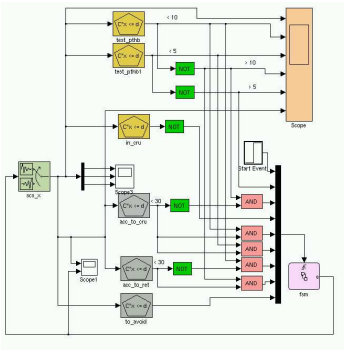

Figure. 4 ACC CheckMate Model

To enable formal analysis of the MEMS based gyroscope ACC system in CheckMate, we circumvent this problem through the use of a Look Up Table (LUT) to macromodel the gyroscope. The data points for the LUT is obtained for a range of velocity values by carrying out dynamic simulation on an exact macro-model of the MEMS gyroscope in the SS framework (Figure 3). To integrate the LUT macro-model of the MEMS gyroscope in CheckMate it is necessary to make changes in its implementation code in Matlab. The LUT is accessed through function calls in CheckMate to get the desired outputs of the gyroscope to obtain the trajectories needed in the formal analysis of the ACC system.

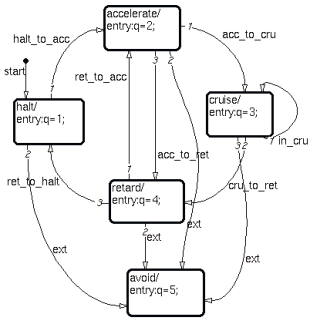

Figure. 5 ACC CheckMate State Transistion Graph

Figures 4 and 5 represent the CheckMate model for the ACC system and its associated state transition graph. The LUT based gyroscope macro-model was included between the switched dynamic block and the PTHB modules in Figure 4. One restriction imposed by CheckMate is that for formal analysis it assumes a hybrid system to be closed. Our ACC system model can easily be seen to be open with respect to the velocity of the leading vehicle V_L, as the control actions in the hybrid automata of the ACC system (Figure 1), depends on the behavior of the leading vehicle resulting from changes in its velocity V_L. While the SS framework easily allows modeling of an open hybrid system, it needs some effort to model this in CheckMate. We model such a scenario by addition of a redudant equation in terms of V_L, V_T and proximity (x_p) in which we render V_L as a parameter (V_T and x_p as the closed system state variables). The equation used is, $\int (V_L - V_T)dt = x_p$.

5 Results, Modeling Issues and Extensions

Table 1 lists the properties that were verified formally in CheckMate. Some of these properties are related to checking for safety critical conditions, such as checking of cruise velocity limit and tracking vehicle velocity limits within halt range.

Table 1: System Property Summary

Functional Specification	Property Validation
$R = 0 : x_p > x_{cru}$	Pass
$A = 0 ; x_p < x_{halt}$	Pass
$State \neq CRUISE : x_p < x_{xcruise}$	Pass
$x_p > 0 \; \forall \; t$	Pass

The sensing of velocity introduces error in the control process. This can be comprehended either as an error between the state velocity and the engine sensed velocity, or as a delay in the measured velocity. We use an error approximation to capture this effect of the gyroscope. This also provides another motivation for the design of the ACC system to be more realistic. The reset feature in CheckMate invoked in the sink state of a transition edge, at the end of a state transition, can easily hide and abstract out real life behavior which maybe needed in the optimization route based on the formal analysis. We show below, how we can avoid using this feature and still be able to obtain authentic PIHA models by transforming the hybrid automata. We validate such transformations to the ACC system model by both, simulation and formal analysis.

From the formal analysis results obtained from the proposed model a property was found to be failing. This was mainly because of a step change in velocity from a non-zero value to zero due to a transition from *RETARD* to *HALT* state. As a result of this analysis, the Stateflow graph shown in Figure 5, was refined by addition of

an extra retardation state (Figure 6). A lower bound on the retardation rate of the dynamics in this intermediate state was computed. This ensures that the vehicle actually retards to zero velocity before transitioning to the *HALT* state, while at the same time maintains the proximity constraint associated with the *HALT* state. The CheckMate model of the ACC system shown in Figure 6, was constructed for the hybrid automata model shown in Figure 1. The simulation results for this property is shown in Figure 7. The formal property was able to capture a faulty state that captures a crash phenomenon. The refined hybrid automata as shown in Figure 6 was analysed both, formally and with simulation. The property passed in both forms of analysis.

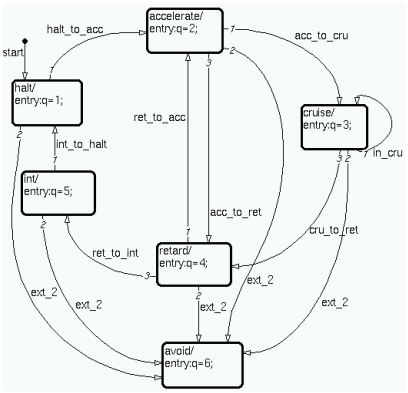

Figure. 6 Refined FSM for ACC CheckMate STG

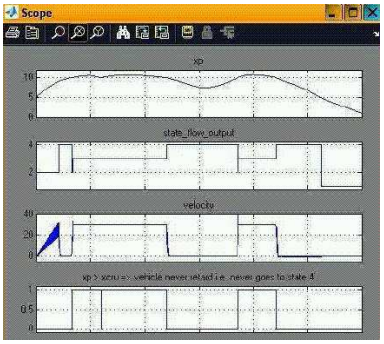

Figure 7 SS Output of ACC for Different Properties

We next, show an extension of our approach in which we optimize the ACC system parameters. We do this by using formal runs in CheckMate to identify fail-safe regions of operation for a given continuous set of initial states. Based on the formal analysis, the lower bound on the retard rate could be computed. The vehicle should halt before the proximity reduces below x_{halt}. We introduce a parameter η, in a new intermediate state which is a function of x_{halt}. The vehicle retards through this state such that at $x = x_{halt}$, velocity is zero. This translates the velocity constraint for the intermediate state as, $V_{final}^2 = V_{initial}^2 - 2R\eta$. Now the upper bound on $V_{initial}$ is V_{cru} and V_{final} should tend to zero at the end of the state. Hence retardation rate can be constrained as $R \geq V_{cru}^2/2\eta$. Hence if a suitable constraint for η is selected (eg. $\eta = 2x_{halt}$) a lower bound on the retardation rate for the new state can be computed. As a result the other parameters of the system (*viz.* accelaration, retardation rates and cruise limits) could be re-computed.

These same parameters can be directly translated to frequency requirements of the MEMS gyroscope that needs to measure this velocity. In [7] it is shown that system parameter variations can result in different topology selection of gyroscopes. This helps in selecting appropriate MEMS gyroscope topologies for the optimized ACC system parameters as the choice of system parameters can be directly mapped to compiler based approaches for topology synthesis of gyroscopes [7]. Given these parameters we translated this into a topology selection problem, through the framework presented in [7].

6 CONCLUSIONS

In this paper we proposed a formal verification approach for MEMS based embedded systems. The approach was demonstrated on a MEMS based adaptive cruise control system. The MEMS component is a gyroscope used for speed measurement. CheckMate, a public domain formal verification tool, was modified and deployed to include MEMS based components for formal analysis. The approach was also used to make topology selections of gyroscopes based on ACC system requirements.

REFERENCES

[1] Checkmate Carnegie Mellon University Website. http://www.ece.cmu.edu/ webk/checkmate/

[2] "Claire J. Tomlin et al., Computational Techniques for the Verification of Hybrid Systems, Proc. IEEE, 91,7, pp 986-1001, 2003.

[3] Jairam S, Subir K Roy and Navakanta Bhat, A Transformation Based Method for Formal Analysis of Hybrid Systems, Proc. VDAT, Aug 2007, Kolkata India.

[4] Gary Fedder et. al, Integrated Microelectromechanical Gyroscopes, Journal of Aerospace Engg, 6,2, pp 65-75, 2002.

[5] Mohit S et. al, Design, modelling and simulation of vibratory micromachined gyroscopes, Journal of Physics Conference Series, Apr 34, pp 757-763, 2006.

[6] Chutinan A and Krogh B. H,, Computational techniques for hybrid system verification, IEEE Tran. on Automatic Control, 48, 1, pp 64-75, 64-75, 2003

[7] Jairam S and Navakanta Bhat,GyroCompiler: A Soft-IP Model Synthesis and Analysis Framework for Design of MEMS based Gyroscopes, International Conf. on VLSI Design, pp 589-594, Jan 2008.