

Multi-mode Sampling Period Selection for Embedded Real Time Control

Rajorshee Raha, Soumyajit Dey, Partha Pratim Chakrabarti, Pallab Dasgupta
Department of Computer Science & Engineering
Indian Institute of Technology Kharagpur, INDIA
{rajorshee.raha, soumya, ppchak, pallab}@cse.iitkgp.ernet.in

ABSTRACT

Recent studies have shown that adaptively regulating the sampling rate results in significant reduction in computational resources in embedded software based control. Selecting a uniform sampling rate for a control loop is robust, but overtly pessimistic for sharing processors among multiple control loops. Fine grained regulation of periodicity achieves better resource utilization, but is hard to implement online in a robust way. In this paper we propose multi-mode sampling period selection, derived from an offline control theoretic analysis of the system. We report significant gains in computational efficiency without trading off control performance.

General Terms

Real Time Control, Automotive Software

Keywords

Real Time Control, Automotive Software

1. INTRODUCTION

Embedded software-based control systems have traditionally been implemented by assuming fixed sampling rates and fixed task periods [3]. The sampling rate is derived from a control theoretic analysis [9] of the system in a manner that guarantees desired level of control performance at all reachable states of the system. A uniform period can be implemented robustly, since we can analyze the sampling rate of all the control tasks and then choose an appropriate computational infrastructure that can statically schedule a periodic execution of the software components of these control tasks. The schedule does not change during execution and hence the control performance is deterministic.

Recent studies confirm a widely accepted belief, namely that a uniform sampling rate is not a good choice when multiple control loops share a common computing resource such as an Electronic control unit (ECU). These studies establish

that the sampling period can be regulated to achieve significant benefits in computational performance without any trade off in control performance. In fact, it has also been shown that a non-uniform scheduling strategy can balance the sampling rates among the control loops sharing a ECU in such a way that the overall control performance improves [4, 5].

Uniform sampling rate is typically a pessimistic choice, since we need to choose a sampling rate that guarantees control performance at all reachable states of the system. It is often the case that the selected rate is necessary at only specific control states of the system, whereas at all other states a much lesser sampling rate suffices. Adaptive sampling derives its benefit from this fact by intelligently regulating the sampling period as needed to maintain the desired level of control performance [4].

Fine grained regulation of the sampling rate may theoretically determine the optimal balance between computational efficiency and control performance but such schemes are difficult to implement in practice due to non-determinism in timing introduced by the computational infrastructure (including message delays, execution time variations in different paths of the control software, etc). If the sampling rate is known a priori then it becomes possible to develop an appropriate schedule for all control tasks sharing a ECU with adequate consideration for these types of non-determinism.

In this paper we profess the use of coarse grained regulation of the sampling rate. Specifically we propose an approach where for each control loop, a limited number of sampling rates are chosen and a control theoretic analysis is used to determine the switching criteria between these modes. Therefore, for each control loop we have a set of *sampling states*, and an automaton that captures the switching between these sampling states. The *global sampling state* of the system is a concatenation of the sampling states of the control loops in the system. For each global sampling state, the exact schedule for the control tasks is precomputed considering all types of non-determinism arising out of the execution of software tasks. The reachable global sampling states are chosen based on available computational bandwidth and the relative priorities of the control loops.

The primary objective of this paper is to establish the benefit of using multiple discretely chosen sampling rates. In the process, we also present the following enabling contributions.

1. We outline the basis for choosing the various sampling rates based on use case analysis.
2. We present an analytical approach which determines

the criteria for switching between sampling rates so that control performance is not hampered.

3. We present the construction of an automaton based scheduler, which implements the switching between sampling rates of the controller.

We present the necessary background for the work in Section 2. We outline the methodology of multi-mode sampling period selection in Section 3. Subsequently, we use the control theoretic model of an Anti-lock Braking System (ABS) as a running example to validate our approach.

2. BACKGROUND STUDY

In this section, we outline the mathematical relation between the sampling period of a discrete time controller and its control stability. Any discrete time feedback control system can be represented as shown in Fig. 1 [2], here $x(t)$ is the input to the system, $e(t)$ is the error signal, $u(t)$ is the controller output and $y(t)$ is the plant output fed back to the controller using a sensor.

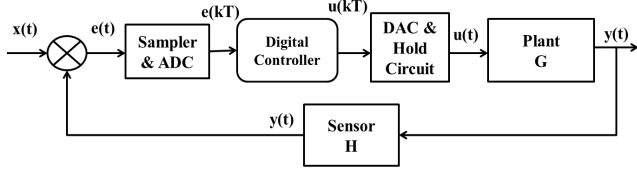


Figure 1: Discrete Time Control System

Generally, the sampling of the continuous signal is done at a constant rate T , which is known as the sampling period or interval. The sampled signal $e_k = e(kT)$, is the discretized signal with $k \in \mathbb{N}$. In general, for any transfer function, the control signal output depends on n previous control signal output instances and m previous error signal instances [2]. The situation may be represented as,

$$u_k = -a_1 u_{k-1} - a_2 u_{k-2} - \dots - a_n u_{k-n} + b_0 e_k + b_1 e_{k-1} + b_2 e_{k-2} + \dots + b_m e_{k-m} \quad (1)$$

The discrete signals u_{k-1}, u_{k-2}, \dots are the delayed versions of u_k by sampling period $T, 2T, \dots$ respectively. In Laplace domain, a time delay is introduced into a signal by multiplying its Laplace transform by the operator e^{-Ts} . Let the Laplace domain representation of u_k and e_k be $U(s)$ and $E(s)$ respectively [7]. Hence, u_{k-1}, u_{k-2}, \dots can be represented in Laplace frequency domain as $e^{-Ts}U(s), e^{-2Ts}U(s), \dots$ respectively and similarly for e_{k-1}, \dots . Thus, the corresponding Laplace domain representation of Eq. 1 shall be,

$$U(s) = -a_1 e^{-Ts}U(s) - a_2 e^{-2Ts}U(s) - \dots + b_0 E(s) + b_1 e^{-Ts}E(s) + b_2 e^{-2Ts}E(s) + \dots \quad (2)$$

Substituting e^{-Ts} with the discrete frequency domain operator z [2] and simplifying this further we get the discrete time transfer function of the controller $C(z)$ as,

$$\frac{b_0 z^n + b_1 z^{n-1} + \dots + b_m z^{n-m}}{z^n + a_1 z^{n-1} + \dots + a_n} = b_0 \frac{\prod_{j=1}^m (z - z_j)}{\prod_{i=1}^n (z - p_i)} z^{n-m} \quad (3)$$

where z_j are the zeros and p_i are the poles of the transfer function. Behavior of any discrete time controller can be observed by analyzing the poles and zeros of the corresponding transfer function [2]. The positions of the poles and zeros differ for different sampling intervals (T). Correspondingly, the control stability of the overall system gets effected. More related background is provided in Appendix A.

3. METHODOLOGY OUTLINE

In this section, we outline our proposed methodology of multi-mode sampling period selection for embedded real time control. The main steps of this proposed multi-mode methodology are as follows,

- Step I : Developing a control theoretic model of the corresponding system.
- Step II : Classification of different modes based on different control parameters and selection of best possible sampling rate for the corresponding operating modes.
- Step III : Construction of a supervisory automaton for controlling the mode switching.

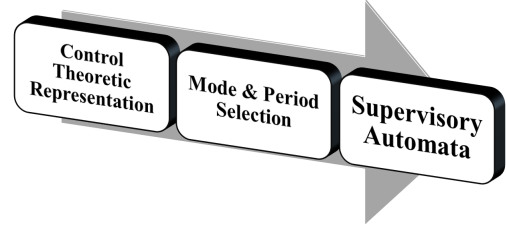


Figure 2: Methodology Outline

In the following sections we demonstrate the proposed methodology with an extensive analysis of ABS as a running example. In Section 4, we present the control model of the ABS. In Section 5, we divide the driving pattern of a vehicle into multiple modes parameterized by vehicular characteristics like velocity, brake pedal pressure and slip. For each such mode, we choose a sampling frequency which ensures stability guarantee of the ABS. In Section 6, we outline our approach for guard condition selection for switching between different modes, and synthesize a scheduler automaton which may supervise the mode selection depending on vehicular dynamics. In Section 7, we provide experimental results supporting the proposed approach.

4. CONTROL MODEL

ABS is an automobile safety critical driver assistance system which prevents the wheels from locking and avoids uncontrolled skidding. An abstract block diagram of a vehicle with ABS is shown in Fig. 5. For designing the vehicle we used a simplified quarter car model as shown in Fig. 3.

Here, m is the mass of the quarter vehicle, V_x is lateral speed of the vehicle, ω is the angular speed of the wheel, F_N is the vehicle vertical force, F_x is the frictional force transmitted to the road, M_b is the braking torque, R is the wheel radius and J_ω is the wheel inertia. Wheel slip λ is given as, $\lambda = 1 - \omega R / V_x$.

The effective braking force is dependent on the frictional force [1] transmitted to the road which is related to F_N as, $F_x = -\mu F_N$, where μ is the frictional coefficient of the road

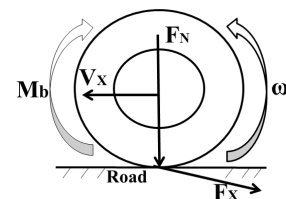


Figure 3: 1/4 Car Forces and Torques [1]

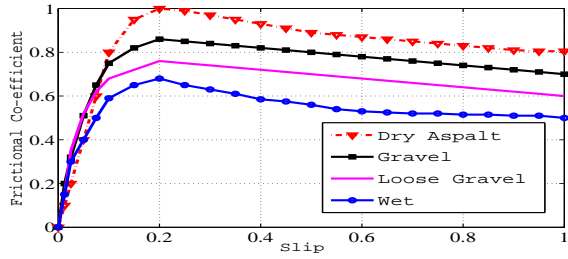


Figure 4: $\mu - \lambda$ Curve [10]

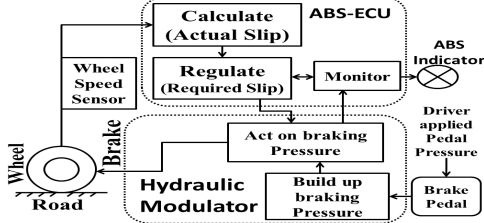


Figure 5: ABS Overview [1]

surface. Thus, as evident from Fig. 4, the amount of slip will vary depending on road conditions.

Relationship between wheel slip and frictional coefficient can be approximated, $\mu = f(\lambda)$, using a piecewise linear function [6] as,

$$\mu = \begin{cases} \alpha\lambda, & \lambda \leq 0.2 \\ -\frac{1}{2}\lambda + \frac{3}{4} + \beta, & \lambda > 0.2 \end{cases} \quad (4)$$

where, $\alpha \in [0, 8]$ and $\beta \in [-0.1, 0.1]$. The non-linear equations for designing a quarter car model can be given as,

$$\begin{aligned} \dot{V}_x &= -\frac{1}{m}F_N\mu \\ \dot{\omega} &= \frac{R}{J_w}F_N\mu - \frac{M_b}{J_w} \\ \dot{\lambda} &= -\frac{1}{V_x}\left[\frac{1}{m}(1-\lambda) + \frac{R^2}{J_w}\right]F_N\mu + \frac{1}{V_x}\frac{R}{J_w}M_b \end{aligned} \quad (5)$$

Using Taylor series expansion method [6, 8] for linearizing a nonlinear system we obtain a linear (affine) system description from Eq. 5 as,

$$\begin{aligned} \dot{x} &= A_l x + E_l + B_l u^* \\ y &= C_l x + D_l u^* \end{aligned} \quad (6)$$

where, $x^T = [V_x, \lambda]$, $u^* = M_b \cdot V_x$, $y = [\lambda]$, A_l , B_l , C_l , D_l are system input and output matrices respectively. E_l , $l = f(x)$ are the affine term and function telling the validation of linearization. The formation of this state space equation from the nonlinear equations are described in details in Appendix B. The objective of ABS controller is to decelerate the vehicle as fast as possible, while maintaining its steer ability by minimizing wheel slip. The main components of ABS are the ABS-ECU, hydraulic modulator, and wheel speed sensor. The ECU constantly monitors the wheel rotational speed through the wheel speed sensors, and also measures the actual slip. The controller's task is to maintain the braking torque within a certain range. The braking force is applied to the wheels by the hydraulic modulator. It rapidly pulses the brakes to prevent wheel lock up, even during panic braking in extreme conditions and promises shortest possible distance under most conditions. We can

design a discrete PID controller for this purpose as,

$$\begin{aligned} M_b &= K_p e + K_i \int edt + K_d \frac{de}{dt} \\ M_b(z) &= [K_p + \frac{K_i T}{z-1} + \frac{K_d(z-1)}{T}]E(z) \end{aligned} \quad (7)$$

Here K_p , K_i , K_d are the proportional, integral, and derivative gain respectively of the PID controller. T is the sampling interval. $M_b(z)$, $E(z)$ are the discrete domain representation of the the braking torque, M_b and the error signal, $e = \lambda_d - \lambda$, which is the difference between the desired slip (λ_d) and actual slip (λ). Observe that when $\lambda = \lambda_d$ then $e = 0$, i.e. $\dot{\lambda} = 0$. Substituting this in Eq. 5, M_b can be represented as,

$$M_b = [(\lambda - 1)\frac{J_w}{R} - mR]\dot{V}_x \quad (8)$$

Hence, it is obvious from Eq. 7 & 8 that the control performance, i.e. stability of the ABS controller will vary for different values of T , V_x and λ .

5. MODE AND PERIOD SELECTION

The candidate sampling modes and periods for a controller are determined by partitioning its input space based on the use-case scenarios and stability of the control law in those scenarios. For example, in ABS, the adequacy of a sampling rate in a given scenario depends on the urgency of braking (which is a function of vehicle speed and pedal pressure) and the slip ratio (which is a function of the vehicle speed and friction on the road). In general, we select the different possible sampling frequencies of the controller using the following approach.

1. Identify vehicular parameters which impact the controller output (i.e. braking torque in this case).
2. Identify multiple possible driving scenarios and corresponding ranges of vehicular parameters and also the probable driver response.
3. Perform stability analysis followed by identification of maximal acceptable sampling period in each driving scenario.

Steps 2 and 3 may have to be iterated to arrive at a gainful combination of sampling modes. A sampling mode is effective towards gaining computational efficiency only if the controller stays in that mode for a non-trivial period of time. Therefore it is necessary to relate the sampling modes with different use-case scenarios. In ABS, we estimate different possible traffic scenarios (city traffic, suburban or medium traffic & highway traffic) and the variation in traffic density, traffic regulations and corresponding average cruising speed and driver reaction to arrive at the sampling modes. We categorize the brake pedal pressure range as low, mild, medium and high, and also consider various speed ranges and slip ratios. For each of the scenarios, we determine the sampling rate at which the controller is stable. Through this study we selected three sampling modes, as outlined below:

- **NO Mode:** This mode requires the least sampling rate among the three chosen mode, and targets scenarios where the vehicle is cruising at low to medium speeds (such as in city traffic). Considering average cruising speed and the driver reaction, we arrive at the operating sampling rate

$T_s = 0.2ms$ in which the ABS achieves satisfactory control performance as given by Fig. 6. For a given velocity (X axis) and slip (Y axis), we carry out the standard unit circle analysis and plot the maximum magnitude among the different pole positions (Z axis) of the transfer function corresponding to $T_s = 0.2ms$. The stability variation for different slip values is due to the piecewise linear function given in Eq. 4. It may be observed from Fig. 6, all the poles are of magnitude ≤ 1 for velocity range $[0 \dots 85]km/h$ and slip range $[0 \dots 0.65]$ thus ensuring stable vehicular dynamics. Thus correspondingly brake pedal pressure variation range is selected as $[low, mild, medium]$ estimating the probability of the above mentioned slip range. We carry out similar analysis for the other cruising modes and derive satisfactory sampling intervals.

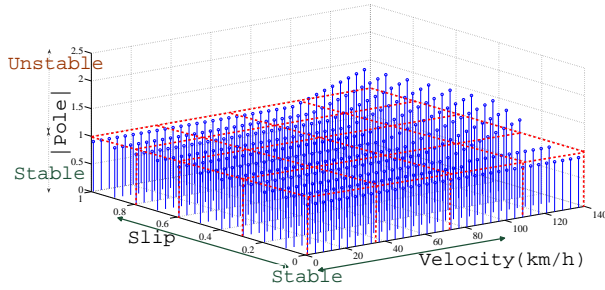


Figure 6: N0 Mode Stability Guarantee

• **N1 Mode:** This mode uses higher sampling rate than N0, and targets suburban traffic scenarios. The maximum sampling interval with stability guarantee is found to be $T_s = 0.15ms$ as shown in Fig. 7.

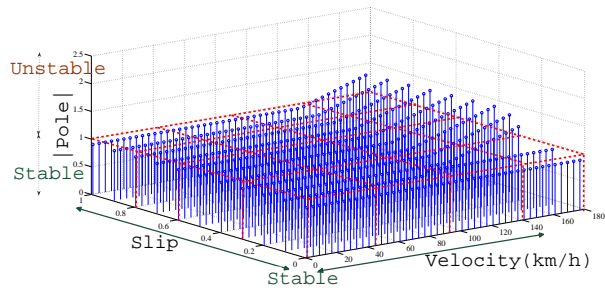


Figure 7: N1 Mode Stability Guarantee

• **E Mode:** This mode uses a sampling rate that is adequate in all scenarios. Existing approaches for choosing a uniform sampling mode will choose this sampling rate. We choose the sampling frequency for which the vehicle remains stable considering all velocity and brake pedal pressure variations as shown in Fig. 8. The corresponding sampling period for our model is found to be $T_s = 0.1ms$. We designate this as *emergency* sampling mode, in case of any driving irregularity the controller switches to this mode, thus ensuring vehicular stability.

Z-domain unit circle stability analysis is used to show the vehicular parameters and stability relationship graphically for extensive parameter ranges. Similar observations can be obtained mathematically using other nonlinear system stability [2] criteria like, Lyapunov, Nyquist, Routh-Hurwitz or Bode plot analysis. The effective braking pressure range is varied in each of these modes to achieve satisfactory performance, depending upon driving scenarios. The mode switching and detailed switching criterion are discussed in the next section.

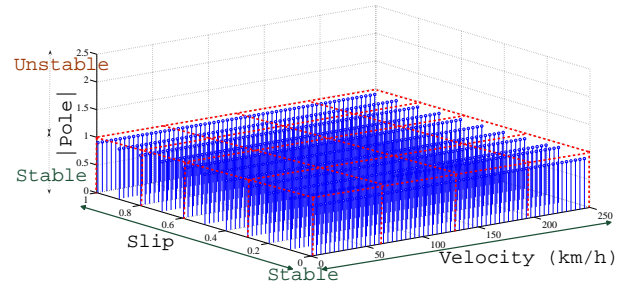


Figure 8: E Mode Stability Guarantee

6. SUPERVISORY AUTOMATA

Our analysis of the different possible driving scenarios and the choice of sampling periods entails the creation of a scheduler which may dynamically switch the controller among different sampling modes. Such a supervisory automaton is shown in Fig. 9. The criteria for switching between the three

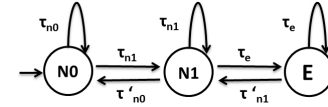


Figure 9: Scheduler Automata

modes of the automaton is chosen based on our observations about vehicular parameters vis-a-vis stability. Let “ v ” and “ bpp ” be the velocity and brake pedal pressure respectively. The brake pedal pressure is divided into the ranges low, mild, medium and high. High brake pedal pressure represents the probability of larger value of slip. Medium, mild and low brake pedal pressure signifies lesser value of slip. The guard conditions for switching between states are given as,

$$\begin{aligned}
 \tau_{n0} &= (v \in [0 \dots 85] \ \& \ bpp \in [low, mild, medium]) \\
 &\quad | (v \in [85 \dots 140] \ \& \ bpp \in [low, mild]) \\
 \tau'_{n0} &= (v \in [0 \dots 80] \ \& \ bpp \in [low, mild, medium]) \\
 &\quad | (v \in [80 \dots 135] \ \& \ bpp \in [low, mild]) \\
 \tau_{n1} &= (v \in [0 \dots 85] \ \& \ bpp \in [high]) \\
 &\quad | (v \in [85 \dots 140] \ \& \ bpp \in [medium, high]) \\
 &\quad | (v \in [> 140] \ \& \ bpp \in [low, mild]) \\
 \tau'_{n1} &= (v \in [0 \dots 80] \ \& \ bpp \in [high]) \\
 &\quad | (v \in [80 \dots 135] \ \& \ bpp \in [medium, high]) \\
 &\quad | (v \in [> 135] \ \& \ bpp \in [low, mild]) \\
 \tau_e &= (v \in [> 140] \ \& \ bpp \in [medium, high])
 \end{aligned}$$

The guard conditions for switching between ‘N0’ to ‘N1’ and vice versa are selected to be ‘ τ_{n1} ’ and ‘ τ'_{n0} ’ respectively. Similarly, for ‘N1’ to ‘E’ and vice versa, switching conditions are ‘ τ_e ’ and ‘ τ'_{n1} ’ respectively. The guard conditions are selected ensuring some amount of hysteresis while mode switching. For example, the automaton switches from mode ‘N0’ to mode ‘N1’ in case $bpp = high$ and $v \in [0 \dots 85]$. However, the automaton switches from mode ‘N1’ to mode ‘N0’ when $v \in [0 \dots 80]$ and bpp is not *high*. Similarly, the automaton switches from mode “N1” to E when bpp is *medium* or *high* and $v \in [> 140]$, however, the automaton switches from mode ‘E’ to mode ‘N1’ when $v \in [> 135]$ and bpp is *low* or *mild*.

Whenever the automaton makes a transition from a mode with lower sampling period to a mode with higher sampling period (e.g. E to N1), it ensures that the guard conditions are valid for a certain prefixed number of clock cycles. In that way, unwanted glitches due to faulty sensor readings are expected to be filtered out.

The main objective of synthesizing the scheduler automaton was to reduce ECU bandwidth requirement. Scheduling in E mode signifies a sampling periodicity of $0.1ms$, while the sampling periodicity of $N0$ and $N1$ modes are $0.2ms$ and $0.15ms$ respectively. If we notice the mode switching scenarios, we observe that when the car is cruising at a certain speed and no brake pedal pressure is applied, the controller is scheduled using infrequent sampling periods ($N0$ or $N1$). Further, in scenarios when the car is cruising at a certain speed and brake pedal pressure is applied, the mode switching will be supervised by the respective scheduler automaton ensuring that it will switch to a more frequent sampling mode. Thereby, in a general cruising scenario, a multi-mode controller ensures nearly 30% – 50% ECU bandwidth saving as shown in our simulation results.

7. RESULTS

The initial part of this section is devoted towards establishing the motivation for multi-mode sampling through experimental results. The latter part of the section demonstrates the gain in computational bandwidth and the benefit of effective sharing of computational resources between multiple controllers.

We analyze the performance of the ABS for different sampling intervals. Observing the μ vs λ curve in Fig. 4 carefully, we notice that the peak point on most of the road scenarios belong to the range $[0, 0.2]$. Hence, for effective braking with maximum possible road friction, we set $\lambda_d = 0.2$ as the desired slip. In our experimental setup, when brake force is applied with current velocity $V = 100km/h$, it is expected to gradually decrease until $V = 0km/h$ and throughout this deceleration phase the slip value should be as close as possible to the desired slip ($\lambda_d = 0.2$) thus ensuring smooth braking. Finally the slip value should be 1 (normalized slip) when the vehicle comes to rest.

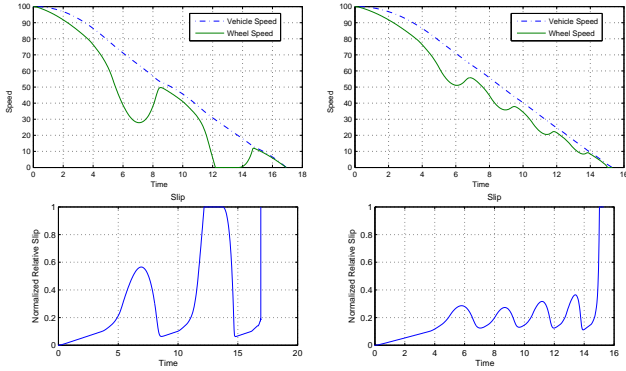


Figure 10: Slip Variation: the Left & Right column figures correspond to sampling time $T_{s1} = 1s$ & sampling time $T_{s1} = 0.01s$ respectively.

We observe that the slip varies significantly for two different sampling rates as given by Fig. 10. Observe from the left column in Fig. 10 that with a choice of moderate sampling rate, the slip varies drastically thus leading to undesired perturbations in vehicular speed while braking. However, as shown in the right column of the same figure, with the higher sampling rate, the slip exhibits a well damped trajectory around the desired value ($\lambda_d = 0.2$) leading to smoother vehicular deceleration.

Our notion of control performance is based on ensuring

the stability of the system. We empirically observe the variation of stability with sampling rate. For stability analysis, we used unit circle as well as bode plot analysis [2]. We take a moderate velocity, say $V = 60km/h$ and calculate the stability of the system for different sampling intervals. We show one such example Bode plot for stability analysis in Fig. 11. It may be observed that with the sampling interval $T_s = 0.01s$, the system is unstable. However, if we drastically reduce the sampling interval to $0.1ms$, the system becomes stable. In Appendix C.1 and C.2 more such results are given.

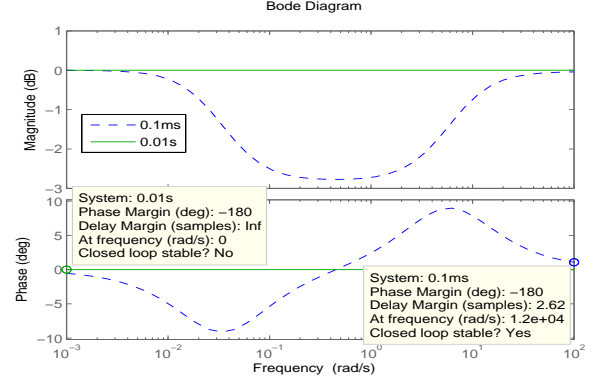


Figure 11: Stability Variation: $T_s = 0.01s$, Unstable & $T_s = 0.1ms$, Stable

We have employed our methodology of multi-mode sampling period selection for the ABS running example. We consider a braking scenario with the initial and final speed being $200km/h$ and $0km/h$ respectively and compare the minimum stopping distance achieved by our ‘Multi-mode’ ABS Controller with the existing Matlab model of ABS with fixed periodicity. The simulation results are provided in Table 1 considering different possible road surfaces. We ob-

Table 1: Stopping Distance in Kilometer

ABS Controller	Road Surface			
	Dry Asphalt	Gravel	Loose Gravel	Wet
Existing model	3.073	3.424	3.771	4.269
Multi-mode	3.080	3.433	3.795	4.289

serve that the stopping distance is nearly same for both the controllers. For this ‘panic braking’ scenario, the estimated percentage of time spent in each of E , $N1$ and $N0$ modes is highlighted in Fig. 12. It is evident from Fig. 12 that we can save significant amount of ECU bandwidth (30% - 50%) using a multi-mode controller as compared to the controller with fixed periodicity.

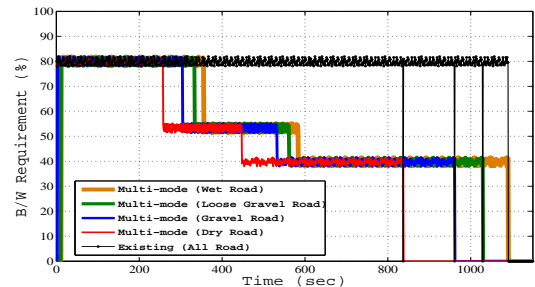


Figure 12: ECU, Panic Braking Scenario

Further, we investigate the utility of our multi-mode ABS controller in a general cruising scenario where the car is being driven in a speed range of 0km/h to 200km/h , in various traffic densities. The ECU bandwidth requirement for this scenario is shown in Fig. 13. It is evident that the multi-mode controller requires much lesser bandwidth compared to the existing controller since the supervisory automaton schedules the controller in the infrequent sampling modes for most of the time as described in Section 6. In that way, we can guarantee a significant amount of bandwidth saving which may be utilized for scheduling other tasks.

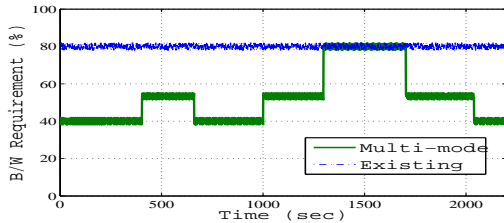


Figure 13: ECU, General Cruising Scenario

We further demonstrate the promise of multi-mode sampling for a multiple ECU scenario taking an Adaptive Cruise Control (ACC) system as a running example. ACC is an automobile safety critical driver assistance system which automatically adjusts the vehicle speed in order to maintain a safe distance from vehicles ahead. In case of an ACC system, the driver sets a safe cruising speed and a desired safe distance (from preceding vehicle) as controller inputs. The other inputs of an ACC controller which comes from the radar sensor are preceding vehicle speed and approximate distance from preceding vehicle as shown in Fig. 14.

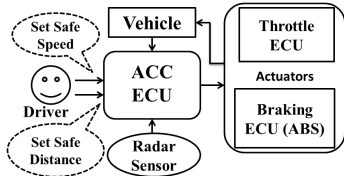


Figure 14: ACC System Overview [1]

ACC is a drive assist system for highway cruising which monitors the inputs and decides cruising speed or distance to lead vehicle. When the applied brake pedal pressure is high, the ACC is overridden by the braking controller (ABS). The operating modes of ACC controller can be classified into 'active', 'suspended' and 'idle' while our multi-mode ABS operates in three different modes as discussed previously. Generally, the ACC system and the braking controller are mapped to separate ECUs. We can achieve significant reduction in bandwidth requirement by sharing an ECU between these features.

When the ACC is suspended, because the applied brake pedal pressure is high, we schedule the ABS controller with frequent sampling rate and the ACC controller with relatively infrequent sampling rates. On the other hand, when the ACC is active, i.e. applied brake pedal pressure is low or null, then the ACC controller is scheduled with frequent sampling rate and the ABS controller is scheduled with relatively infrequent sampling rate. In Fig. 15, we show the bandwidth requirement of both ACC and ABS controllers when scheduled in a single ECU while providing satisfactory control performance.

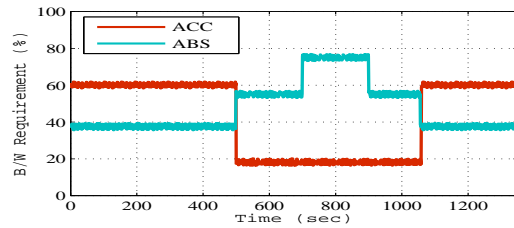


Figure 15: ECU Sharing: ACC-ABS

8. CONCLUSIONS

The present work provides a methodology for adaptively regulating the sampling rate of embedded software based controllers leading to significant reduction of computational resource requirement. Applying the methodology on single controller based systems like ABS has shown that 30% - 50% reduction in ECU bandwidth requirement is possible. Further, it was also shown that the method smoothly scales up for multiple controller based systems. Our future research shall focus on giving a sound formal underpinning to the method of creating multiple sampling modes for software based controllers and creating a tool flow which mechanizes the synthesis of such multi-mode controllers.

9. REFERENCES

- [1] *Safety, Comfort and Convenience Systems*. Robert Bosch GmbH, 2006.
- [2] K. Åström and B. Wittenmark. *Computer controlled systems: theory and design*. Prentice Hall, 1984.
- [3] G. Buttazzo. Research trends in real-time computing for embedded systems. *SIGBED Rev.*, 3(3).
- [4] A. Cervin, M. Velasco, P. Marti, and A. Camacho. Optimal online sampling period assignment: Theory and experiments. *IEEE Transactions on Control Systems Technology*, 19(4), 2011.
- [5] D. Henriksson and A. Cervin. Optimal on-line sampling period assignment for real-time control tasks based on plant state information. In *44th IEEE CDC-ECC*, 2005.
- [6] H. Khalil. *Nonlinear systems*. Macmillan Pub.Co., 1992.
- [7] B. Kuo. *Automatic control systems*. Prentice Hall, 1962.
- [8] M. Schinkel and K. Hunt. Anti-lock braking control using a sliding mode like approach. In *Proceedings of the ACC*, volume 3, 2002.
- [9] D. Seto, J. Lehoczky, L. Sha, and K. Shin. On task schedulability in real-time control systems. In *17th IEEE RTSS*, 1996.
- [10] J. Wong. *Theory of Ground Vehicles*. Wiley, 2001.

APPENDIX

A. BASIC CONTROL SYSTEM

Any continuous feedback control system[7] can be represented as shown in Fig. 16, where $x(t)$ is the input to the system, $e(t)$ is the error signal, $u(t)$ is the controller output and $y(t)$ is the plant output fed back to the controller using a sensor(H). Correspondingly, a Discrete time Feedback

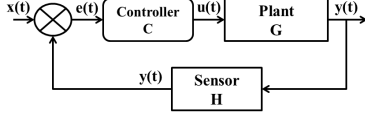


Figure 16: Continuous Time Control System

Control System can be represented as shown in Fig. 17, where the dotted box highlights the discretized controller [2]. Generally, the sampling of the continuous signal is done

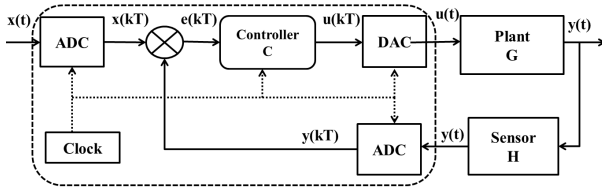


Figure 17: Discrete Time Control System

at a constant rate T , which is known as the sampling period or interval. The sampled signal $y_k = y(kT)$ is the discretized signal with $k \in \mathbb{N}$. For understanding the design of a discretized control system, let us first consider a simple continuous domain transfer function for the controller C in Fig. 16 given as follows.

$$C(s) = \frac{U(s)}{E(s)} = \frac{K(s+a)}{(s+b)}$$

The corresponding time domain representation shall be,

$$\frac{du}{dt} + bu = K\left(\frac{de}{dt} + ae\right)$$

The Euler's approximation of the first order derivative is represented as:

$$\frac{dx}{dt} \approx \frac{x_{k+1} - x_k}{\Delta t} \quad (9)$$

Applying Euler's approximation of first order derivative [2] on the continuous differential equation, we get the following discrete difference equation.

$$\frac{u_{k+1} - u_k}{\Delta t} + bu_k = K\left(\frac{e_{k+1} - e_k}{\Delta t} + ae_k\right)$$

Generally, Δt , K , a and b are fixed. The digital controller updates the control signal every cycle as per the following equation,

$$u_{k+1} = -a_1 u_k + b_0 e_{k+1} + b_1 e_k \quad (10)$$

where, $b_0 = K$, $b_1 = K(a\Delta t - 1)$, $a_1 = (b\Delta t - 1)$. In general, for any transfer function, the control signal output depends on n previous control signal output instances and m previous error signal instances which can be represented as[2],

$$u_k = -a_1 u_{k-1} - a_2 u_{k-2} - \dots - a_n u_{k-n} + b_0 e_k + b_1 e_{k-1} + b_2 e_{k-2} + \dots + b_m e_{k-m} \quad (11)$$

The discrete signals u_{k-1}, u_{k-2}, \dots are the delayed versions of u_k by sampling period $T, 2T, \dots$ respectively.

A.1 Stability: Discrete Time Control System

The stability property of this system can be defined from the impulse response of a system as

- Asymptotic stable system: The steady state impulse response is zero.

$$\lim_{k \rightarrow \infty} y(k) = 0$$

- Marginally stable system: The steady state impulse response is different from zero, but limited.

$$\lim_{k \rightarrow \infty} 0 < y(k) < \infty$$

- Unstable system: The steady state impulse response is unlimited.

$$\lim_{k \rightarrow \infty} y(k) = \infty$$

where $y(k)$ is the impulse response of the corresponding system. The impulse response for different stability property is illustrated in Fig. 18. Let us assume a control system with

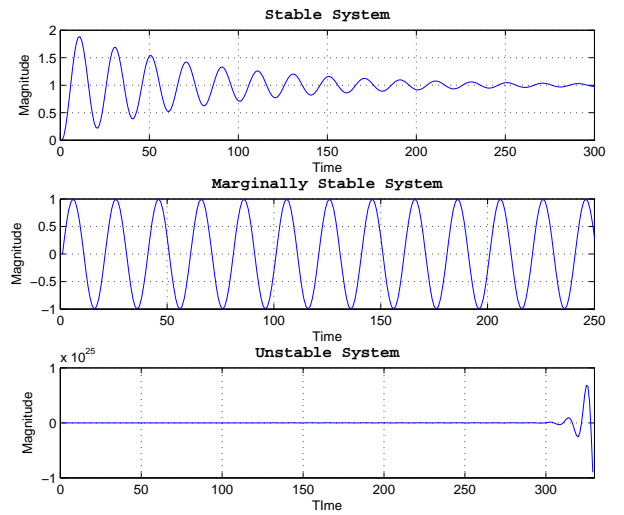


Figure 18: Impulse Response and Stability

input u and output y . The transfer function of any discrete time control system can be represented as

$$C(z) = \frac{y(z)}{u(z)} = \frac{bz}{(z-p)}$$

where p is the pole which is in general a complex number and can be written in polar form as $p = me^{j\theta}$ where m is the magnitude and θ is the phase. The impulse response of the system can be given as

$$y(k) = Z^{-1}\left\{\frac{bz}{z-p}\right\} = b|m|^k e^{jk\theta}$$

Thus, it is the magnitude m which determines if the steady state impulse response converges towards zero or not. The relationship between stability and pole placement can be stated as follows.

- Asymptotic stable system: All poles lie inside (none is on) the unit circle, or what is the same: all poles have magnitude less than 1.
- Marginally stable system: One or more poles but no multiple poles are on the unit circle.
- Unstable system: At least one pole is outside the unit circle.

The situation is graphically shown in Fig. 19.

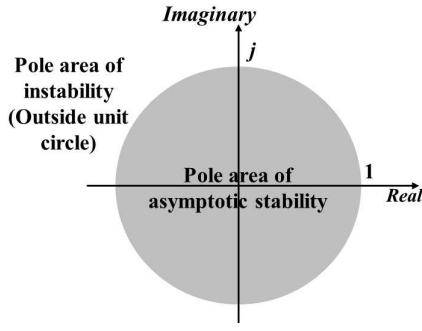


Figure 19: Unit Circle: Stability areas in Complex Plane

B. QUARTER VEHICLE MODELING

A quarter car model is shown in figure 20. Here, m is the

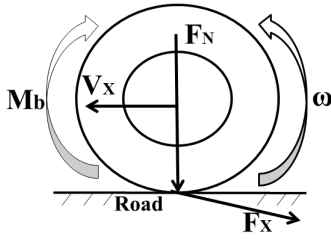


Figure 20: 1/4 car forces and torques

mass of the quarter vehicle, V_x is lateral speed of the vehicle, ω is the angular speed of the wheel, F_N is the vehicle vertical force, F_x is tire frictional force, M_b is the braking torque, R is the wheel radius and J_ω is the wheel inertia. Wheel slip λ is represented as:

$$\lambda = 1 - \frac{\omega R}{V_x}$$

The relationship between F_N and F_x [1] is given as:

$$F_x = -\mu(\lambda)F_N$$

where $\mu(\lambda)$ is the frictional coefficient of the road surface. The non-linear equations for designing a quarter car model can be given as[8]:

$$\begin{aligned} \dot{V}_x &= -\frac{1}{m}F_N\mu(\lambda) \\ \dot{\omega} &= \frac{R}{J_\omega}F_N\mu(\lambda) - \frac{M_b}{J_\omega} \\ \dot{\lambda} &= \frac{\dot{V}_x(1-\lambda) - \dot{\omega}R}{V_x} \\ \dot{\lambda} &= -\frac{1}{V_x}[\frac{1}{m}(1-\lambda) + \frac{R^2}{J_\omega}]F_N\mu(\lambda) + \frac{1}{V_x}\frac{R}{J_\omega}M_b \end{aligned}$$

For linearizing the system approximation using the Taylor series expansion can be expressed as [6]: $f(\lambda, V_x) \approx f(\lambda', V'_x) + \frac{df}{d\lambda}|_{\lambda', V'_x}(\lambda - \lambda') + \frac{df}{dV_x}|_{\lambda', V'_x}(V - V'_x)$. From this from this we can derive a linear (affine) system description as:

$$\begin{aligned} \dot{x} &= A_l x + E_l + B_l u^* \\ y &= C_l x + D_l u^* \\ l &= f(x) \end{aligned}$$

where, A_l , B_l , C_l , D_l are system input and output matrices respectively. E_l are the affine terms and $f(x)$ is the function telling the validity of linearization and $x^T = [V_x, \lambda]$, $u^* = M_b \cdot V_x$, $y = [\lambda]$. From Fig. 4 the relationship between

wheel slip and frictional coefficient can be approximated by using piecewise linear function as,

$$\mu(\lambda) = \begin{cases} \alpha\lambda, & \lambda \leq 0.2 \\ -\frac{1}{2}\lambda + \frac{3}{4} + \beta, & \lambda > 0.2 \end{cases} \quad (12)$$

where, $\alpha \in [4.8, 5.1, 5.46, 6.4]$ and $\beta \in [-0.1, 0.1]$.

$$A_l = \begin{bmatrix} 0 & -\alpha \frac{F_N}{V_x} \\ \alpha \frac{F_N R^2 \lambda'}{V_x^2 J_\omega} & \alpha \frac{F_N R^2}{V_x J_\omega} \end{bmatrix} \quad (13)$$

$$E_l = \begin{bmatrix} 0 \\ -\alpha \frac{F_N R^2 \lambda'}{V_x J_\omega} \end{bmatrix} \quad (14)$$

and for $\lambda > 0.2$

$$A_l = \begin{bmatrix} 0 & \frac{F_N}{4m} \\ (-\frac{\lambda'}{2} + \frac{3}{4}) \frac{F_N R^2}{V_x^2 J_\omega} \pm 0.1 \frac{F_N R^2}{V_x^2 J_\omega} & \frac{F_N R^2}{4V_x J_\omega} \end{bmatrix} \quad (15)$$

$$E_l = \begin{bmatrix} (-\frac{3}{4} \pm 0.1) \frac{F_N}{m} \\ (\frac{\lambda'}{2} - \frac{3}{2}) \frac{F_N R^2}{V_x J_\omega} \pm 0.2 \frac{F_N R^2}{V_x J_\omega} \end{bmatrix} \quad (16)$$

The stability analysis of this system can be done using Lyapunov, Bode, Nyquist, Unit Circle or Hurwitz stability criteria [8].

C. STABILITY V/S SAMPLING PERIOD

In this section we provide few examples relating to the variation in stability with respect to sampling periods.

C.1 Simple Examples:

Let the transfer function of any arbitrary system be,

$$U(s) = \frac{s+0.5}{ms^2+bs+u}$$

Case 1: Let $m=2$, $b=-0.5$, $u=1$. For $T_s=2s$ the system unstable but for $T_s=1s$ the system is stable. The values of m , b , u are same for both of the sampling intervals. The corresponding stability response is shown in Fig. 21.

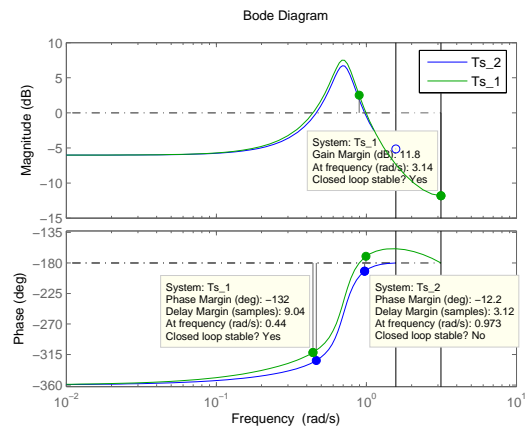


Figure 21: $T_s = 2s$, Unstable; $T_s = 1s$, Stable

Case 2: Let $m=5$, $b=1$, $u=10$. In this case the system stable for both the sampling intervals $T_s=2s$ and $T_s=1s$. The values of m , b , u are same for both of the sampling intervals. The corresponding stability response is shown in Fig. 22.

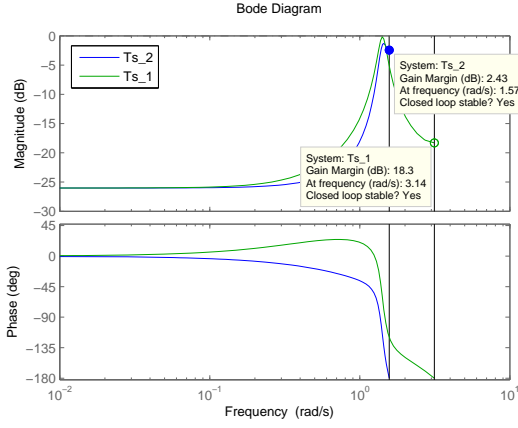


Figure 22: $T_s = 2s$, and $T_s = 1s$, Both Stable

C.2 ABS example

We provide more examples of stability variation with sampling interval for the ABS controller model.

Case 1: For $V = 10$ Km/h, $\lambda = 0.1$, the stability response is shown in Fig. 23.

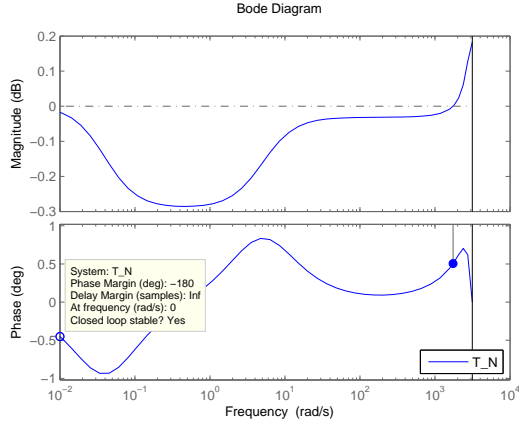


Figure 23: $T_s = 0.6$ ms, Stable

Case 2: For $V = 100$ Km/h, $\lambda = 0.6$, the stability response is shown in Fig. 24.

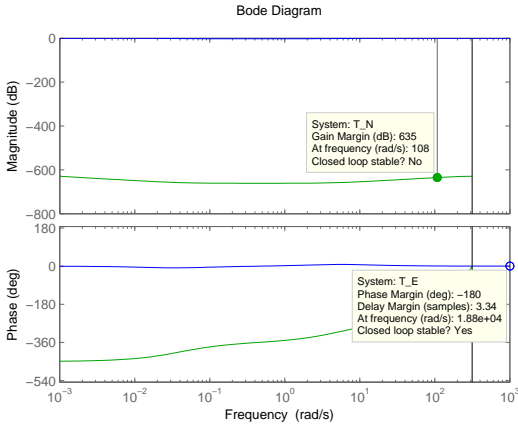


Figure 24: $T_s = 0.6$ ms, Unstable; $T_s = 0.1$ ms, Stable

Case 3: For $V = 40$ Km/h, $\lambda = 0.2$, the stability response is shown in Fig. 25.

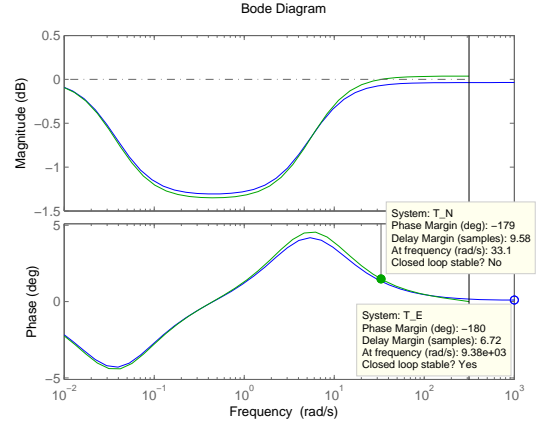


Figure 25: $T_s = 0.6$ ms, Unstable; $T_s = 0.3$ ms, Stable

Case 4: For $V = 15$ Km/h, $\lambda = 0.3$, the stability response is shown in Fig. 26.

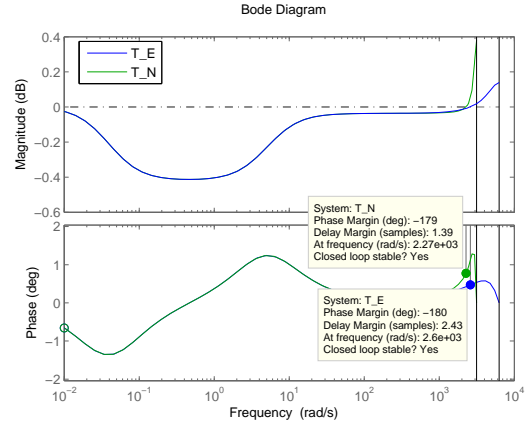


Figure 26: $T_s = 0.6$ ms, and $T_s = 0.1$ ms, Both Stable

From these examples we can observe that sampling period has a major role to play in deciding the stability of a software based controller.