Benchmarks for Model Transformations and Conformance Checking

Xiaoqing Jin, Jyotirmoy V. Deshmukh, James Kapinski, Koichi Ueda, and Ken Butts *

Abstract

Suitable open benchmark models help to assess, evaluate, and compare tools and techniques, help the research community understand the nature and complexity of the problems facing industrial practitioners, and also help industry to embrace new techniques. We present two benchmark models from the automotive powertrain control domain. Each model has a unique level of complexity. The models are intended to challenge the research community while maintaining a manageable scale. Also, we give our observations and discuss challenges for the research community.

Category: Industrial **Difficulty:** High

1 Introduction

In an industrial model-based design context, different stages in the design cycle lead to the creation of different models of the same underlying cyberphysical system. In an automotive context, controller development usually follows such an iterative design cycle. Control design models are often enhanced in a process known as *refactoring* that typically introduces implementation details into the model, such as controller sampling or effects of fixed point number representations (based on an original floating point controller design). It is an implied requirement that the refactored model not introduce any undesirable behavior compared to the original model; hence, the ability to test that refactored models faithfully represent behaviors in the original model would be useful. Formal analysis techniques such as verification rarely scale to the complexity and size of the final implemented models. Thus, while refactoring in an industrial context usually involves adding complexity to the model, in the context of formal analysis techniques, refactoring might involve simplifying a complex model.

With this proliferation of models across the design cycle, it is essential to have a formal notion of what it means for models to be conformant. Unfortunately, the very definition of conformance is an open problem. For cyberphysical systems, while researchers have used the notions of bisimulation relations [7] and behavior relations [11], it is unclear whether these notions are effective for checking conformance for the models that we consider. An alternative is to use notions of equivalence formulated in temporal logics [3].

*Powertrain Control, Model-based Development, Toyota Technical Center, USA. Contact: {firstname.lastname}@tema.toyota.com

In Sec. 2, we present a model for the powertrain control problem of regulating the air-fuel (A/F) ratio [5]. The A/F ratio has direct implications on the rate of exhaust gas emissions, driveability and fuel efficiency. Catalytic converters work most efficiently to reduce the amount of undesirable exhaust gases when the A/F ratio is at the stoichiometric value. The controller implemented in this model regulates the A/F ratio to this set value. This model, while much simpler than an industrial model, contains many features typical of an industrial system. The plant model has subsystems with variable transport delays, highly nonlinear and nonpolynomial dynamics, and lookup tables (LUTs) representing a data-driven physical model. The controller model has feedback based on Proportional + Integral (PI) control, and two state estimators that provide feedforward control. The model presented in Sec. 2 restricts the model presented in [10] to the `normal` mode of operation, but enhances the controller for this mode. The enhancement consists of an open-loop estimator for a phenomenon known as *wall wetting* that impacts the actual amount of fuel used during ignition.

In Sec. 3, we perform a series of model transformations to obtain a second, simplified model in which the system is a finite-dimensional continuous-time dynamical system with polynomial dynamics. The purpose of this model is to serve as a benchmark for formal verification techniques and to evaluate conformance with the original model. In Sec. 4, we present a comparison of common control-theoretic properties such as overshoot, undershoot, and settling time for both models when stimulated by identical input signals. This comparison is by no means a formal conformance check, but we hope to motivate development of metrics to measure the efficacy of automatic model transformation and conformance checking techniques. Finally, we summarize some of these research challenges for which the models presented herein could serve as benchmarks.

2 A/F Ratio Control Model

In this section, we present an automotive air-fuel control model; in the sequel, we present a simplified version of the model using a series of local, modular transformations. The model we present is a closed-loop model, i.e., it contains a model of the plant and a model of the controller. The former encapsulates physical processes such as throttle and intake manifold air dynamics, and fuel dynamics. The latter describes a controller that incorporates two open-loop estimators implementing feedforward control and a PI controller implementing feedback control. For brevity, we present all the equations in the appendix.

Throttle and Intake Manifold Dynamics. The throttle and intake manifold models are taken from the Simulink Demo palette [2], which is based on the work by Crossley and Cook [6]. For simplicity, we exclude the exhaust gas recirculation system. We assume that the throttle angle θ (in degrees) and the engine speed n in rpm (or $\omega = n\frac{\pi}{30}$ in rad/sec) are exogenous inputs to the model.

The rate at which the throttle plate introduces air into the intake manifold (denoted $\dot{m}_{af}$) is a function that is a product of a function encoding geometric features of the throttle (shown in Eq. 1) and a physical phenomenon involving the atmospheric and intake manifold pressure p (See Eq. 2). The rate at which air exits the intake manifold into the cylinder (denoted $\dot{m}_c$) is specified by the *pumping polynomial* (shown in Eq. 3), a function of p and the engine speed ω. According to the ideal gas law, the rate of change of the manifold pressure is proportional to the difference of the rates at which the air enters the manifold ($\dot{m}_{af}$) and exits the manifold ($\dot{m}_c$). This gives us the ODE in Eq. 4.

Cylinder and Exhaust. The air-fuel path aspects, including the cylinder and exhaust dynamics, are largely based on the development described in [9]. There is a variable delay $\Delta(m_c, n)$ in transporting the exhaust gas produced by the engine to the oxygen (O_2) sensor. This delay depends on the air mass into the cylinder m_c and the engine speed in rpm n. The delayed A/F ratio passes through two first-order transfer functions, representing the exhaust gas transport dynamics and the O_2 sensor dynamics. The output of the transfer functions is the measured A/F ratio, denoted $\lambda_m(t)$. In Eq. 5, we give the ODE govering $\lambda_m(t)$.

Wall Wetting. This subsystem models dynamics of fuel injection, wall film deposition, and evaporation in the manifold. Based on the dynamics described in [9], we obtain the dynamic relation between the rate at which fuel is injected into the manifold ($\dot{m}_\psi$) with the actual rate at which it is aspirated into the cylinder ($\dot{m}_\varphi$) (Eqs. 6-7).

Controller. The controller contains three parts, the first two are feedforward components that estimate the manifold pressure (p) and the amount of liquid fuel that has collected on the manifold wall (m_f). The third is a Proportional + Integral (PI) feedback controller. The objective of the controller is to determine the rate of fuel that should be injected into the manifold, also known as the fuel command (F_c).

The two feedforward estimators are open-loop. The estimator for the manifold pressure is based on a measurement of $\dot{m}_{af}$ or the inlet air mass flow rate. In a real system, such an estimator is a carefully designed system that compensates for phenomena such as sensor noise; for example an extended Kalman filter. For simplicity, we choose an "almost perfect" observer, i.e., we assume almost perfect knowledge of the pumping polynomial (modulo some multiplicative error factor) to observe state p (intake manifold pressure), and use the observer state (denoted p_e) to compute the estimated air mass flowing into the cylinder. The air mass flow estimator dynamics are given by Eqs. 8 and 9.

The feedforward estimator for m_f is based on the Aquino model for the wall wetting dynamics [4]. The estimator uses a lookup table to characterize the parameters in the Aquino model. The dynamics of the estimator are given by Eqs. 10 and 11.

The PI controller dynamics are given by Eq. 12 and the output of the controller is given by Eq. 13. The controller is compliant with a standard published by the MathWorks Automotive Advisory Board (MAAB), which is used by the automotive industry. We use Version 3.0 of the MAAB standard for the model [1].

Error Factor Correction. A constant error factor, c_{24}, is included in the oxygen sensor measurement. Also, the inlet air mass flow rate $\dot{m}_{af}$ is measured by the controller; a constant error factor, c_{23}, is included in this measurement. The fuel command produced by the controller (F_c) and the actual fuel produced by the actuator ($\dot{m}_\psi$) may be different due to actuator error; a multiplicative fuel injector actuator error c_{25} is included to account for this (i.e., $\dot{m}_\psi = c_{25}F_c$).

3 Model Transformations

In this section, we present a simplification of the system dynamics described in Sec. 2. We present the simplification as a set of transformations that can be applied in some order.

Subsystem Deletion. This is the simplest of transformations, but requires designer insight. Some subsystems model physical phenomena that have an effect on the system dynamics but can be ignored for the purposes of designing a basic control law for the system. For instance, we remove the wall wetting subsystem, which has the effect of producing an imprecise but useful representation of the dynamics associated with the control loop. Neglecting the wall wetting dynamics will surely cause the simplified model to behave somewhat differently as compared to the original model; the designer's intuition dictates whether or not the difference is significant. Also, as an effect of removing the wall wetting subsytem, we also remove the feedforward observer in the controller compensating for the fuel dynamics.

First-order Approximation. In this transformation, we replace a higher-order filter with a first-order filter; the time constant of the first-order filter is selected to be comparable to the dominant time constant of the original filter. This is a standard model order reduction technique. In our application, we replace the sensor dynamics and the transport dynamics by a single first order filter.

Finitization. In this transformation, we replace infinite dimensional components such as variable transport delays with finite, first-order approximations. In our application, we have a variable transport delay representing the time it takes for the exhaust gases to reach the A/F sensor. We remove this delay and approximate its effect by increasing the time constant of the first-order filter described above.

After repeated applications of the above transformations, we obtain the ODE in Eq. 15 for the A/F ratio.

Polynomialization. This set of transformations involves using polynomial approximations of nonpolynomial expressions in the model. We observe that this process can be applied in a *nested* fashion, i.e., each nonpolynomial subexpression in a model could be replaced with a polynomial, or it could be applied in a *global* fashion, i.e., an entire expression representing the RHS of an ODE (which could contain both polynomial and nonpolynomial subexpressions) could be replaced by a polynomial. While the global method of polynomialization may be preferable, as it can provide a Taylor series approximation of the vector field, it is often more practical to employ the nested method, since this method is more amenable to industrial models, where the approximation can be applied directly to individual subsystems rather than an analytic representation of the vector field. We also note that polynomialization fixes a maximum degree k for the polynomial, and is usually over a certain range of values for the underlying variables; we call this the domain. Finally, we need a metric to measure the error between a nonpolynomial expression and its polynomial replacement. We typically use nonlinear least-squares fitting. In our application, we apply the polynomialization transformation as follows:

1. We aproximate the square root function on the RHS of Eq. 4 with a second degree polynomial function (shown in Eq. 14), which is accurate in the domain

$0.5 \leq p \leq 1.0$.

2. We replace the rational function on the RHS of Eq. 15 with a degree 2 polynomial function, which is accurate in the domain $1.0 \leq \dot{m}_c \leq 20.0$ and $0.5 \leq F_c \leq 1.2$.

Discrete-to-Continuous Transformation. Here, we replace a discrete-time subsystem with a continuous-time version. This requires that we ignore sampling effects (such as in the controller) and obtain the smoothed dynamics. We obtain continuous-time formulations for the PI controller, and the cylinder mass airflow estimator.

Simplified Model. Applying the above transformations, we obtain the polynomial system representation shown in Eqs. 16-20.

4 Conformance Analysis

Note that the models presented in Sections 2 and 3 have 2 inputs: the engine speed n (in rpm) and the throttle angle θ (in degrees). We consider an experimental setup where n is fixed; this reflects a test scenario for real engines where the speed is held constant by a dynamometer. We allow the throttle angle to be a pulse train signal of duration 30 seconds, with a period of 10 seconds, an initial delay of 3 seconds, a pulse-width of 5 seconds, a minimum amplitude of $8.8°$, and a maximum amplitude in $[8.8°, 90°]$. Engine speeds in $[1000, 2000]$ rpm correspond to the operating range for which the model transformation is valid. In order to compare the two models, we pick three engine speeds within this range, and perform 100 simulations, such that for each simulation n is fixed, and the pulse amplitude is selected randomly in a Monte Carlo fashion.

We define the normalized A/F ratio μ as $\frac{(\lambda - 14.7)}{14.7}$, where λ is the A/F ratio. As regulating μ to 1 is the control objective, we compare the models on the basis of this signal. We use control-theoretic properties of μ, such as the maximum overshoot, minimum undershoot, and settling time as criteria for comparison. We define a settling region of $\pm 1\%$ of the reference value for μ (which is 1.0) for the cases where the engine speed is $[1000, 1500]$ rpm. For higher speeds we use a settling region of $\pm 2\%$. We also measure the RMS error between the signal μ_c for the complex model, and the signal μ_s for the simplified model.

Table 1 presents the results from the model comparison tests. Observe that, for low engine speeds, the two models exhibit comparable behavior for all three criteria, and the RMS error between the two μ signals is low. At the higher speeds, the model fidelity considerably degrades. This can be partly ascribed to the RHS of the ODE in Eq. 17 being a polynomialization of the RHS of the ODE in Eq. 15. This approximation is valid for a certain range of $\dot{m}_c$ and F_c values, both of which are functions of ω. Further, the polynomialization is performed in a nested fashion. In future work, it would be interesting to refactor the simplified model so as to extend the valid ranges of operation and polynomialize the global expressions.

We now briefly discuss the key challenge problems for conformance checking.

Conformance Metrics. Identifying effective metrics is difficult. It is unclear whether a single metric over the space of signals could serve as *the* definition for behavioral conformance. Our experience suggests that the notion of conformance should be application-specific.

Table 1: Comparison between a complex control model (Sec. 2) and its transformation to a polynomial dynamical system (16).

Engine	Overshoot			Undershoot			Settling Time			RMS
Speed	M1	M2	%Δ	M1	M2	%Δ	M1	M2	%Δ	error
1000	0.12	0.12	0.2	-0.13	-0.13	3.5	0.00	0.00	0.0	1.60
1500	0.16	0.21	31.2	-0.15	-0.23	58.4	0.86	0.77	10.4	2.32
2000	0.89	0.37	58.4	-0.98	-0.42	56.5	1.29	0.74	42.6	7.28

Systematic transformations. Transformations preserving transient and stability characteristics would be valuable. One direction is to evaluate nonlinear model order reduction techniques [8] and the metrics used therein.

Abstractions for Verification. Model transformations that generate abstract models that conservatively respect properties such as the reachable state space, and more general temporal properties are valuable for verification efforts.

References

[1] Mathworks automotive advisory board, control algorithm modeling guidelines using matlab, simulink, and stateflow. `http://www.mathworks.com/automotive/standards/docs/MAAB_Style_Guideline_Version3p00_pdf.zip`, 2012.

[2] *Simulink User's Guide*. Natick, MA, 2012.

[3] H. Abbas, B. Hoxha, G. Fainekos, J. V. Deshmukh, J. Kapinski, and K. Ueda. Conformance testing as falsification for cyber-physical systems. *CoRR arXiv:1401.5200 [cs]*, 2014.

[4] C. Aquino. Transient a/f control characteristics of a 5 liter central fuel injection engine. In *SAE Paper 810494*, 1981.

[5] J. A. Cook, J. Sun, J. H. Buckl, I. V. Kolmanovsky, H. Peng, and J. W. Grizzle. Automotive powertrain control - a survey. *Asian J. of Control*, 8:237–260, 2006.

[6] P. R. Crossley and J. A. Cook. A nonlinear engine model for drivetrain system development. In *Proc. of Int. Conf. on Control*, volume 2, pages 921–925, 1991.

[7] A. Girard and G. J. Pappas. Approximate bisimulation: A bridge between computer science and control theory. *Eur. J. Control*, 17(5-6):568–578, 2011.

[8] C. Gu. *Model Order Reduction of Nonlinear Dynamical Systems*. PhD thesis, University of California, 2011.

[9] L. Guzzella and C. Onder. *Introduction to Modeling and Control of Internal Combustion Engine Systems*. Springer-Verlag, Berlin Heidelberg, 2nd edition edition, 2010.

[10] X. Jin, J. V. Deshmukh, J. Kapinski, K. Ueda, and K. Butts. Powertrain control verification benchmark. In *Hybrid Systems: Computation and Control (HSCC) (*To appear*)*. ACM, 2014.

[11] A. Rajhans and B. H. Krogh. Heterogeneous verification of cyber-physical systems using behavior relations. In *Hybrid Systems: Computation and Control (HSCC)*. ACM, 2012.

Appendix

A: Equations for Original Model

In this section, we outline the equations for the model described in Sec. 2. We use the notation $\dot{x}$ for a variable representing flow rate that is an algebraic function of state variables and inputs; we use dx/dt to represent the derivative of a state variable, which is expressed as a function of x and other state variables and inputs.

The function encoding the geometry of the throttle is given as a function of the throttle angle θ:

$$\hat{\theta} \quad = \quad c_6 + c_7\theta + c_8\theta^2 + c_9\theta^3. \tag{1}$$

The inlet air mass flow rate $\dot{m}_{af}$ is then given by the product of the above function, and a function encoding a physical phenomenon relating the atmospheric pressure (c_{10}) to the intake manifold pressure p:

$$\dot{m}_{af} \quad = \quad 2\hat{\theta}\sqrt{\frac{p}{c_{10}} - \left(\frac{p}{c_{10}}\right)^2}. \tag{2}$$

The pumping polynomial is a function of the engine speed ω (in rad/sec) and the intake manifold pressure p:

$$\dot{m}_c \quad = \quad c_{12}\left(c_2 + c_3\omega p + c_4\omega p^2 + c_5\omega^2 p\right). \tag{3}$$

Finally, the ODE for the intake manifold pressure is described as follows:

$$\frac{dp}{dt} = c_1\left(2\hat{\theta}\sqrt{\frac{p}{c_{10}} - \left(\frac{p}{c_{10}}\right)^2} - c_{12}\left(c_2 + c_3\omega p + c_4\omega p^2 + c_5\omega^2 p\right)\right). \tag{4}$$

The ODE governing the measured A/F ratio λ_m is given below. Recall that $\Delta(m_c, n)$ is a variable delay that depends on the air mass entering the cylinder, and the engine speed n in rpm. The lookup table for $\Delta(m_c, n)$ is given in Table 4.

$$\frac{d^2\lambda_m(t)}{dt^2} = \frac{1}{0.002}\left[-0.12\frac{d\lambda_m(t)}{dt} - \lambda_m(t) + \lambda_c(t - \Delta(m_c, n))\right]. \tag{5}$$

The mass of the fuel flowing into the cylinder is given by:

$$\dot{m}_\varphi = (1 - \kappa(\omega, m_c))\dot{m}_\psi + \frac{m_f}{\tau(\omega, m_c)}, \tag{6}$$

where $\dot{m}_\varphi$ is the fuel mass flow rate into the cylinder, $\dot{m}_\psi$ is the fuel mass flow rate into the intake manifold. The dynamic equation for the mass of fuel stored in the fuel film, m_f, is given by:

$$\frac{d}{dt}m_f = \kappa(\omega, m_c)\dot{m}_\psi - \frac{m_f}{\tau(\omega, m_c)}. \tag{7}$$

Equations (6) and (7) are taken from [9] Equations 2.60 and 2.61.

In the model $1 - \kappa(\cdot)$ and $\tau(\cdot)$ are given as 2D lookup tables. These lookup tables are estimated values taken from [9] Figure 2.21, and presented in Table 5. The table axes are air mass entering the cylinder (m_c) and engine speed (n).

The state equation for the manifold pressure estimator is

$$\begin{aligned} p_e[k+1] &= p_e[k] + 0.01 \cdot c_1 \cdot \Big(\hat{\dot{m}}_{af}[k] - \\ &\quad \big(c_2 + c_3\omega[k]p_e[k] + c_4\omega[k]p_e[k]^2 + c_5\omega[k]^2 p_e[k]\big)\Big), \end{aligned} \tag{8}$$

where the $X[k]$ is the signal X sampled at time increment k, and $\hat{\dot{m}}_{af}[k] = c_{23}\dot{m}_{af}[k]$ is the measured value of $\dot{m}_{af}$ modified by an error factor (c_{23}). The output of the air mass estimator component is

$$\hat{\dot{m}}_c[k] = c_2 + c_3\omega[k]p_e[k] + c_4\omega[k]p_e[k]^2 + c_5\omega[k]^2 p_e[k]. \tag{9}$$

The state equation for the wall wetting dynamics estimator is given by

$$f_w[k] = P(\omega[k-1])f_w[k-1] + R(\omega[k-1])F_c[k-1], \tag{10}$$

where f_w is the estimated amount of liquid fuel on the manifold wall, and $F_c[k-1]$ is the previous value of the fuel command. $P(\cdot)$ and $R(\cdot)$ are given as 1D lookup tables in Table 6 based on engine speed (n). The output of the estimator is given by:

$$f_i[k] = \frac{\frac{\hat{\dot{m}}_c[k]}{c_{11}} - (1 - P(\omega[k]))f_w[k]}{1 - R(\omega[k])}. \tag{11}$$

The feedback PI controller update equation is given by

$$i[k+1] = i[k] + c_{24}\lambda[k] - c_{11}. \tag{12}$$

The controller output command is given by

$$F_c[k] = (1 + c_{13}(c_{24}\lambda[k] - c_{11}) + c_{14}i[k])f_i[k]. \tag{13}$$

B: Equations for Simplified Model

The polynomial approximation of the square root expression in (4) is given below:

$$\sqrt{\frac{p}{c_{10}} - \left(\frac{p}{c_{10}}\right)^2} \approx c_{20}p^2 + c_{21}p + c_{22}. \tag{14}$$

After applying various transformations (except polynomialization) we obtain the following:

$$\frac{d\lambda}{dt} = c_{26}\left(\frac{c_{12}\cdot\left(c_2 + c_3\omega p + c_4\omega p^2 + c_5\omega^2 p\right)}{c_{25}\cdot F_c} - \lambda\right). \tag{15}$$

We now give the polynomial ODEs describing the simplified model.

$$\begin{aligned}
\frac{d}{dt}p &= c_1\left(2\hat{\theta}\left(c_{20}p^2 + c_{21}p + c_{22}\right) - c_{12}\left(c_2 + c_3\omega p + c_4\omega p^2 + c_5\omega^2 p\right)\right) &(16)\\
\frac{d}{dt}\lambda &= c_{26}\left(c_{15} + c_{16}c_{25}F_c + c_{17}c_{25}^2F_c^2 + c_{18}\dot{m}_c + c_{19}\dot{m}_c c_{25}F_c - \lambda\right) &(17)\\
\frac{d}{dt}p_e &= c_1\left(2c_{23}\hat{\theta}\left(c_{20}p^2 + c_{21}p + c_{22}\right) - \left(c_2 + c_3\omega p_e + c_4\omega p_e^2 + c_5\omega^2 p_e\right)\right) &(18)\\
\frac{d}{dt}i &= c_{14}\left(c_{24}\lambda - c_{11}\right), &(19)
\end{aligned}$$

where F_c is given by:

$$F_c = \frac{1}{c_{11}}\left(1 + i + c_{13}(c_{24}\lambda - c_{11})\right)\left(c_2 + c_3\omega p_e + c_4\omega p_e^2 + c_5\omega^2 p_e\right), \tag{20}$$

and $\dot{m}_c$ is given by (3).

C: Model Parameters and Lookup Tables

List of model parameters (constants):

Table 2: Model Parameters.

Param	Value	Unit	Description
c_1	0.41328	RT/Vm	
c_2	-0.366		Coefficient for *Pumping* polynomial
c_3	0.08979		Coefficient for *Pumping* polynomial
c_4	-0.0337		Coefficient for *Pumping* polynomial
c_5	0.0001		Coefficient for *Pumping* polynomial
c_6	2.821		Coefficient for $f(\theta)$ polynomial
c_7	-0.05231		Coefficient for $f(\theta)$ polynomial
c_8	0.10299		Coefficient for $f(\theta)$ polynomial
c_9	-0.00063		Coefficient for $f(\theta)$ polynomial
c_{10}	1.0	bar	Atmospheric pressure
c_{11}	$14.7/12.5$		Desired air-fuel ratio (all other modes / `power` mode)
c_{12}	0.9		Manifold pressure estimate error factor
c_{13}	0.05		Proportional gain for PI controller
c_{14}	0.03		Integral gain for PI controller
c_{15}	13.893		Coefficient for A/F polynomial
c_{16}	-35.2518		Coefficient for A/F polynomial
c_{17}	20.7364		Coefficient for A/F polynomial
c_{18}	2.6287		Coefficient for A/F polynomial
c_{19}	-1.592		Coefficient for A/F polynomial
c_{20}	-2.3421		Coefficient for square root polynomial
c_{21}	2.7799		Coefficient for square root polynomial
c_{22}	-0.3273		Coefficient for square root polynomial
c_{23}	1.0		MAF sensor constant error factor
c_{24}	1.0		Oxygen sensor constant error factor
c_{25}	1.0		Fuel injector actuator error factor
c_{26}	4.0		First-order transfer function constant
u_1		degrees	Throttle angle
u_2		rad/sec	Engine speed

Table 3: Intermediate variables.

State	Unit	Description
$\dot{m}$	g/s	Air mass flow
$\dot{m}_c$	g/s	Air flow to cylinder
$\hat{u}_1$		Output of throttle angle polynomial
F_c	g/s	Commanded fuel

Table 4: Delay LUT.

n	m_{chrg}	Delay
800	0.05	0.25
1000	0.05	0.20
1500	0.05	0.20
2000	0.05	0.20
3000	0.05	0.20
800	0.15	0.30
1000	0.15	0.25
1500	0.15	0.20
2000	0.15	0.20
3000	0.15	0.20
800	0.20	0.40
1000	0.20	0.30
1500	0.20	0.20
2000	0.20	0.20
3000	0.20	0.20
800	0.25	0.80
1000	0.25	0.60
1500	0.25	0.40
2000	0.25	0.30
3000	0.25	0.20

Table 5: LUTs for $1 - \kappa(\cdot)$ and $\tau(\cdot)$.

n	m_c	$1 - \kappa(\cdot)$	$\tau(\cdot)$
1000	0.1	0.80	0.40
1500	0.1	0.70	0.30
2000	0.1	0.70	0.35
2500	0.1	0.80	0.30
3000	0.1	0.90	0.20
1000	0.2	0.70	0.22
1500	0.2	0.66	0.22
2000	0.2	0.65	0.40
2500	0.2	0.73	0.35
3000	0.2	0.85	0.50
1000	0.3	0.66	0.20
1500	0.3	0.66	0.22
2000	0.3	0.63	0.50
2500	0.3	0.66	0.40
3000	0.3	0.80	0.35
1000	0.4	0.60	0.35
1500	0.4	0.60	0.30
2000	0.4	0.60	0.45
2500	0.4	0.60	0.50
3000	0.4	0.70	0.40

Table 6: LUTs for $P(\cdot)$ and $R(\cdot)$.

n	$P(\cdot)$	$R(\cdot)$
1000	0.4	0.5
2000	0.65	0.95
3000	0.5	0.9