Horizon-1 predictive control of networked controlled vehicle drivetrains

Preprints of the 18th IFAC World Congress
Milano (Italy) August 28 - September 2, 2011

Horizon-1 predictive control of networked controlled vehicle drivetrains

C. F. Caruntu * **M. Lazar** ** **S. Di Cairano** *** **R. H. Gielen** **
P. P. J. van den Bosch **

* *Department of Automatic Control and Applied Informatics, "Gheorghe Asachi" Technical University of Iasi, Romania (e-mail: caruntuc@tuiasi.ro)*
** *Department of Electrical Engineering, Eindhoven University of Technology, The Netherlands (e-mails: m.lazar@tue.nl, r.h.gielen@tue.nl, p.p.j.v.d.bosch@tue.nl)*
*** *Ford Research and Adv. Engineering, Dearborn, MI, USA (e-mail: sdicaira@ford.com)*

Abstract: Design of a controller that damps driveline oscillations, while compensating network-induced time-varying delays, can be a challenging problem considering that vehicle drivetrains are characterized by fast dynamics that are subject to physical and control constraints. As such, the goal of this paper is to provide a control design methodology that can cope with all these challenges and limitations and still yield an effective solution. To this end, firstly, a method for determining a worst case upper bound on the delays that can be introduced by a Controller Area Network (CAN) is presented. Then, a polytopic approximation technique is applied to obtain a discrete-time model of the closed-loop CAN system. Thirdly, a horizon-1 predictive controller based on flexible control Lyapunov functions is designed for the resulting model with polytopic uncertainty and hard constraints. Several tests performed on a benchmark model indicate that the proposed design methodology can handle both the performance/physical constraints and the strict limitations on the computational complexity.

Keywords: Vehicle drivetrain, Oscillations damping, Networked control systems, Predictive control, Lyapunov methods.

1. INTRODUCTION

When a vehicle is subjected to accelerations, the elasticity of the various components in the driveline may cause torsional vibrations or disturbances. In recent years, the greater demand for passenger comfort, which requires a reduction of the noise and vibration characteristics of vehicles, has led to the design of proper models for conventional vehicle drivetrains and to the development of different control strategies to minimize the effects of drivetrain oscillations: robust pole placement (Stewart et al., 2005), $\mathcal{H}_\infty$ optimization (Lefebvre et al., 2003), linear quadratic gaussian control design with loop transfer recovery (LQG/LTR) (Berriri et al., 2008) and even model predictive control (MPC) (Rostalski et al., 2007). Although the majority of the control strategies that are implemented on real vehicles are based on heuristics and look-up tables, it was shown that MPC has a large potential for control of automotive subsystems, e.g., mechatronic actuators (Di Cairano et al., 2007), driveline (Saerens et al., 2008), engine (Di Cairano et al., 2010). This is supported by the fast proliferation of powerful embedded components that enable complex real-time control algorithms.

All of the above control solutions assume that the sensors, controllers and actuators are directly connected, which is not realistic. Rather, in modern vehicles, the control signals from the controllers and the measurements from the sensors are exchanged using a communication network, e.g., Controller Area Network (CAN) or Flexray, among control system components. This brings up a new challenge on how to deal with the effects of the network-induced delays and packet losses in the control loop. The delays may be unknown and time-varying and may degrade the performances of control systems designed without considering delays, up to the point where they can even destabilize the closed-loop system.

As such, the problem considered in this paper is to minimize the oscillations of a vehicle drivetrain while compensating the time-varying delays introduced by CAN. Standard MPC strategies, which typically require a sufficiently long prediction horizon to assure stability and performance, would lead to solutions that are too complex and, at the same time, too conservative to be implemented in real-time control applications, keeping in mind that the physical plants from automotive applications have fast dynamics that require short sampling periods.

The proposed solution consists of three steps: first, a standard continuous-time model of the vehicle drivetrain is derived; second, a technique for estimating the maximum delay that can be induced by a CAN is presented, which enables the usage of the modeling technique developed in (Gielen et al., 2010) for networked control systems (NCS) via polytopic approximations; third, a horizon-1 MPC scheme is designed using the concept of flexible control Lyapunov functions (Lazar, 2009). A simulation experiment designed in collaboration with Ford Research and Advanced Engineering, US, validates the proposed approach and indicates that the developed scheme has the potential to meet the required real-time control specifications.

Notation and basic definitions. $\mathbb{R}$, $\mathbb{R}_+$, $\mathbb{Z}$ and $\mathbb{Z}_+$ are the real, non-negative real, integer and non-negative integer numbers.

For every $c \in \mathbb{R}$ and $\Pi \subseteq \mathbb{R}$, define $\Pi_{\geq c} := \{k \in \Pi \mid k \geq c\}$ and similarly $\Pi_{\leq c}$, $\mathbb{R}_\Pi := \Pi$ and $\mathbb{Z}_\Pi := \mathbb{Z} \cap \Pi$. For a vector $x \in \mathbb{R}^n$, let $\|\cdot\|$ denote an arbitrary p-norm and let $[x]_i, i \in \mathbb{Z}_{[1,n]}$ denote the i-th component of x. Let $\|x\|_\infty := \max_{i \in \mathbb{Z}_{[1,n]}} |[x]_i|$, where $|\cdot|$ denotes the absolute value. For a matrix $Z \in \mathbb{R}^{m \times n}$, let $Z^\top$ denote its transpose. Let $\mathbf{z} := \{z_l\}_{l \in \mathbb{Z}_+}$ with $z_l \in \mathbb{R}^n$ for all $l \in \mathbb{Z}_+$ denote an arbitrary sequence. Define $\|\mathbf{z}\| := \sup\{\|z_l\| \mid l \in \mathbb{Z}_+\}$ and $\mathbf{z}_{[0,k]} := \{z_l\}_{l \in \mathbb{Z}_{[0,k]}}$. A polyhedron, or a polyhedral set, in $\mathbb{R}^n$ is a set obtained at the intersection of a finite number of open and/or closed half-spaces. A polytope is a compact (closed and bounded) polyhedron. Let $\mathrm{Co}(\cdot)$ denote the convex hull and let $\mathrm{int}(\mathcal{S})$ denote the interior of an arbitrary set S. A function $\varphi : \mathbb{R}_+ \to \mathbb{R}_+$ belongs to class $\mathcal{K}$ if it is continuous, strictly increasing and $\varphi(0) = 0$. A function $\varphi : \mathbb{R}_+ \to \mathbb{R}_+$ is said to belong to class $\mathcal{K}_\infty$ if it is of class $\mathcal{K}$ and $\lim_{s \to \infty} \varphi(s) = \infty$.

2. VEHICLE DRIVETRAIN MODELING

This section gives a brief overview of the dynamical model of the drivetrain utilized in Section 3 for controller design. In the literature numerous models were proposed during the last years for conventional powertrains such as: two inertias models, one inertia corresponding to the engine and the other inertia to the vehicle mass and the wheels, (Baumann et al., 2006; Rostalski et al., 2007; Saerens et al., 2008) and three inertias models, one representing the engine, the second representing the gearbox and the final drive and the third inertia corresponding to the vehicle mass and the wheels (Van Der Heijden et al., 2007).

Since this paper focuses on damping the drivetrain oscillations, there is no need to model dynamics that are much faster than the longitudinal dynamics of the vehicle, so the developed model is simple enough for controller design, also knowing that in the designing phase the time-varying delays have to be considered, but complex enough to capture the essential dynamics of the drivetrain. In this case the oscillations originating from driveline flexibilities are of interest.

The two mass model illustrated in Fig. 1, which is a schematic representation of a simplified drivetrain with engaged clutch, has two inertias, one for the engine and gearbox, i.e., J_{eg}, and the other one representing the contributions from the vehicle and from the driving wheels, i.e., J_v, connected through a flexible driveshaft. The engine generates a torque, i.e., T_e, which is the control signal and assumed to be available on demand, e.g., engine torque is requested from the engine controller unit and obtained by modulating spark timing and airflow and is transmitted towards the wheels through the driveline. The drivetrain flexibility is given by the driveshaft and the halfshafts, which transmit the received torque from the gearbox to the wheels. The driving wheels are the final components of the drivetrain, which have the purpose of moving the vehicle by defeating the friction forces with the road surface and the aerodynamic drag.

The engine speed dynamics, modeled as a single inertia system, are described by

$$J_{eg}\dot{\omega}_e = T_e - b_e\omega_e - T_f/i_{tot}, \tag{1}$$

where $J_{eg} := J_e + \frac{J_g}{i_{tot}^2}$ is the equivalent inertia of the engine-gearbox subsystem, with J_g the gearbox inertia and $i_{tot} := i_g i_f$ the overall transmission ratio from the gearbox and the final reduction gear, ω_e is the engine speed, b_e is the engine damping coefficient and T_f is the torque in the flexible driveshaft.

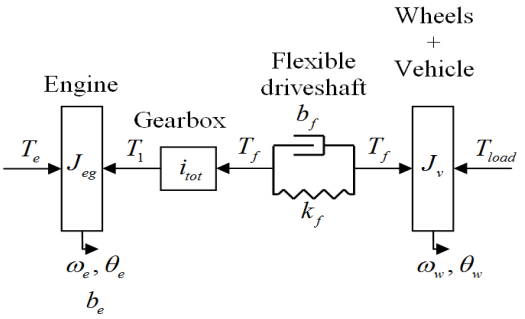

Fig. 1. Simplified drivetrain schematic representation.

The torque in the driveshaft, which is modeled as a spring and damper system, can be expressed as

$$T_f = b_f\left(\omega_e/i_{tot} - \omega_w\right) + k_f\left(\theta_e/i_{tot} - \theta_w\right), \tag{2}$$

where b_f and k_f are the damping coefficient and the elasticity factor of the driveshaft, θ_e is the engine angle and θ_w and ω_w are the wheel angle and speed, respectively.

Analogously, the wheel and vehicle body dynamics are modeled as a single inertia with

$$J_v\dot{\omega}_w = T_f - T_{load}, \tag{3}$$

and the tire was accounted as a rolling element without slip, i.e., $v_v = r_w\omega_w$, where v_v is the vehicle speed and r_w is the effective wheel radius. The vehicle inertia can be obtained by adding the wheel inertia to the equivalent inertia of the vehicle mass, i.e.,

$$J_v = J_w + m_v r_w^2, \tag{4}$$

where m_v is the mass of the vehicle and J_w is the wheel inertia.

The load torque is modeled as

$$T_{load} = T_{airdrag} + T_{roll} + T_{grade}, \tag{5}$$

where $T_{airdrag}$ is the aerodynamic drag torque of the vehicle body, T_{roll} is the rolling torque of the tires and T_{grade} is the torque due to road grade. For a more detailed definition of these torques the interested reader is referred to (Caruntu et al., 2011).

By introducing the engine speed, the wheel speed and the torsion in the driveshaft, also called axle wrap, as state variables, i.e.,

$$x_1^m = \omega_e, \quad x_2^m = \omega_w, \quad x_3^m = \frac{\theta_e}{i_{tot}} - \theta_w, \tag{6}$$

the drivetrain model can be written in a state space form as

$$\dot{x}^m(t) = A_c^m x^m(t) + b_c^m u^m(t) + f_c^m, \tag{7}$$

where $x^m = \begin{pmatrix} x_1^m & x_2^m & x_3^m \end{pmatrix}^\top$ is the system state, $u^m \in \mathbb{R}$ is the system input, $f_c^m = \begin{pmatrix} 0 \\ -\frac{T_{roll}+T_{grade}}{J_v} \\ 0 \end{pmatrix}$, $b_c^m = \begin{pmatrix} \frac{1}{J_{eg}} \\ 0 \\ 0 \end{pmatrix}$ and $A_c^m = \begin{pmatrix} -\frac{b_e}{J_{eg}} - \frac{b_f}{i_{tot}^2 J_{eg}} & \frac{b_f}{i_{tot} J_{eg}} & -\frac{k_f}{i_{tot} J_{eg}} \\ \frac{b_f}{i_{tot} J_v} & -\frac{b_a + b_f}{J_v} & \frac{k_f}{J_v} \\ \frac{1}{i_{tot}} & -1 & 0 \end{pmatrix}$.

The engine torque (control input) is restricted by lower and upper bounds and by a torque rate constraint (see (Caruntu et al., 2011) for details), i.e.,

$$0 \leq u^m(t) \leq T_e^{max}, \tag{8a}$$

$$T_e^m \leq \dot{u}^m(t) \leq T_e^M, \tag{8b}$$

where T_e^{max} is the maximum torque that can be generated by the internal combustion engine and T_e^m, T_e^M are torque rate bounds.

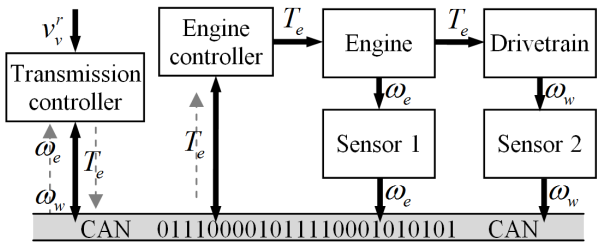

Fig. 2. Drivetrain control architecture.

The engine and wheel speeds are bounded, i.e.,

$$\omega_e^{min} \leq x_1^m(t) \leq \omega_e^{max}, \tag{9a}$$

$$\omega_w^{min} \leq x_2^m(t) \leq \omega_w^{max}, \tag{9b}$$

where ω_e^{min}, ω_e^{max} are the idle and the maximum speed of the engine, respectively, and ω_w^{min}, ω_w^{max} are the minimum and the maximum speed of the wheel, respectively.

The control objective is to reach a desired value of the wheel speed, i.e., x_2^{ss}, as fast as possible and with minimum overshoot. Note that for a desired wheel speed value x_2^{ss}, it is possible to obtain the steady-state engine speed x_1^{ss}, axle wrap x_3^{ss} and engine torque u^{ss}.

To facilitate the implementation of the horizon-1 MPC algorithm of (Lazar, 2009), a coordinate transformation is performed on (7), i.e.,

$$x(t) = x^m(t) - x^{ss}(t), u(t) = u^m(t) - u^{ss}(t), \tag{10}$$

where $x^{ss} = (\, x_1^{ss} \;\; x_2^{ss} \;\; x_3^{ss} \,)^\top$, which yields the following system description

$$\dot{x}(t) = A_c x(t) + b_c u(t), \tag{11}$$

where A_c and b_c are the transformed system matrices.

Discretizing model (11) yields

$$x_{k+1} = A_d x_k + b_d u_k, \tag{12}$$

for all $k \in \mathbb{Z}_+$, where A_d and b_d are the transformed and discretized system matrices and x_k and u_k are the state and the input of the system at time instant $k \in \mathbb{Z}_+$.

Using (10) and (12), the constraints given in (8) and (9) can be converted to:

$$\begin{aligned} x_{1,k} \in [\underline{b}^{x_1}, \bar{b}^{x_1}], x_{2,k} \in [\underline{b}^{x_2}, \bar{b}^{x_2}], \\ u_k \in [\underline{b}^{u}, \bar{b}^{u}], \Delta u_k \in [\underline{b}^{\Delta u}, \bar{b}^{\Delta u}], \end{aligned} \tag{13}$$

for suitably defined constants, where $\Delta u_k = u_k - u_{k-1}$.

The control objective can now be formulated as stabilization of the zero equilibrium of system (12), while fulfilling the constraints (13).

3. DRIVETRAIN CONTROL ARCHITECTURE

The complete network-controlled architecture considered in this paper, which is graphically depicted in Fig. 2, consists of the following operations:

- the sensors measure the outputs of the system and send the samples to the controller through CAN;
- the transmission controller receives the measurements from the sensors and the desired velocity reference $v_v^r = r_w \omega_w^r$ and computes the required torque, while handling the physical/control constraints and the delays;
- the control signal, i.e., the torque computed by the controller, is sent to the engine controller through CAN;
- the engine controller actuates the spark timing and airflow as requested for driveline control.

3.1 CAN-induced delays in automotive applications

To assure the consistent operation of a vehicle, hard real-time controller transmission deadlines have to be met, so there appears the need to determine reasonable bounds for delays of safety critical data transmission. The result presented in (Klehmet et al., 2008), which aims to provide a method for calculating the worst-case response time of each message sent on CAN in automotive applications, is used to determine the upper bound of the delays that are induced by the communication network, i.e.,

$$\tau_{large,j} \leq \frac{(j+2)l}{R - \sum_{i=0}^{j-1}(l/c_i)}, \tag{14}$$

where j is the priority of the node for which the upper bound is calculated, $l = 136$ bits denotes the maximum frame length, $R = 500$ kbps is the rate of a high-speed CAN and c_i is the cycle length of the i-th priority message. A cycle length c_n, corresponding to a message of priority n, represents the period after which the message is repeated.

3.2 Networked control system model

Given the explicit upper bound derived previously, the task is now to obtain a discrete-time model of the network-controlled architecture that accounts for time-varying delays within the provided bound. To this end, consider the continuous-time drivetrain system (11) with input delay induced by CAN, i.e.,

$$\begin{aligned} \dot{x}(t) &= A_c x(t) + b_c u(t), \\ u(t) &= u_k, \forall t \in [t_k + \tau_k, t_{k+1} + \tau_{k+1}], \end{aligned} \tag{15}$$

where $t_k = kT_s, k \in \mathbb{Z}_+, T_s \in \mathbb{R}_+$ denotes the sampling period and assume that $u(t) = u_0$ for all $t \in [0, \tau_0]$ with $u_0 \in \mathbb{R}^m$ some predetermined constant vector. $u_k \in \mathbb{R}$ is the control action generated at time $t = t_k$. $\tau_k \in \mathbb{R}_{[0,\tau_{large}]}$ denotes the delay induced by the network at time $k \in \mathbb{Z}_+$ and $\tau_{large} \in \mathbb{R}_+$ is the maximal possible delay.

Consider the maximum delay determined using (14) as $\tau_{large} = (\Upsilon + \upsilon)T_s$, where $\Upsilon \in \mathbb{Z}_+$ and $\upsilon \in \mathbb{R}_{[0,1)}$. Assuming that $u_k = \psi_k$ for all $k \in \mathbb{Z}_{[-\Upsilon-1,-1]}$ with $\boldsymbol{\psi}_{[-\Upsilon-1,-1]}$ some predetermined vector, the discretized model is

$$\begin{aligned} x_{k+1} = A_d x_k + b_d u_k + \Delta_0(\tau_k)(u_{k-1} - u_k) \\ + \Delta_1(\tau_k)(u_{k-2} - u_{k-1}) + \dots \\ + \Delta_\Upsilon(\tau_k)(u_{k-\Upsilon-1} - u_{k-\Upsilon}), \end{aligned} \tag{16}$$

where $A_d = e^{A_c T_s}$, $b_d = \int_0^{T_s} e^{A_c(T_s-\theta)} d\theta b_c$,

$$\Delta_i(\tau_k) := \begin{cases} 0, & \tau_{k-i} - iT_s \leq 0 \\ \int_0^{\tau_{k-i}-iT_s} e^{A_c(T_s-\theta)} d\theta b_c, & 0 < \tau_{k-i} - iT_s < T_s \\ \int_0^{T_s} e^{A_c(T_s-\theta)} d\theta b_c, & T_s \leq \tau_{k-i} - iT_s \end{cases} \tag{17}$$

for all $k \in \mathbb{Z}_+$ and $i \in \mathbb{Z}_{[0,\Upsilon]}$ with

$$\tau_k \geq \tau_{k-1} - T_s. \tag{18}$$

The inequality (18) is due to the fact that within the same message ID group, the messages sent on CAN are queued and they are resent until they are received, i.e., they can not be lost in the network. Thus, a message transmitted at a certain time can never arrive before a message that was transmitted at a previous time. Notice that this feature of CAN is different from NCS setups considered in the control literature, see, e.g, (Gielen and

Lazar, 2009), where it is assumed that a newer control update can arrive before an older one and thus the latter can be ignored.

A polytopic over-approximation of the nonlinear functions $\Delta_i(\tau_k)$ can be found using the Cayley-Hamilton theorem as presented in (Gielen et al., 2010). Therefore, the following set is defined

$$\Delta^{\bar{\tau}} := \text{Co}(\{\bar{\Delta}_l\}_{l\in\mathbb{Z}_{[1,L]}}), \tag{19}$$

with $\bar{\tau} \in \mathbb{R}_{[0,T_s]}$, $\bar{\Delta}_l \in \mathbb{R}^{n\times m}$ such that $\Delta_i(\tau_k) \in \Delta^{\bar{\tau}}$ for all $\tau_k \in \mathbb{R}_{[0,\bar{\tau}]}$ and $L \in \mathbb{Z}_{\geq 1}$ is finite. In (Gielen et al., 2010) several methods to create the polytope (19) were assessed. This reference is referred for further details and assume for the remainder of this paper that the polytopic set (19) is known.

As $\Delta_i(\tau_k) \in \Delta^{T_s}$ for all $i \in \mathbb{Z}_{[0,\Upsilon-1]}$ and $\Delta_\Upsilon(\tau_k) \in \Delta^{\upsilon T_s}$, it is obtained that (16) is contained in

$$x_{k+1} \in \phi(x_k, \mathbf{u}_{[k-\Upsilon-1,k]}), \quad k \in \mathbb{Z}_+, \tag{20}$$

where the inputs $\mathbf{u}_{[-\Upsilon-1,0]}$ are equal to some fixed, predetermined values and

$$\begin{aligned}\phi(x_k, \mathbf{u}_{[k-\Upsilon-1,k]}) := \{&A_d x_k + b_d u_k \\ &+ \sum_{i=0}^{\Upsilon} \Delta_i(u_{k-i-1} - u_{k-i}) \mid \\ &\Delta_i \in \Delta^{T_s},\, i \in \mathbb{Z}_{[0,\Upsilon-1]},\ \Delta_\Upsilon \in \Delta^{\upsilon T_s}\}.\end{aligned} \tag{21}$$

Observe that the input vectors $\mathbf{u}_{[k-\Upsilon-1,k]}$ are known at time $k \in \mathbb{Z}_+$. The above model will be used in the next section for controller design purposes.

4. HORIZON-1 PREDICTIVE CONTROLLER DESIGN

In this section, a horizon-1 MPC scheme for vehicle drivetrain oscillations damping that employs a flexible Lyapunov function (Lazar, 2009) to attain stability and high regulation performance is proposed. Furthermore, it is shown that for a particular choice of the Lyapunov function candidate the proposed design leads to a low-complexity linear program that can be solved efficiently within the required time range, while strictly enforcing the constraints and considering the delays induced by CAN, via the usage of the model described in Subsection 3.2. Similar techniques were applied in (Gielen and Lazar, 2009), so the interested reader is referred to this paper for further details.

4.1 Horizon-1 MPC Set-up

Consider the non-autonomous system

$$x_{k+1} \in \phi(x_k, \mathbf{u}_{[k-\Upsilon-1,k]}), \quad k \in \mathbb{Z}_+, \tag{22}$$

where $x_k \in \mathbb{X} \subseteq \mathbb{R}^n$ is the state at the discrete time instant k and $\mathbf{u}_{[k-\Upsilon-1,k]} \in \mathbb{U}^{\Upsilon+2} := \mathbb{U} \times \ldots \times \mathbb{U} \subseteq \mathbb{R}^m \times \ldots \times \mathbb{R}^m$, are the control inputs starting with discrete time instant $k - \Upsilon - 1$ up to and including k. The mapping $\phi : \mathbb{R}^n \times \mathbb{R}^m \times \ldots \times \mathbb{R}^m \rightrightarrows \mathbb{R}^n$ is an arbitrary set-valued function with $\phi(0, \mathbf{0}_{[-\Upsilon-1,0]}) = \{0\}$. We assume that $0 \in \text{int}(\mathbb{X})$ and $0 \in \text{int}(\mathbb{U})$. Next, let $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$ and let $\rho \in \mathbb{R}_{[0,1)}$.

Definition 1. A function $V : \mathbb{R}^n \to \mathbb{R}_+$ that satisfies

$$\alpha_1(\|x\|) \leq V(x) \leq \alpha_2(\|x\|), \quad \forall x \in \mathbb{R}^n, \tag{23}$$

and for which there exists a control law, possibly set-valued, $\pi : \mathbb{R}^n \rightrightarrows \mathbb{U}$ such that $x^+ \in \mathbb{X}$ and that

$$V(x^+) \leq \rho \max_{\theta\in\mathbb{Z}_{[-\Upsilon-1,0]}} V(x_\theta), \tag{24}$$

for all $x_\theta \in \mathbb{X}$, $\theta \in \mathbb{Z}_{[-\Upsilon-1,0]}$, $u_\theta \in \pi(x_\theta)$ and all $x^+ \in \phi(x, \mathbf{u}_{[-\Upsilon-1,0]})$ is called a *control Lyapunov-Razumikhin function (CLRF)* for the difference inclusion corresponding to system (22). □

Consider the following inequality corresponding to (24)

$$V(x_{k+1}) \leq \rho \max_{\theta\in\mathbb{Z}_{[-\Upsilon-1,0]}} V(x_{k+\theta}) + \lambda_k, \tag{25}$$

for all $k \in \mathbb{Z}_+$ and all $x_{k+1} \in \phi(x_k, \mathbf{u}_{[k-\Upsilon-1,k]})$. Here λ_k is a variable which allows for additional freedom in the evolution of the CLRF, i.e., it can increase if (24) is too conservative at time instant $k \in \mathbb{Z}_+$, possibly due to active state/input constraints. Based on (25) the following optimization problem is formulated.

Let $\alpha_3, \alpha_4 \in \mathcal{K}_\infty$ and $J : \mathbb{R} \to \mathbb{R}_+$ be an arbitrary function such that $\alpha_3(|\lambda|) \leq J(\lambda) \leq \alpha_4(|\lambda|)$ for all $\lambda \in \mathbb{R}$. Let $V(\cdot)$ be a candidate CLRF for system (20).

Problem 2. Assume that at time $k \in \mathbb{Z}_+$, x_k and $\mathbf{u}_{[k-\Upsilon-1,k-1]}$ are known and minimize the cost $J(\lambda_k)$ over u_k and λ_k subject to

$$u_k \in \mathbb{U},\ \phi(x_k, \mathbf{u}_{[k-\Upsilon-1,k]}) \subseteq \mathbb{X},\ \lambda_k \geq 0, \tag{26a}$$

$$V(x_{k+1}) \leq \rho \max_{\theta\in\mathbb{Z}_{[-\Upsilon-1,0]}} V(x_{k+\theta}) + \lambda_k, \tag{26b}$$

for all $x_{k+1} \in \phi(x_k, \mathbf{u}_{[k-\Upsilon-1,k]})$, with $\rho \in \mathbb{R}_{[0,1)}$. □

Let $\pi(x_k) := \{u_k \in \mathbb{R}^m \mid \exists \lambda_k \in \mathbb{R} \text{ s.t. (26) holds}\}$ and let $\phi_{cl}(x_k, \pi(x_k), \mathbf{u}_{[k-\Upsilon-1,k-1]}) := \{\phi(x_k, u, \mathbf{u}_{[k-\Upsilon-1,k-1]}) \mid u \in \pi(x_k)\}$. Furthermore, let $\mathcal{V}_\Gamma := \{x \in \mathbb{R}^n \mid V(x) \leq \Gamma\}$ for any $\Gamma \in \mathbb{R}_+$. Let λ_k^* denote the optimum in Problem 2 for all $k \in \mathbb{Z}_+$.

Next, the main stability result is stated.

Theorem 3. Suppose that $V(\cdot)$ is a function that satisfies (23) and that $\mathbb{X}$ and $\mathbb{U}$ are bounded. Furthermore, suppose that Problem 2 is feasible for all $(x_0, \mathbf{u}_{[-\Upsilon-1,0]}) \in \mathbb{X} \times \mathbb{U}^{\Upsilon+2}$ and that $\lim_{k\to\infty} \lambda_k^* = 0$. Then, the origin of the difference inclusion

$$x_{k+1} \in \phi_{cl}(x_k, \pi(x_k), \mathbf{u}_{[k-\Upsilon-1,k-1]}), \quad k \in \mathbb{Z}_+ \tag{27}$$

is attractive. Moreover, if $\exists \Gamma \in \mathbb{R}_{>0}$ such that $V(\cdot)$ is a CLRF for initial conditions in $\mathcal{V}_\Gamma$ for system (20), then system (27) is asymptotically stable in $\mathbb{X}$. □

The proof of Theorem 3 follows from standard arguments employed in proving input-to-state stability and Lyapunov stability and consideration of the time-delay setting, and is therefore omitted here. The key of the stability proof is the limiting condition $\lim_{k\to\infty} \lambda_k^* = 0$, which is guaranteed by the following lemma.

Lemma 4. Let $\Omega \in \mathbb{R}_+$ be a fixed constant to be chosen a priori and let $\rho \in \mathbb{R}_{[0,1)}$. If

$$0 \leq \lambda_k \leq \rho(\lambda_{k-1}^* + \rho^{k-1}\Omega), \quad \forall k \in \mathbb{Z}_+, \tag{28}$$

then $\lim_{k\to\infty} \lambda_k = 0$.

For a proof of Lemma 4 we refer to (Caruntu et al., 2011) for brevity. By augmenting Problem 2 with constraint (28) the property $\lim_{k\to\infty} \lambda_k^* = 0$ is thus guaranteed, which is sufficient for asymptotic stability.

4.2 Implementation as a single linear program

In this subsection it is shown how a CLRF defined using infinity norms and corresponding control law can be obtained.

Moreover, it is indicated how the optimization problem derived above can be formulated as a single linear program (LP).

Consider the following cost function to be minimized

$$\begin{aligned} J_1(x_k, \mathbf{u}_{[k-\Upsilon-1,k]}, \lambda_k) &:= J_{\text{MPC}}(x_k, \mathbf{u}_{[k-\Upsilon-1,k]}) + J(\lambda_k) \\ &:= \|Qx_{k+1}\|_\infty + \|Ru_k\|_\infty + G\lambda_k, \end{aligned} \tag{29}$$

where $x_{k+1} \in \phi(x_k, \mathbf{u}_{[k-\Upsilon-1,k]})$ and the matrices Q and R are known full-column rank matrices of appropriate dimensions and $G \in \mathbb{R}_{>0}$.

The constraints (13) can be specified as

$$\begin{aligned} 0 - u^{ss} \le u_k \le T_e^{max} - u^{ss}, \\ -T_e^{\Delta} \le \Delta u_k \le T_e^{\Delta}, \\ x^{min} \le H(A_d x_k + b_d u_k \\ + \sum_{i=0}^{\Upsilon} \Delta_i(\tau_k)(u_{k-i-1} - u_{k-i})) \le x^{max} \end{aligned} \tag{30}$$

where $x^{min} = \begin{pmatrix} \omega_e^{min} - x_1^{ss} \\ \omega_w^{min} - x_2^{ss} \end{pmatrix}$, $x^{max} = \begin{pmatrix} \omega_e^{max} - x_1^{ss} \\ \omega_w^{max} - x_2^{ss} \end{pmatrix}$, $H = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$ and $\Delta_i(\tau_k) \in \Delta^{T_s}$ for $i \in \mathbb{Z}_{[0,\Upsilon-1]}$ and $\Delta_\Upsilon(\tau_k) \in \Delta^{\upsilon T_s}$.

Note that constraint satisfaction of the difference inclusion (20) can be guaranteed by guaranteeing constraint satisfaction for all combinations of the vertices of the sets Δ^{T_s} and $\Delta^{\upsilon T_s}$.

Now, consider the following infinity-norm based candidate CLRF, i.e.,

$$V(x) = \|Px\|_\infty, \tag{31}$$

where $P \in \mathbb{R}^{p \times n}$, $p \ge n$, is a full-column rank matrix to be determined. This function satisfies (23) with $\alpha_1(s) = \frac{\underline{\sigma}}{\sqrt{p}} s$, where $\underline{\sigma}$ is the smallest singular value of P, and with $\alpha_2(s) = \|P\|_\infty s$. Substituting (20) and (31) in (26) yields

$$\begin{aligned} \|P(A_d x_k + b_d u_k + \sum_{i=0}^{\Upsilon} \Delta_i(\tau_k)(u_{k-i-1} - u_{k-i}))\|_\infty \le \\ \le \rho \max_{\theta \in \mathbb{Z}_{[-\Upsilon-1,0]}} \|Px_{k+\theta}\|_\infty + \lambda_k, \end{aligned} \tag{32}$$

where $\mathbf{x}_{[k-\Upsilon-1,k]}$, P, $\mathbf{u}_{[k-\Upsilon-1,k-1]}$ and $\rho \in \mathbb{R}_{[0,1)}$ are known. Furthermore, $\Delta_i(\tau_k) \in \Delta^{T_s}$ for $i \in \mathbb{Z}_{[0,\Upsilon-1]}$ and $\Delta_\Upsilon(\tau_k) \in \Delta^{\upsilon T_s}$.

Although (29) and (32) appear to be nonlinear in the optimization variables, due to using horizon 1 only, the optimization problem can be recast as a LP via an appropriate set of equivalent linear inequalities. By the infinity norm definition, for $\|x\|_\infty \le c$, $x \in \mathbb{R}^n$, to be satisfied, it is necessary and sufficient to require that $\pm[x]_j \le c$ for all $j \in \{1, 2, ..., n\}$.

Solving Problem 2, which includes minimizing the cost function (29), can be reformulated as the following problem.

Problem 5.

$$\min_{u_k, \lambda_k} (\epsilon_k^1 + \epsilon_k^2 + \epsilon_k^3) \tag{33}$$

subject to (28), (30), (32), and

$$\begin{aligned} \pm[Q(A_d x_k + b_d u_k + \sum_{i=0}^{\Upsilon} \Delta_i(\tau_k) \times \\ \times (u_{k-i-1} - u_{k-i}))]_j \le \epsilon_k^1, j \in \{1, 2, 3\}, \end{aligned} \tag{34a}$$

$$\pm Ru_k \le \epsilon_k^2, \tag{34b}$$

$$G\lambda_k \le \epsilon_k^3, \tag{34c}$$

where $\Delta_i(\tau_k) \in \Delta^{T_s}$ for $i \in \mathbb{Z}_{[0,\Upsilon-1]}$ and $\Delta_\Upsilon(\tau_k) \in \Delta^{\upsilon T_s}$.

When (34) is imposed for all combinations of the vertices of the sets Δ^{T_s} and $\Delta^{\upsilon T_s}$ Problem 5 is a LP. Indeed, $\mathbf{x}_{[k-\Upsilon-1,k]}$, λ_{k-1}^* and all $\mathbf{u}_{[k-\Upsilon-1,k-1]}$ are known at time $k \in \mathbb{Z}_+$ and thus, all constraints are linear in the unknowns u_k, λ_k and $\epsilon_k^{1,2,3}$.

The horizon-1 MPC algorithm can be summarized as follows.

Algorithm 1. At each instant $k \in \mathbb{Z}_+$:
Step 1: Obtain the current state x_k;
Step 2: Solve the LP (33)-(34) and pick any feasible solution u_k^f;
Step 3: Use $u_k = u_k^f$ as control action.

5. SIMULATION RESULTS

The discrete-time model (21) was implemented in Matlab and three different control strategies were applied in order to damp driveline oscillations: a PID controller, an explicit MPC controller and the horizon-1 predictive controller proposed in this paper. The sampling period of the system was chosen as $T_s = 0.01$s and the values of the parameters used in the simulations were obtained with the help of Ford Research and Advanced Engineering, US, and are given in Table 1.

The upper bound of the delays that are induced by CAN was calculated using the methodology described in Subsection 3.1, which resulted in $\tau_{large} = 1.7T_s = 0.017$s. The delays are time-varying and uniformly distributed in the interval $[0, \tau_{large}]$, as shown in Fig. 3. The delays were considered in the design phase only for the horizon-1 predictive controller proposed in this paper, but they were introduced in simulations for all the strategies. The purpose for applying the control strategies that do not take delay into account is to illustrate that these network-induced time-varying delays can have adverse effects

Table 1. Simulation vehicle parameter values

Symbol	Value	Measure Unit	Description
J_e	0.184	[kg ·m^2]	Engine inertia
J_g	1.1828	[kg ·m^2]	Transmission parts inertia
J_w	5.38	[kg ·m^2]	Wheel inertia
b_f	42	[kg · m^2 · s^{-1} · rad^{-1}]	Flexible driveshaft damping
k_f	6000	[kg · m^2 · s^{-2} · rad^{-1}]	Flexible driveshaft stiffness
b_e	0.15	[kg · m^2 · s^{-1} · rad^{-1}]	Engine damping coefficient
b_a	2.7	[kg · m^2 · s^{-1} · rad^{-1}]	Approximation factor
i_g	3.778		Gearbox ratio (1$^{\text{st}}$ gear)
i_f	3.667		Final drive ratio
m_v	1094	[kg]	Vehicle mass
r_w	0.281	[m]	Wheel radius
T_e^{max}	120	[N · m]	Maximum engine torque
T_e^{Δ}	2.5	[N · m]	Maximum engine torque increase/decrease
ω_e^{min}	62.83	[rad· s^{-1}]	Engine idle speed
ω_e^{max}	523.6	[rad· s^{-1}]	Maximum engine speed
ω_w^{min}	0	[rad· s^{-1}]	Minimum wheel speed
ω_w^{max}	247.1	[rad· s^{-1}]	Maximum wheel speed

on the controllers performances as it is shown in what follows. On the other hand, to the best of the authors' knowledge, a systematic method for handling time-varying delays is not yet available within existing explicit MPC design techniques.

The control objective is to reach a desired speed reference in a short time, but, at the same time, to increase the passenger comfort by reducing the oscillations that occur when the vehicle is subjected to accelerations. All control strategies were tuned to have the same response regarding the wheel speed, so the results could be discussed w.r.t. the drivetrain oscillations.

A PID controller was designed and tuned to have a fast response, which yielded the proportional, integral and derivative terms $K_R = 13$, $T_i = 9$ and $T_d = 0.1$, respectively. Also, an explicit MPC controller was designed for the delay-free affine model and the same operating constraints using the MultiParametric Toolbox for Matlab. The cost function

$$\min_{\mathbf{u}_{[0,N-1]}} \left(\|P_N x_N\|_\infty + \sum_{i=0}^{N-1} \|Q_x x_i\|_\infty + \|R_u u_i\|_\infty \right), \quad (35)$$

with $P_N = 11I_3$, $Q_x = 0.1I_3$, where I_3 is the 3-dimensional identity matrix, and $R_u = 0.5$ was employed in the corresponding finite horizon optimization problem. A feasible solution to the corresponding multiparametric LP problem could be obtained for a maximum prediction horizon of two sampling periods ($N = 2$), which resulted in a controller defined by 614 regions. For a prediction horizon larger than 2, despite using a powerful workstation and several robust solvers, a solution could not be obtained, which reflects the non-trivial nature of the considered case study. Moreover, the synthesis method based on the uncertain linear model suffers from an exponential increase in complexity when delays larger than the sampling period are considered. Furthermore, time-varying delays cannot be incorporated in the design of the explicit MPC controller.

The horizon-1 predictive controller was designed using the following weight matrices of the cost (29): $Q = 11I_3$, $R = 0.5$ and $G = 1$. The technique presented in (Gielen and Lazar, 2009) was used for the off-line computation of the infinity norm based local CLRF $V(x) = \|Px\|_\infty$ for $\rho = 0.99$ and the affine model of the drivetrain in closed-loop with $u_k := Kx_k$. The following matrices were obtained:

$$P = \begin{pmatrix} 2.2669 & 32.1093 & 946.2197 \\ 2.5314 & -71.6993 & -363.0408 \\ 2.4062 & 81.7867 & 741.4772 \end{pmatrix}, \quad (36)$$
$$K = \begin{pmatrix} -6.3471 & 45.7387 & -229.6221 \end{pmatrix}.$$

Note that the control law $u_k := Kx_k$ was only employed off-line, to calculate the weight matrix P of the local CLRF $V(\cdot)$, and it was never used for controlling the system. The evolution of the CLRF relaxation variable λ_k^* and the corresponding upper bound defined by (28) for $\rho = 0.99$, $\lambda_0^* \approx 50$ and $\Omega = 350$ is shown in Fig. 4, for this particular simulation. Notice that this leads to an initial upper bound on λ_k at time $k = 1$ approximately equal to 400, as it can be seen in Fig. 4. Therein, it can also be observed that λ_k^* may be small or even 0 for some time after which it is allowed to increase again, as long as this does not violate the upper bound. However as $k \to \infty$, λ_k^* is forced to converge to 0, which in turn implies asymptotic stability as guaranteed by Theorem 3.

Clearly, no nominal stability guarantee (i.e., in the absence of delay) can be obtained for the affine system in closed-loop with the PID controller. The nominal stability of the closed-loop system that corresponds to the explicit MPC scheme was analyzed a posteriori, but the explicit MPC controller failed all the stability tests that can be performed with the Multi Parametric Toolbox (e.g., piecewise quadratic, linear and even polynomial Lyapunov functions were searched for). For the proposed horizon-1 MPC scheme, recursive feasibility implies asymptotic stability and due to the employed model, this stability guarantee is valid for all considered delay realizations. However, recursive feasibility is not a priori guaranteed and hinges mainly on the constraint (28) on the future evolution of λ_k^*. For the considered case study, through extensive simulations, the employed $\Omega = 350$ proved to be large enough to guarantee recursive feasibility for a wide range of operating scenarios.

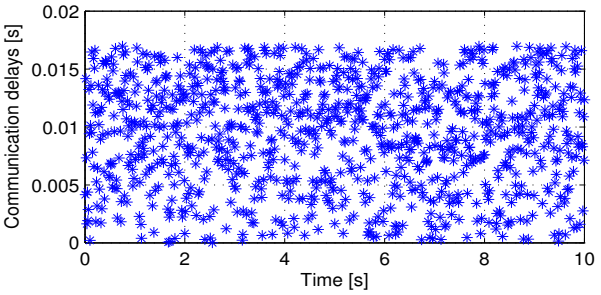

Fig. 3. Delays induced by CAN.

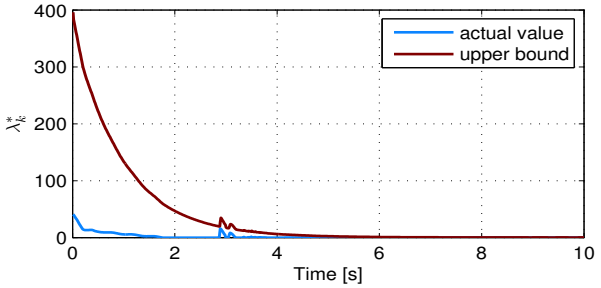

Fig. 4. History of λ_k^* throughout the simulation.

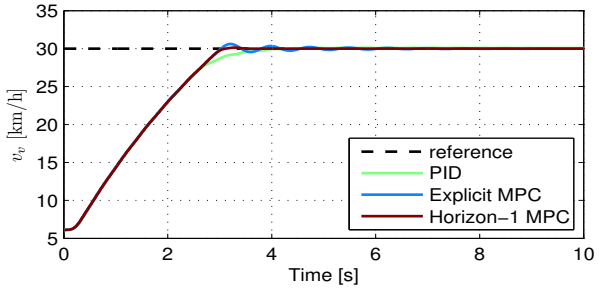

Fig. 5. Vehicle velocity.

In what follows the performance of the resulting closed-loop systems for each technique is analyzed using the trajectories plotted in Fig. 5, Fig. 6 and Fig. 7. The corresponding worst case time needed for computation of the control input is well within the allowed sampling interval for all considered control algorithms. This is a notable achievement for the horizon-1 MPC controller, as it takes into account in the designing phase the time-varying delays that appear on CAN (which in turn yields a more complex prediction model).

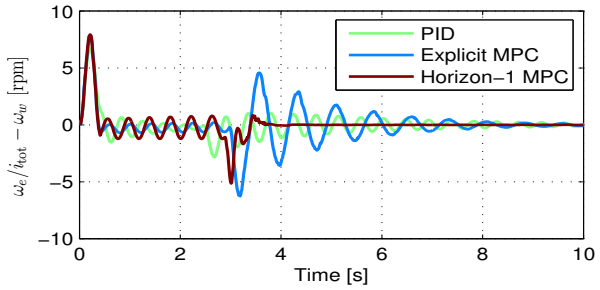

Fig. 6. Speed difference.

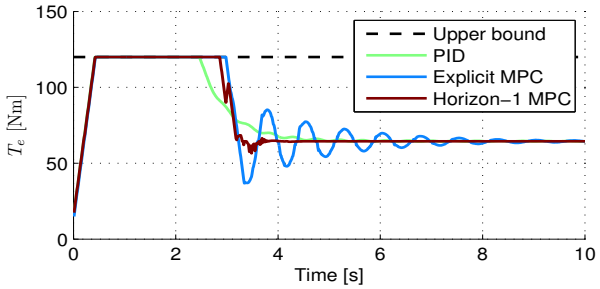

Fig. 7. Engine torque (control signal).

In Fig. 5, the vehicle velocity is represented, as obtained by the three different controllers, and it can be seen that in terms of rising time the controllers have almost the same response.

In Fig. 6 one can see that for the first 2.5 seconds the difference between the engine speed divided by the total gear ratio i_{tot} and the wheel speed is almost the same for the three controllers. After that, the driveline oscillations go to 0 for the horizon-1 predictive controller proposed in this paper, but the amplitude of the oscillations obtained with the explicit MPC controller increases by approximately a factor 4 w.r.t. the ones obtained with horizon-1 predictive controller and this difference in speeds keeps on oscillating further for both the the PID and the explicit MPC controller. These increased oscillations amplitude and frequency are undesirable because they can produce excessive wear to the driveline components and reduced comfort.

The engine torque is represented in Fig. 7, where one can see that the constraint on the upper bound of the control signal is satisfied by all tested control strategies. Note that, although the PID controller does not enforce constraints on control command, its output was saturated in order to enforce the engine limitations, i.e., the torque limit T_e^{max}.

6. CONCLUSIONS

Different strategies for controlling the drivetrain oscillations and different methods to compensate the delays that appear in communication networks were reported in the literature, but none in which the drivetrain is a component of a networked control system. This paper proposed a novel method of modeling the CAN-induced delays for control purposes and a horizon-1 MPC strategy to minimize the drivetrain oscillations that can handle both the performance/physical constraints and the time-varying delays. A flexible Lyapunov function was employed to obtain a non-conservative stability guarantee for the developed horizon-1 MPC scheme. A simulation experiment designed in collaboration with Ford Research and Advanced Engineering, US, validated the proposed approach and indicated that the developed scheme has the potential to meet the required real-time control specifications.

ACKNOWLEDGEMENTS

The first author is supported by the Romanian research grant SICONA - 12100/2008. The second author is supported by the Netherlands, Veni grant 10230, awarded by STW and NWO.

REFERENCES

Baumann, J., Torkzadeh, D.D., Ramstein, A., Kiencke, U., and Schlegl, T. (2006). Model-based predictive anti-jerk control. *Control Engineering Practice*, 14, 259–266.

Berriri, M., Chevrel, P., and Lefebvre, D. (2008). Active damping of automotive powertrain oscillation by a partial torque compensator. *Control Engineering Practice*, 16, 874–883.

Caruntu, C.F., Balau, A.E., Lazar, M., van den Bosch, P.P.J., and Di Cairano, S. (2011). A predictive control solution for driveline oscillations damping. In *Hybrid Systems: Computation and Control*. Chicago, USA.

Di Cairano, S., Bemporad, A., Kolmanovsky, I.V., and Hrovat, D. (2007). Model predictive control of magnetically actuated mass spring dampers for automotive applications. *International Journal of Control*, 80, 1701–1716.

Di Cairano, S., Yanakiev, D., Bemporad, A., Kolmanovsky, I., and Hrovat, D. (2010). Model predictive powertrain control: an application to idle speed regulation. Lecture Notes in Control and Information Sciences. Springer.

Gielen, R.H. and Lazar, M. (2009). Stabilization of networked control systems via non-monotone control Lyapunov functions. In *48th IEEE Conference on Decision and Control*, 7942–7948. Shanghai, China.

Gielen, R.H., Olaru, S., Lazar, M., Heemels, W.P.M.H., van de Wouw, N., and Niculescu, S.I. (2010). On polytopic inclusions as a modeling framework for systems with time-varying delays. *Automatica*, 46, 615–619.

Klehmet, U., Herpel, T., Hielscher, K.S., and German, R. (2008). Delay bounds for CAN communication in automotive applications. In *14th GI/ITG Conference Measurement, Modelling and Evaluation of Computer and Communication Systems*. Dortmund, Germany.

Lazar, M. (2009). Flexible control Lyapunov functions. In *American Control Conference*, 102–107. St. Louis, MO, USA.

Lefebvre, D., Chevrel, P., and Richard, S. (2003). An H-infinity-based control design methodology dedicated to the active control of vehicle longitudinal oscillations. *IEEE Transactions on Control System Technology*, 11, 948–956.

Rostalski, P., Besselmann, T., Maric, M., Van Belsen, F., and Morari, M. (2007). A hybrid approach to modelling, control and state estimation of mechanical systems with backlash. *International Journal of Control*, 80, 1729–1740.

Saerens, B., Diehl, M., Swevers, J., and Van den Bulck, E. (2008). Model predictive control of automotive powertrains - first experimental results. In *47th IEEE conference on Decision and Control*, 5692–5697. Cancun, Mexico.

Stewart, P., Zavala, J.C., and Flemming, P.J. (2005). Automotive drive by wire controller design by multi-objectives techniques. *Control Engineering Practice*, 13, 257–264.

Van Der Heijden, A.C., Serrarens, F.A., Camlibel, M.K., and Nijmeijer, H. (2007). Hybrid optimal control of dry clutch engagement. *International Journal of Control*, 80, 1717–1728.