Model Predictive Control of a Heavy-Duty Truck Based on Gaussian Process

Fernando Henrique Morais da Rocha, Valdir Grassi Jr
Sao Carlos School of Engineering (EESC)
University of Sao Paulo (USP)
Sao Carlos, SP, Brazil
E-mail: fernandorocha@usp.br, vgrassi@usp.br

Vitor Campanholo Guizilini, Fabio Ramos
School of Information Technologies
University of Sydney
Sydney, NSW, Australia
Email: vitor.guizilini@gmail.com, fabio.ramos@sydney.edu.au

***Abstract*—A recent trend on the automotive market is the incorporation of several driver assistance systems into common vehicles. Among the most studied systems, lies the cruise control (speed), either for platooning or to ease off driver's tasks. For the design of this controller, particularly when applied to large and heavy vehicles, a longitudinal model that represents satisfactorily all the features of this complex system is necessary. In this sense, a Gaussian Process model was applied within Model Predictive Control to regulate the speed of a heavy-duty truck. The controller takes the variance produced by the Gaussian Process model into account on the optimization of the control signal. The proposed controller achieved low tracking error even on hard conditions, like steep roads.**

I. Introduction

Recently, the automotive industry is pushing the development of several concepts which can make traffic more efficient [1]. In this sense, the control of vehicles has been studied for several years in many different fields, like automated highway systems [2], vehicle stability control [3], autonomous vehicles and several kinds of driver assistance systems. Automatic cruise controls, active rollover protection and parking assistance are some systems that are available on the market.

Heavy-duty vehicles are the most probable candidate to be completely automated [4]–[6]. It's been shown that longitudinal control for heavy duty vehicles is quite different from passengers cars [7], [8]. The main differences are related to vehicle dynamics, affecting modeling and control design directly. In fact, the longitudinal and platooning control of heavy duty vehicles are harder to achieve due to the following characteristics:

- Low power to mass ratio: the control input saturates easily and the vehicle has limited acceleration capability.
- Mass dominant: the mass of the vehicles are big, greatly affecting the dynamics.
- Large Actuator delay: appearing on both throttle and braking systems, and also on automatic gearboxes.

Model Predictive Control [9], [10] is a class of computer control algorithms that predicts future responses of a plant based on its system model. Control actions are obtained by repeatedly minimizing a cost function over a finite horizon, in a receding horizon strategy.

The popularity of MPC is to a great extent owed to the ability of MPC algorithms to deal with constraints that are frequently met in control practice and are often not well addressed with other approaches. MPC algorithms can handle hard state and rate constraints on inputs and states [11].

However, for this very complex system, the traditional modeling is very complicated, so empirical system identification techniques were employed, where the model is built (or learned) based on data collected during vehicle operation.

In this work, a Model Predictive Controller is employed to tackle the longitudinal control of a heavy duty road truck. The system model is obtained using Gaussian Processes, a data-driven probabilistic modeling technique that has the advantage of providing an indication of the model quality by the variance.

II. Dynamic System Modeling with Gaussian Process

Consider a set of N vectors of dimension D containing noisy input data, $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N]$, and a vector containing observed output data from the system to be modeled, $\mathbf{y} = [y(1), y(2), \cdots, y(N)]^T$. The goal of the modeling process is to build a model (a function $f(\cdot)$) from $\mathbf{X}$ and $\mathbf{y}$, and then find the output $y(N+1)$ when the system is presented with a new input vector $\mathbf{x}^*$.

A Gaussian Process is an example of probabilistic, non-parametric modeling technique with uncertainty prediction. A Gaussian Process is defined as a collection of random variables that have a joint multivariate Gaussian distribution with the form: $f(x^1), \cdots, f(x^n) \sim \mathcal{N}(\mu(f(x)), \Sigma)$, where Σ_{pq} represents the covariance between the points x^p and x^q. The mean function $\mu(f(x))$ (which can be stationary and is usually assumed as $\mu(f(x)) = 0$) and the covariance function $\Sigma_{pq} = Cov(x^p, x^q)$ completely define a Gaussian process. Assuming a relation between the inputs x and the outputs y as $y = f(x)$, we have $Cov(y^p, y^q) = Cov(x^p, x^q)$, where $C(\cdot,\cdot)$ is any function that generates a positive definite covariance matrix. The most common choice for covariance function is the *Squared exponential*, seen in

(1):

$$k_{SE}(r) = v_1 \exp\left[-\frac{1}{2}\sum_{d=1}^{D} w_d (x_d^p - x_d^q)^2\right] + v_0 \quad (1)$$

where $v_0, v_1, w_d, d = 1, \cdots, D$ are the hyperparameters of the covariance function and D is the input dimension. Other common covariance functions are: Matérn, Exponential, Rational-Quadratic, among others [12]. Given a training set, the hyperparameters $\Theta = [w_1 \cdots w_D \; v_0 \; v_1]^T$ have to be identified (learned). The model is determined by $f(\cdot)$, $\mathbf{X}$ and $\mathbf{y}$, and not by the parameters of a fixed model structure, and this is what makes it a probabilistic and non-parametric approach [13]. In this sense, the probability of the hypothesis $f(\mathbf{x}^*)$ according to a dataset $\mathbf{X}$ and $\mathbf{y}$ can be seen on (2):

$$p(f(\mathbf{x}^*|\mathbf{X}, \mathbf{y}) = \frac{p(\mathbf{y}|f(\mathbf{x}^*, \mathbf{X}))p(f(\mathbf{x}^*))}{P(\mathbf{y}|\mathbf{X})} \quad (2)$$

The term $p(\mathbf{y}|f(\mathbf{x}^*, \mathbf{X}))$ is the conditional likelihood of the model and, as it is a probability distribution, represents represents the model output as a mean and variance, $p(f(\mathbf{x}^*))$ represents the prior knowledge from the model. Based on the covariance function, a set of hyperparameters is determined using the training set $\mathbf{X}, \mathbf{y}$, and then the posterior value can be calculated.

The hyperparameters are obtained through the maximization of the likelihood $p(f(\mathbf{x}^*|\mathbf{X}, \mathbf{y})$. Since an analytical solution is very difficult to calculate, approximations or other techniques are necessary, such as the one described below.

The calculation of the model output for a given covariance function [13] is straightforward. On (2), it can be seen that the posterior probability depends on the hyperparameters through the likelihood $p(\mathbf{y}|f(\mathbf{x}^*, \mathbf{X})$. Applying the log on this probability, which now can be calculated analytically, we obtain:

$$L(\mathbf{\Theta}) = -\frac{1}{2}\log(|\mathbf{K}|) - \frac{1}{2}\mathbf{y}^T\mathbf{K}^{-1}\mathbf{y} - \frac{N}{2}\log(2\pi) \quad (3)$$

where $\mathbf{y}$ is the $N \times 1$ vector containing the training outputs, $\mathbf{K}$ is the $N \times N$ covariance matrix of the training inputs. The hyperparameters are obtained through the maximization of (3), and this is known as *Maximum Likelihood Method*. Any optimization method can be used, however, we note that due to a matrix inversion at every iteration, depending on the size of the dataset, this method can be computationally demanding [13].

As can be seen on (1), there is one hyperparameter for each input dimension, so, through the learning process, it is possible to identify the most relevant input dimensions or the ones that does not contribute to the output by analysing the magnitude of the hyperparameters. If it is zero, or near zero, it means that the corresponding dimension has little impact and can be removed. This effect is called Automatic Relevance Determination, idea developed by MacKay and Neal [14].

The described method is easily applied to regression problems. From the training set $\mathbf{X}$, the covariance matrix $\mathbf{K}_N$ of order $N \times N$ is determined. Then, for a new input vector $\mathbf{x}^*$, a new covariance matrix $\mathbf{K}_{N+1}$ of order $N + 1 \times N + 1$ is obtained by:

$$\mathbf{K}_{N+1} = \begin{bmatrix} [\mathbf{K}_N] & [\mathbf{k}(\mathbf{x}^*)] \\ [\mathbf{k}(\mathbf{x}^*)^T] & [k(\mathbf{x}^*)] \end{bmatrix} \quad (4)$$

where $\mathbf{k}(\mathbf{x}^*)$ is the $N \times 1$ covariance vector between the new input and the training data, and $k(\mathbf{x}^*)$ is the variance of the new input data. For the new input data, the corresponding output distribution is given by $\hat{y}(N+1)|\mathbf{x}^* \sim \mathcal{N}(\mu(\mathbf{x}^*), \sigma^2(\mathbf{x}^*))$, where:

$$\mu(\mathbf{x}^*) = \mathbf{k}(\mathbf{x}^*)^T\mathbf{K}^{-1}\mathbf{y} \quad (5)$$

$$\sigma^2(\mathbf{x}^*) = k(\mathbf{x}^*) - \mathbf{k}(\mathbf{x}^*)^T\mathbf{K}^{-1}\mathbf{k}(\mathbf{x}^*) + v_0 \quad (6)$$

A more detailed explanation for the application of Gaussian Processes on regression problems can be seen in [12] and [15].

The described approach can model static nonlinearities, and can also be used for modeling of dynamical systems using delayed inputs and outputs as regressors, as a NARX (Nonlinear AutoRegressive model with eXternal input), as can be seen on Fig. 1, where the GP input vector $\mathbf{x} = [u(k), u(k-1), \ldots, u(k-n), y(k), y(k-1), \ldots, y(k-n)]$ is used to predict the output of the system in the next time step.

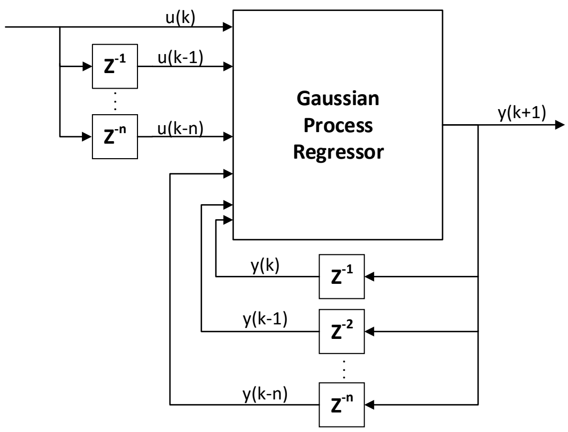

Figure 1. Gaussian Process NARX-like model [16].

Another important feature of the Gaussian Process models is the fact that not only the output is predicted from new input data, but also the confidence about this prediction, which can be used to identify regions of the input space where the prediction is not good enough, due to lack of training data in that region, or the complexity of the system.

III. Model Predictive Controller

The term Model Predictive Control is used to make reference to a wide range of controllers which make explicit use of a model of the process to obtain the control signal for the system by minimizing an objective function [9]. The main ideas of this type of controller are:

- Explicit use of a model to predict the system output at a future horizon;
- Calculation of a sequence of control signals by optimizing a cost function that measures the system performance;
- Receding horizon strategy, in which the horizon is shifted towards to the future at each time instant, and the first item from the calculated control signal is applied.

The appeal of this idea from a control engineering perspective is obvious because it provides a systematic approach to the design of strategies that achieve optimal performance. MPC is arguably the most widely accepted modern control strategy because it offers, through its receding horizon implementation, an eminently sensible compromise between optimality and speed of computation [10]. A block diagram of the control strategy can be seen in Fig. 2.

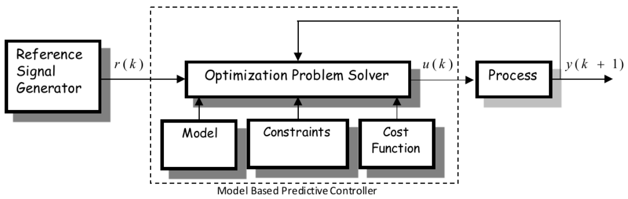

Figure 2. Model-based Predictive Controller diagram [17].

The benefits of optimal control are, however, difficult to achieve in the case of systems with nonlinear models and systems that are subject to constraints on input variables or model states, which is the case for most real systems. For both these cases, in general, it is not possible to derive analytic expressions for the optimal control solution, and iterative optimization (numeric) methods are necessary.

In order to solve the MPC control problem, it is necessary to define some elements:

Model: The model of the system. Practically every possible form of modeling a process can be used for MPC, like Impulse and Step response, transfer functions, state spaces, nonlinear models and various other approaches, like Neural-Networks, Fuzzy logic, Gaussian Process [18] [19] [20], [21].

Cost Function: Many different cost function can be used. The general aim is that the future output y on the considered horizon should follow a determined reference signal (r) and, at the same time, the control effort (Δu) necessary for doing so should be penalized [9]. The most used expression for such cost function is:

$$J(P,C,r,y) = \sum_{j=1}^{P} \delta(j)[r(k+j)-y(k+j)]^2 + \sum_{j=1}^{C} \lambda(j)[\Delta u]^2 \quad (7)$$

where $y(k+j)$ is the model predicted output for every step in the prediction horizon P ($j = 1, \dots, P$), $r(k+j)$ is the reference trajectory and C is the control horizon, which defines that after an interval $C < P$ the proposed control signal is kept constant ($u(k+j) = u(k+C)$, for any $C \leq j \leq P$). The parameters $\delta(j)$ and $\lambda(j)$ are sequences that consider the future behavior and can be tuned to obtain, for example, smoother control with less effort or tighter control.

Constraints: In practice all processes are subject to constraints. The actuators have a limited field of action and a determined slew rate, and this limitations has to be respected for a safe and efficient operation of the system.

In order to obtain the control signal sequence $u(k+j)$ that leads the system to the desired set-point, it is necessary to minimize the functional J in (7), such that the prediction error between $r(k+j)$ and $y(k+j)$ is a minimum. As it is a receding horizon strategy, only the first element of this vector $u(k+1)$ is used, rejecting the rest and repeating the calculations at the next sampling time.

In this paper, the process model to be used on the MPC is a Gaussian Process, so one issue that can appear for applied MPC is the efficiency of a numerical solution. Nonlinear programming optimization algorithm is very demanding for computation. Various approximations and other approaches (e.g. approximation of explicit solution [22]) exist to decrease computational load. Some works that further investigate the utilization of GP models for control can be found in [11], [22]–[25].

IV. Gaussian Process based MPC applied to a Heavy-Duty Truck

The vehicle to be modeled is a 7 meters long Heavy-Duty road truck with 9 ton weight. The drivetrain has a 12.7 l motor, with maximum power of 360 hp, maximum torque of 1850 Nm and a 14-gear automatic gearbox. The brake system has compression release engine brake and pneumatic drum brakes. For this system, traditional modeling techniques are not capable of properly reproducing the dynamics of the vehicle, hence, the system was modeled with a Gaussian process model.

A. Longitudinal GP Model of the Truck

The datasets for training were collected from the real vehicle by driving it on the *campus* of University of São Paulo, São Carlos. The vehicle data were obtained from its CAN (Controller Area Network), accessed through a ROS node [26]. The collected data contained the vehicle speed, throttle and brake commands, motor RPM and gear. To measure the road slope, a MTi-100 IMU, from Xsens, was used. The sampling frequency for all measurements is 5 Hz. The following datasets were collected:

- DS1 - Regular driving by a human - 4500 samples;
- DS2 - Step input applied through the ROS node - 160 samples.

The model was trained to predict the truck speed at instant $k+1$, based on the speed of the previous instant, the previous control input u, and road slope α, as $y(k+1) = f(\mathbf{x})$ with $\mathbf{x} = [v(k), u(k), \alpha(k)]$. The covariance used was a sum of the linear, squared exponential and Matérn

1/2. All the Gaussian Process trainings and predictions on this work were performed using the CVPP[1] toolbox.

The output of the model trained with dataset DS1 can be seen on Fig. 3. The model follows the system real behavior, with low variance, due to the good amount of data samples.

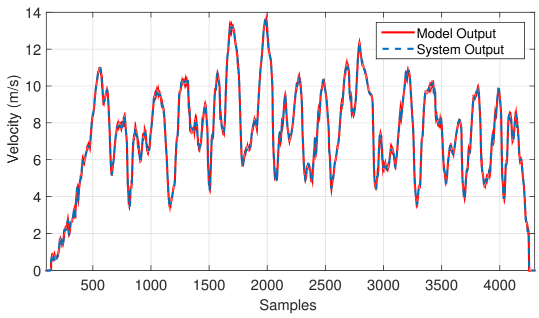

Figure 3. Estimated speed for DS1.

For the dataset DS2, the speed and control signal can be seen on Fig. 4, where the actuator delay is clearly visible.

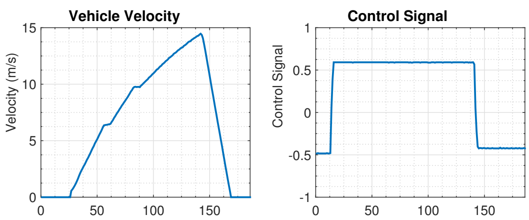

Figure 4. Speed and control input for training set DS2.

It is possible to note on Fig. 5, that even with a small training set, the model is accurate, as the system step response is smooth.

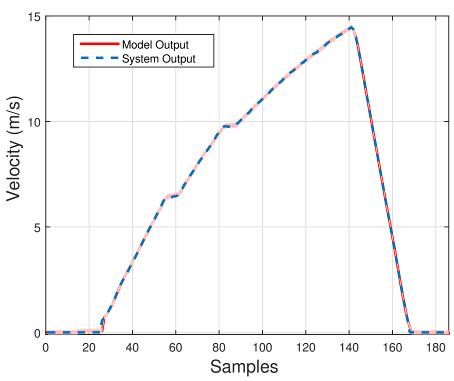

Figure 5. Estimated speed for DS2.

In order to verify the generalization capabilities of the model, the data from DS1 was applied to the model trained with DS2, and even with few data samples and a small part of the input space excited on DS2, it is able to reproduce the DS1 behavior with good accuracy.

[1]https://bitbucket.org/vguizilini/cvpp

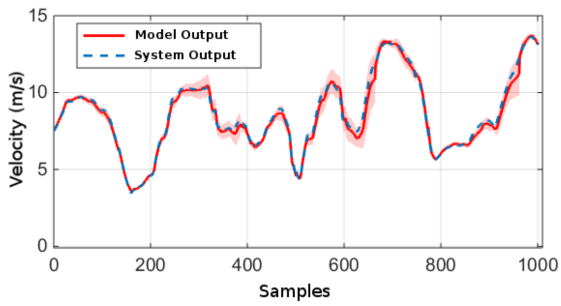

Figure 6. DS1 data applied to model trained with DS2.

To assess the model performance, the *n-fold cross-validation* technique was used with 10 folds ($n = 0$). The performance was measured using the *Root Mean Square Error* (RMSE) and the more probabilistic-suited method *Negative Log Probability Density* (NLPD) [12], [16]. The results are listed on Table I. It can be noted that both models describe the system behavior with small errors, and the model trained with the bigger dataset shows more confidence about the predictions, which can be noted by the NLPD measurement.

Table I
PERFORMANCE OF THE TRAINED MODELS.

	RMSE	NLPD
DS1	0.004	-2.17
DS2	0.011	-1.32
DS1-DS2	0.012	-1.14

B. Predictive Control of the Truck Velocity

The model obtained through the dataset DS2 was used to perform the controller architecture depicted in Fig. 2. The control inputs $u(k+j)$ are obtained by minimizing the cost function (7), with $\delta(j) = 1$ and $\lambda(j) = 0.3$ for all j, and under the constraints:

$$\begin{aligned} 0 \leq y(k+j) &\leq 20 \\ -0.5 \leq u(k+j) &\leq 0.6 \\ \sigma(k+j) &\leq 0.2 \end{aligned} \tag{8}$$

Besides the constraints due to system limitations, a hard constraint on variance value $\sigma \leq 0.2$ has been defined in order to avoid unpleasant responses that are consequence of model uncertainties.

It was proved in [27] that a suitably long horizon is adequate to ensure stability on a receding horizon scheme, even when dynamics and/or cost change in real-time. In this work, it was used a small horizon of 2 time-steps to test our controller in hard situations.

To verify the performance of the controller, a simulation of the system was conducted using the model trained with the dataset DS1 to simulate the truck response. The reference signal was created artificially such that there are two sudden change on the set-point: at $k = 4$ seconds, it goes from zero to 10 m/s, and at $k = 40$ seconds, it goes from 10 to 5 m/s. On the first test, the road was considered flat. The response of the controller for this first test, as well as the input signal sequence applied can be seen on Fig. 7. After the settling time, the controller was able to follow the reference with less than 2.5% error.

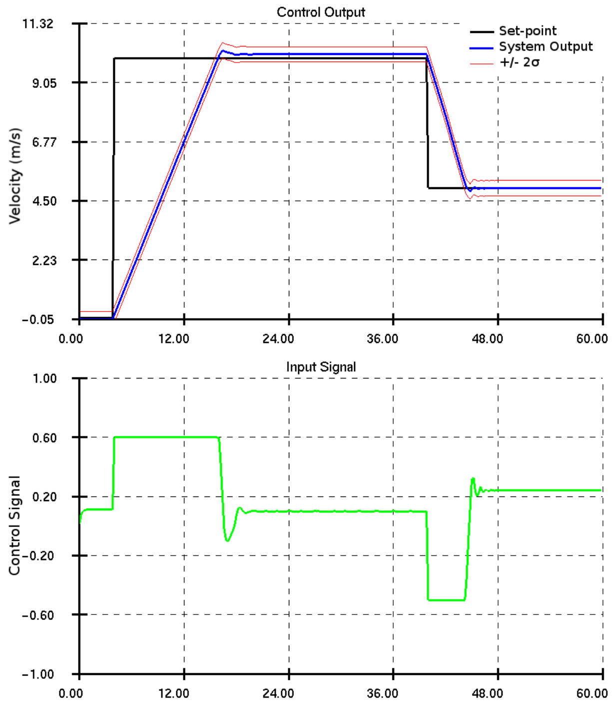

Figure 7. System output and control signal for a flat road.

Two other tests were executed, with the slope of the road set to a steep condition (10%). On Fig. 8 and 8 it is shown the system output running uphill and downhill respectively. Even on this hard conditions, the control is able to track the reference. On the uphill case, it takes more time to gain speed, due to the big mass of the vehicle and saturation of the control signal. Also due to the vehicle weight, on the downhill case, after the vehicle catch up the reference, it spends most of the time braking to maintain the speed.

V. Conclusions

In this work, a Model Predictive Controller with Gaussian Process model is applied to undertake the longitudinal control of a long-haul road truck. A Gaussian Process Model of the truck longitudinal dynamics were learned from real data. The obtained model achieved high accuracy also provided information about its confidence on the predictions, which can be used to achieve a more robust control or even guide collection of new data to improve the model. Simulations of the closed-loop system presented a very good performance of the MPC controller, even at extreme conditions, like small prediction horizon and very steep roads, showing that Gaussian Process models offer an attractive possibility for control design. Some modifications can be applied to the principle presented in this work to better suit it for practical application, like accelerating computations and online adjusting the model.

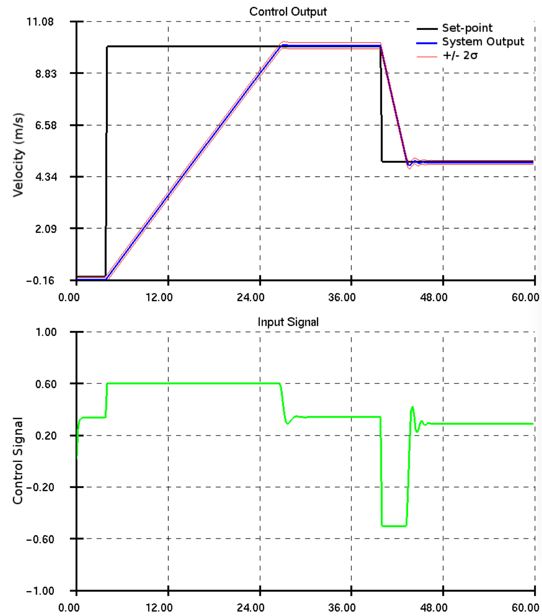

Figure 8. System output and control signal for an uphill run.

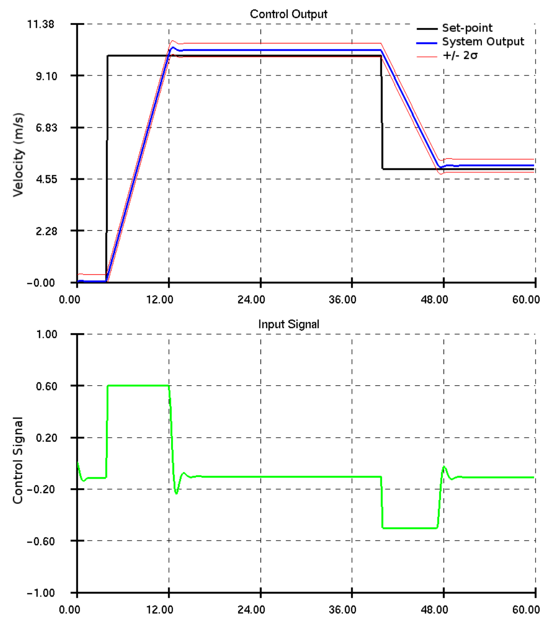

Figure 9. System output and control signal for a downhill.

The extensions, and the application of the technique on a real truck, are planned as future work.

Acknowledgment

The authors thank CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico and FAPESP - Fundação de Amparo à Pesquisa do Estado de São Paulo for the financial support. This research was supported by funding from the Faculty of Engineering and Information Technologies, The University of Sydney, under the Faculty Research Cluster Program.

References

[1] X.-Y. Lu and J. Hedrick, "Longitudinal control design and experiment for heavy-duty trucks," in *American Control*

Conference, 2003. Proceedings of the 2003, vol. 1, June 2003, pp. 36–41.

[2] P. Ioannou, *Automated Highway Systems*. Plenum, 1997.

[3] R. G. Hebden, C. Edwards, and S. K. Spurgeon, "Automotive steering control in a split- manoeuvre using an observer-based sliding mode controller," *Vehicle System Dynamics*, vol. 41, no. 3, pp. 181–202, 2004.

[4] U. S. Department of Transportation, "Intelligent vehicle initiative needs assessment," Federal Transit Administration, Tech. Rep. FTA-TRI-11-99-33 DOT-VNTSC-FTA, November 1999.

[5] R. Bishop, "Intelligent vehicle applications worldwide," *Intelligent Systems*, vol. 15, no. 1, pp. 78–83, 2000.

[6] R. Shanker, A. Jonas, S. Devitt, K. Huberty, S. Flannery, W. Greene, B. Swinburne, G. Locraft, A. Wood, K. Weiss, J. Moore, A. Schenker, P. Jain, Y. Ying, S. Kakiuchi, R. Hoshino, and A. Humphrey, "Autonomous cars: Self-driving the new auto industry paradigm," Morgan Stanley Research Global, Tech. Rep., 2013.

[7] D. Cho and J. K. Hedrick, "Automotive power train modelling for control," *Transactions of ASME, Journal of Dynamic Systems, Measurement and Control*, vol. 111, pp. 568–576, 1989.

[8] J. K. Hedrick, "Nonlinear controller design for automated vehicle applications," in *Proc. UKACC Int. Conj on Contr.'98*, Swansea, U. K., September 1998, pp. 23–31.

[9] E. F. Camacho and C. B. Alba, *Model Predictive Control*, 2nd ed., ser. Advanced Textbooks in Control and Signal Processing. Springer-Verlag London, 2007.

[10] B. Kouvaritakis and M. Cannon, *Model Predictive Control: Classical, Robust and Stochastic*, 1st ed., ser. Advanced Textbooks in Control and Signal Processing. Springer International Publishing, 2016.

[11] J. Kocijan, R. Murray-Smith, C. E. Rasmussen, and A. Girard, "Gaussian process model based predictive control," in *In Proceedings of 4th American Control Conference*, Boston, MA, 2004, pp. 2214–2218.

[12] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning.* MIT Press, 2006.

[13] J. Kocijan, B. Banko, B. Likar, A. Girard, R. Murray-Smith, and C. E. Rasmussen, "A case based comparison of identification with neural networks and gaussian process models," in *In Proceedings of IFAC ICONS conference*, vol. 1, 2003, pp. 137–142.

[14] R. M. Neal, *Bayesian Learning for Neural Networks*. Springer, 1996.

[15] A. Girard, C. Rasmussen, J. Q. Candela, and M. R. Smith, "Multiple-Step Ahead Prediction for Non Linear Dynamic Systems - A Gaussian Process Treatment with Propagation of the Uncertainty," *Advances in Neural Information Processing Systems*, vol. 15, pp. 545–552, 2002.

[16] J. Kocijan, a. Girard, B. Banko, and R. Murray-Smith, "Dynamic systems identification with Gaussian processes," *Mathematical and Computer Modelling of Dynamical Systems*, vol. 11, no. 4, pp. 411–424, 2005.

[17] M. Abbaszadeh and R. Solgi, "Constrained nonlinear model predictive control of a polymerization process via evolutionary optimization," *Journal of Intelligent Learning Systems and Applications*, vol. 06, no. 01, p. 10, February 2014.

[18] K. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *Neural Networks, IEEE Transactions on*, vol. 1, no. 1, pp. 4–27, Mar 1990.

[19] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its application to modelling and control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 15, no. 1, pp. 116–132, 1985.

[20] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P.-Y. Glorennec, H. k. Hjalmarsson, and A. Juditsky, "Nonlinear black-box modeling in system identification: a unified overview," *Automatica*, vol. 31, no. 12, pp. 1691–1724, 1995.

[21] J. Kocijan, "Gaussian Process Models for Systems Identification," in *9th International PhD Workshop on Systems and Control: Young Generation Viewpoint*, no. October, 2008.

[22] A. Grancharova, J. Kocijan, and T. A. Johansen, "Explicit stochastic nonlinear predictive control based on gaussian process models," in *Proceedings of the European Control Conference 2007*, 2007, pp. 2340–2347.

[23] R. Murray-Smith, D. Sbarbaro, C. E. Rasmussen, and A. Girard, "Adaptive, cautious, predictive control with gaussian process priors," in *In Proceedings of 13th IFAC Symposium on System Identification*, Rotterdam, 2003, pp. 1195–1200.

[24] B. Likar and J. Kocijan, "Predictive control of a gas-liquid separation plant based on a gaussian process model," *Computers and Chemical Engineering*, vol. 31, no. 3, pp. 142–152, 2007.

[25] J. Lourenço, J. Lemos, and J. Marques, "Control of neuromuscular blockade with gaussian process models," *Biomedical Signal Processing and Control*, vol. 8, no. 3, pp. 244–254, 2013.

[26] Open Source Robotics Foundation, "The robot operating system (ros)," http://www.ros.org/, 2015.

[27] A. Jadbabaie and J. Hauser, "On the stability of receding horizon control with a general terminal cost," *IEEE Transactions on Automatic Control*, vol. 50, no. 5, pp. 674–678, 2005.