Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

A Switching System oriented Modeling and Control Strategy for Idle Speed Control of a Hybrid Powertrain ⋆

Micha S. Obergfell * **Steven X. Ding** * **Frank Wobbe** **
Christoph-Marian Goletz ** **Michael Folkers** ** **Heiko Rabba** **

* *Department for Automatic Control and Complex Systems (AKS) University of Duisburg-Essen, 47057 Duisburg, Germany, (e-mail: micha.obergfell@uni-due.de (corresponding author), steven.ding@uni-due.de).*
** *IAV GmbH Ingenieurgesellschaft Auto und Verkehr, Carnotstraße 1, 10587 Berlin, Germany, (e-mail: {frank.wobbe, christoph-marian.goletz, michael.folkers, heiko.rabba}@iav.de)*

Abstract: The market development of partially electrified powertrains in passenger cars motivates the re-consideration of the idle speed control problem. In this paper, a switching system model is first developed to unite the main discrete-event characteristics of the combustion engine and time-continuous characteristics of the electric motor. The presented model is classified as a discrete-time switching system model with linear subsystems.
Based on this description, we further perform a model-based controller design using the lifting technique. Although the optimality property of the controller is bound to the assumption of constant turning speed, it still provides several useful properties. These are the inherent control allocation between electric and combustion engine, the consideration of the discontinuous behavior, and the discrete-time description basis which is important for implementation in a common controller architecture.

Keywords: idle-speed control, hybrid vehicles, switching systems, digital control, sampling rates, lifting technique, model-based control.

1. INTRODUCTION

Current passenger car models for individual mobility are not only driven by internal combustion (IC) engines. A considerable portion uses fully or partially electrified powertrains since these concepts promise significant fuel saving potential if deployed in appropriate usage behavior. Regarding the application of Mild-Hybrid-Electric, Hybrid-Electric, and Plug-In-Electric Vehicles (MHEV, HEV, PHEV), the task of turning speed control in idle mode – which is known and investigated from IC engines since long times – re-arises under significant changes. Hrovat and Sun (1997) and Powell (1987) provide an overview on strategies to control stand-alone IC engines in idle-mode.

The over-actuation of the plant and the possibility to charge or discharge the battery in idle-mode result in two different tasks. One is to select the charge/discharge rate of the battery to optimally suit the expected upcoming driving cycle, as investigated e.g. in Sciarretta and Guzzella (2007) and the other one is to optimally coordinate the actuators in case of disturbances while considering their different dynamics. These tasks are specifically challenging in the idle-mode operation due to the low inertia.

All three hybrid power train configurations (MHEV, HEV, PHEV) contain a mechanical coupling between the IC engine and the electric motor which serves as motor and generator, as schematically illustrated in Fig. 1. The scope of this work does not cover hybrid power train concepts where this mechanical coupling is achieved via vehicle wheels and road surface.

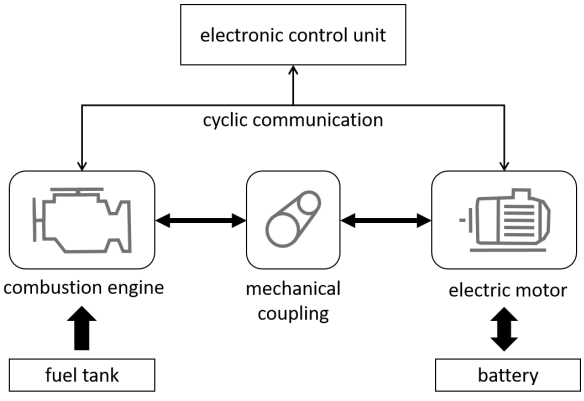

Fig. 1. Illustration of considered MHEV, HEV, and PHEV power train principles in idle mode.

With standard time-continuous or time-discrete descriptions, the discontinuous effects caused by the combustion process have to be approximated e.g. by using dead-time

⋆ The first and second author would like to thank the IAV GmbH for the long-term cooperation and financial support.

descriptions. Using these approximations, the control allocation problem in idle speed control can be handled by different techniques. Kandler (2014), Kandler et al. (2015) use a Model-Predictive and a frequency based control algorithm, assuming time-invariant system dynamics.

Approaches for idle speed control of different engines by using the crankshaft angle as independent variable and applying a transformation to continuous-time differential equations, can be found in Cook and Powell (1988), Kessel et al. (1998).

Four-stroke spark-ignited gasoline engines with four cylinders, mechanically coupled to an electric motor (motor/generator) are considered in this paper. In DeSantis et al. (2004) and DeSantis et al. (2006) a similar model was derived for a stand-alone IC engine, where the temporal difference between combustion segment begin and sampling instance is integrated to the controller design by investigating the calculation of maximum safe sets.

The objective of our work is to propose a modeling approach that considers the crankshaft-angle oriented behavior of the IC engine and the continuous-time dynamics in the framework of a switching system description (SSD). Based on this, a control scheme is proposed, in which the lifting technique is applied to model the dynamic behavior at distinct operating points. The model is derived in discrete-time and includes a description of system dynamics when sampling and engine rotation angle segments are asynchronous.

The remainder of this text is organized as follows: In section 2 the necessary assumptions are described and the model is derived. The lifting technique as well as the controller scheme are presented in section 3. In section 4 we present and evaluate simulation results. Some remarks towards practical implementation in section 5 conclude this paper.

2. MODELING

To model the rather complex combustion engine, Guzzella and Onder (2010) proposed the following approach: The process is split up in smaller sub-processes which are then described separately. The modeling results in gray-box models that are capable of describing the main physical effects in such a way that, with some experimental results, the model parameters can be identified. It turns out that some of the sub-processes (like the motion dynamics, air flow through valves, transport delays, communication delays) can be described based on (differential) equations of time-functions. Other effects like the discontinuous operation and the resulting delays like the *induction-to-power-stroke*-delay can be included in a crankshaft-angle oriented description.

For mean-value models (MVM) the time t is the independent variable and both input and output signals of the model are described as continuous signals. The reciprocating behavior of the engine is approximated by including several delays like the *induction-to-power-stroke*-delay in the model.

Discrete event models (DEM) adopt the crankshaft angle as independent variable. The discretization can accordingly be synchronized with the combustion cycle. The discretization period is the length of a so-called *segment* which is the angular shift in crankshaft angle between the cylinders combustion process. In this model some delays like the time from injection to torque center become constant since they are constant multiples of the segment length. We consider a 4-cylinder four-stroke engine where this segment length is $\varphi_{\Delta\mathrm{seg}} = \pi$.

2.1 The simplified mean-value-model

The model described in (1) and (2) represents the starting point of the following derivation. The application of this model is motivated by the fact that engine control systems consist of numerous actuators and control loops. We consider the generation of torque from the viewpoint of a supervisory control unit. This unit transmits the requested torque value to other control loops for e.g. inlet manifold pressure, throttle valve, ignition timing, spark advance, and others, that adjust the actuators respectively. We assume that the closed-loop behavior of these underlying control-loops is represented by the following transfer functions. The continuous differential equation for the turning speed of the crankshaft ω:

$$\dot{\omega}(t) = \frac{1}{\Theta}\underbrace{\left(T_\mathrm{c}(t) + T_\mathrm{i}(t)\right)}_{=T_\mathrm{ci}(t)} + \frac{1}{\Theta}T_\mathrm{e}(t), \tag{1}$$

contains the sum of the generated torques as well as the crankshaft inertia Θ. The torque generation through the input signals for combustion u_c, ignition u_i, and electric motor u_e,

$$T_\mathrm{c}(t) = u_\mathrm{c}(t-\tau_\mathrm{d}), T_\mathrm{i}(t) = u_\mathrm{i}(t), \dot{T}_\mathrm{e}(t) = {}^1\!/_{\tau_\mathrm{e}}\left(u_\mathrm{e}(t) - T_\mathrm{e}(t)\right), \tag{2}$$

are based on the following main assumptions:

- the combustion is treated as a continuous process,
- the dynamics to describe the air-mass and fuel injection as well as the *induction-to-power-stroke*-delay can be described by a lumped single delay parameter τ_d,
- the electric motor is modeled as a first order low-pass filter with time constant τ_e.

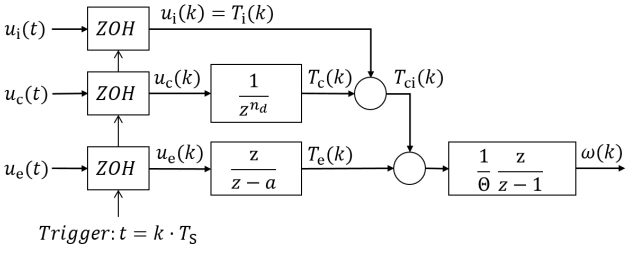

Fig. 2. Block diagram of simplified discrete-time MVM

Using the model from (1)-(2) the system description can be re-written as a discrete-time system using zero-order-hold (ZOH)-elements as shown in Fig. 2. Therefore, the time delay τ_d is increased to the next integer multiple of the sampling time $\tau_\mathrm{d} \approx n_\mathrm{d}\, T_\mathrm{S}$. This time-based discretization is consistent with state-of-the-art control systems, where sampling rates of 20ms are typical according to Guzzella and Onder (2010).

The derivation of the discrete-time state space matrices will be carried out in two stages: Firstly, we derive state space matrices in case the beginning of a power-stroke precisely coincides with the instant in time where the sampling takes place. This means that the crankshaft revolution time is *in-phase* and an integer multiple of the sampling time. Afterwards we calculate the state space matrices if the *in-phase*-assumption does not hold.

2.2 Derivation for in-phase-sampling

The difference between the *in-phase* and the *not-in-phase* situation is illustrated in Fig. 3. In the upcoming derivation we assume that the combustion torque T_{ci} is constant during one power-stroke. This behavior can be modeled as illustrated in Fig. 4. The added right-hand ZOH-element is triggered with the beginning of a power-stroke.

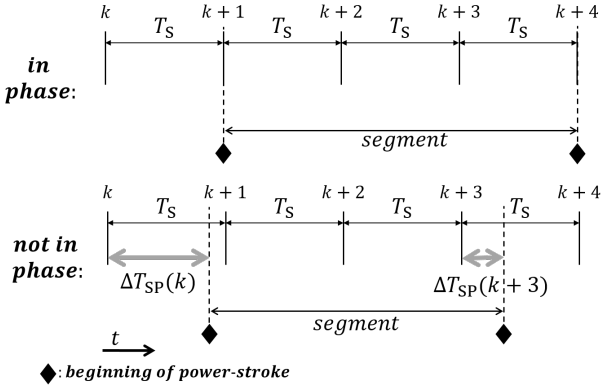

Fig. 3. Timing diagram of sampling and power-strokes for *in-phase* and *not-in-phase* situations

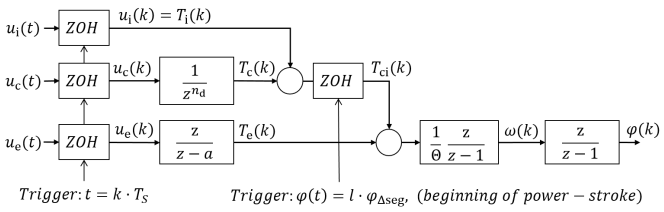

Fig. 4. Block diagram of discrete-time MVM with *in-phase* power-stroke. The torque T_{ci} is constant during each power-stroke.

In the following we omit the state $\varphi(k)$ which is considered as an input in (11), (13) and let $\tau_{\mathrm{d}} = 2 \cdot T_{\mathrm{S}}$ in order to keep the notation compact. Different values result in straightforward extensions of the matrices. We define the state and input vectors as:

$$\boldsymbol{x}(k) = \left[\omega(k),\, T_{\mathrm{e}}(k),\, T_{\mathrm{ci}}(k),\, T_{\mathrm{c}}(k+1)\right]^{\mathrm{T}}, \tag{3}$$

$$\boldsymbol{u}(k) = \left[u_{\mathrm{i}}(k),\, u_{\mathrm{c}}(k),\, u_{\mathrm{e}}(k)\right]^{\mathrm{T}}. \tag{4}$$

We denote those sampling instants where no segment begins together with the sampling interval as *case 0*. When there is a segment beginning with the sampling interval we use *case 1*.

Accordingly, we derive the state space matrices for *case 0* as:

$$\boldsymbol{A}_0 = \begin{bmatrix} 1 & a_{12} & \frac{T_{\mathrm{S}}}{\Theta} & 0 \\ 0 & a_{22} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \boldsymbol{B}_0 = \begin{bmatrix} 0 & 0 & b_{13} \\ 0 & 0 & b_{23} \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{5}$$

and the system matrices for *case 1* as:

$$\boldsymbol{A}_1 = \begin{bmatrix} 1 & a_{12} & 0 & \frac{T_{\mathrm{S}}}{\Theta} \\ 0 & a_{22} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \boldsymbol{B}_1 = \begin{bmatrix} \frac{T_{\mathrm{S}}}{\Theta} & 0 & b_{13} \\ 0 & 0 & b_{23} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{6}$$

Where the matrix elements $a_{12}, a_{22}, b_{13}, b_{23}$ depend on the electric motor time-constant τ_{e}, the crankshaft inertia Θ and the sampling time T_{S}. The relation

$$a_{12}, a_{22}, b_{13}, b_{23} > 0 \tag{7}$$

holds for all physically possible values. In both cases $q \in \{0, 1\}$, the remaining matrices are:

$$\boldsymbol{C}_q = [1\,0\,0\,0]\,, \boldsymbol{D}_q = [0\,0\,0]\,. \tag{8}$$

2.3 Derivation for not-in-phase-sampling

Since the sampling instants and the beginning of one segment do not necessarily coincide, as illustrated by Fig. 3, we show how the model can be enhanced to also describe the system behavior under realistic assumptions. Therefore we firstly find a way to calculate the time between the sampling instance and the subsequent segment change ΔT_{SP}. Then we show how to find appropriate system matrices $\boldsymbol{A}_1^{\dagger}(\Delta T_{\mathrm{SP}}), \boldsymbol{B}_1^{\dagger}(\Delta T_{\mathrm{SP}})$ to describe the system in this sampling instance correctly. Note that with this change from $\boldsymbol{A}_1, \boldsymbol{B}_1$ to $\boldsymbol{A}_1^{\dagger}(\Delta T_{\mathrm{SP}}), \boldsymbol{B}_1^{\dagger}(\Delta T_{\mathrm{SP}})$ the set of matrices to describe the system is no more finite. Still, the matrices $\boldsymbol{A}_0, \boldsymbol{B}_0$ and $\boldsymbol{C}_q, \boldsymbol{D}_q$ for $q \in \{0, 1\}$ stay the same. To calculate ΔT_{SP} we have to use the information of $\varphi(k)$ which is the crankshaft angle. This state is not included in the system model for controller layout since it has no influence on the controller performance.

We start from the continuous time description provided in (1) and (2), and calculate the elapsed angle in the timespan Δt as:

$$\Delta\varphi = \int_{\Delta t} \omega \, dt. \tag{9}$$

This leads to the following nonlinear function for the case that all input signals and the combustion-related torque-values T_{c} and T_{i} remain constant for the integration interval:

$$\Delta\varphi = f(\omega, \Delta t, T_{\mathrm{ci}}, T_{\mathrm{e}}, u_{\mathrm{e}}). \tag{10}$$

Using the Taylor-series approximation, this equation can be solved for Δt:

$$\Delta t = -\frac{\Theta\omega}{M_\Sigma} \pm \sqrt{\left(\frac{\Theta\omega}{M_\Sigma}\right)^2 + \Delta\varphi\frac{\Theta\omega}{M_\Sigma}} \tag{11}$$

using $M_\Sigma = T_{\mathrm{ci}} + u_{\mathrm{e}}$ as well as the first order approximation

$$(1 - e^{-x}) \approx x, \text{for } |x| << 1. \tag{12}$$

Remark: The estimation of Δt can be improved by using the second order approximation instead of (12), if needed.

We can now calculate the time between the beginning of a sampling interval and the next upcoming segment change for every sampling instance:

$$\Delta T_{\mathrm{SP}}(k) = \mathbf{f}(\boldsymbol{x}(k), \varphi(k), \boldsymbol{u}(k)). \tag{13}$$

If $\Delta T_{\mathrm{SP}}(k) < T_{\mathrm{S}}$ holds, we use $\Delta T_{\mathrm{SP}}(k)$ to calculate the state space matrices:

$$\boldsymbol{A}_1^{\dagger}(\Delta T_{\mathrm{SP}}) = \begin{bmatrix} 1 & a_{12} & \frac{\Delta T_{\mathrm{SP}}}{\Theta} & \frac{T_{\mathrm{S}}-\Delta T_{\mathrm{SP}}}{\Theta} \\ 0 & a_{22} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$
$$\boldsymbol{B}_1^{\dagger}(\Delta T_{\mathrm{SP}}) = \begin{bmatrix} \frac{T_{\mathrm{S}}-\Delta T_{\mathrm{SP}}}{\Theta} & 0 & b_{13} \\ 0 & 0 & b_{23} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}. \tag{14}$$

The set of system matrices $\{\boldsymbol{A}_1^{\dagger}(\Delta T_{\mathrm{SP}}), \boldsymbol{B}_1^{\dagger}(\Delta T_{\mathrm{SP}})\}$ can be reduced to a finite set by introducing a discretization of ΔT_{SP} with sufficient accuracy for practical applications.

2.4 Switching System description

Now we can introduce our model description in the context of system descriptions as given in Liberzon (2003), Lunze and Lamnabhi-Lagarrigue (2009). This leads to a classification as an internally forced switching (IFS) system, also called autonomously switched system. IFS means that the switching sequence $q(k)$ is not determined by any control signal directly but results from the system's state $\boldsymbol{x}(k)$ and therefore is indirectly influenced by $\boldsymbol{u}(k)$. The switching sequence:

$$q(k), \text{where } \forall k : q(k) \in \{0, 1, 2, ..., N_q\} \tag{15}$$

represents the activeness of one of the $N_q + 1$ subsystems. The SSD state equations for $\{\boldsymbol{x}(k), \boldsymbol{u}(k)\} \in \mathcal{C}_q$ are:

$$\begin{aligned} \boldsymbol{x}(k+1) &= \boldsymbol{A}_q \boldsymbol{x}(k) + \boldsymbol{B}_q \boldsymbol{u}(k) + d(k), \\ \boldsymbol{y}(k) &= \boldsymbol{C}_q \boldsymbol{x}(k), \end{aligned} \tag{16}$$

where $\mathcal{C}_q$ describes the region of state and input signals where the system matrices $\{\boldsymbol{A}_q, \boldsymbol{B}_q, \boldsymbol{C}_q\}$ are active. The required limitation to a finite number of N_q subsystems in (15) can be achieved by discretizing result ΔT_{SP} in (13).

2.5 Model simulation results

Simulations were taken with the switching sequence $s_{\mathrm{q}} : \{0, 0, 0, 1\}$ for the SSD-model which runs *in-phase*. They are compared to the MVM model shown in Fig. 2. For the simulations we introduced a proportional feedback gain to stabilize the plant using $u_{\mathrm{c}}(k)$. Simulations were taken out with two sinusoidal input signals $u_{\sin}(k)$ with different frequency. The resulting input signal is $u_{\mathrm{c}}(k) = -0.05\,\omega(k) + u_{\sin}(k)$, while $u_{\mathrm{i}}(k) = u_{\mathrm{e}}(k) = 0$. Simulation results in Fig. 5 show that the two different system models perform very similar within low frequency input signals.

The introduced time-delay in the low-frequency plot is caused by the sampling-like behavior. At higher frequencies not only the different frequencies but also the different amplitudes of the output signals become visible. This is caused by the sampling-like behavior of the SSD model. Here, the Nyquist-Shannon theorem, see Nyquist (1928), is not fulfilled.

3. LIFTING CONTROLLER DESIGN

We now propose a controller based on linear quadratic regulator (LQR) solution for the lifted system description, following the idea of Kranc (1957). Optimality is only achieved for those particular switching sequences and resulting turning speeds for which the lifting has been carried out. For simplicity the presented results here are based on those cases in which the *in-phase* assumption, described in section 2.2 holds.

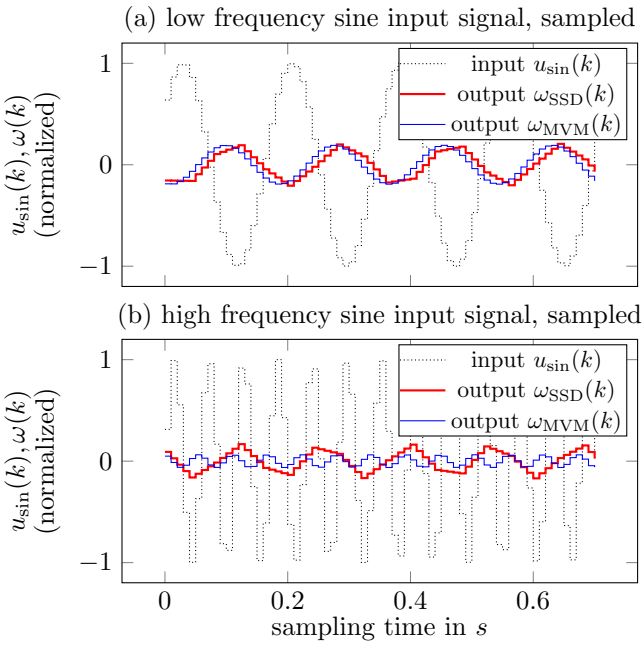

Fig. 5. Normalized external input and output signal for simplified MVM and SSD description at high and low input frequencies. Models are stabilized by proportional feedback. The plot shows signals after transient effects have vanished.

3.1 Lifting system description

The lifting technique can be used to model systems with time-varying state space-, input-, and output-matrices that follow a repetitive behavior after n_{L} timesteps, meaning $\forall k$:

$$\{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D}\}(k + n_{\mathrm{L}}) = \{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D}\}(k) \tag{17}$$

We define the new (lifted) input signal to be:

$$\boldsymbol{u}_{\mathrm{L}}(l) = \begin{bmatrix} \boldsymbol{u}(n_{\mathrm{L}} l + n_{\mathrm{L}} - 1) \\ \vdots \\ \boldsymbol{u}(n_{\mathrm{L}} l) \end{bmatrix}, \boldsymbol{y}_{\mathrm{L}}(l) = \begin{bmatrix} \boldsymbol{y}(n_{\mathrm{L}} l + n_{\mathrm{L}} - 1) \\ \vdots \\ \boldsymbol{y}(n_{\mathrm{L}} l) \end{bmatrix}. \tag{18}$$

The state signal is defined as:

$$\boldsymbol{x}_{\mathrm{L}}(l) = \boldsymbol{x}(l\, n_{\mathrm{L}}), \tag{19}$$

which leads to the new, time invariant state space description

$$\begin{aligned} \boldsymbol{x}_{\mathrm{L}}(l+1) &= \boldsymbol{A}_{\mathrm{L}}(l) \boldsymbol{x}_{\mathrm{L}}(l) + \boldsymbol{B}_{\mathrm{L}}(l) \boldsymbol{u}_{\mathrm{L}}(l), \\ \boldsymbol{y}_{\mathrm{L}}(l) &= \boldsymbol{C}_{\mathrm{L}}(l) \boldsymbol{x}_{\mathrm{L}}(l) + \boldsymbol{D}_{\mathrm{L}}(l) \boldsymbol{u}_{\mathrm{L}}(l). \end{aligned} \tag{20}$$

Definitions of the state space matrices $\{\boldsymbol{A}_{\mathrm{L}}, \boldsymbol{B}_{\mathrm{L}}, \boldsymbol{C}_{\mathrm{L}}, \boldsymbol{D}_{\mathrm{L}}\}$ result from equations (16), (18), (19) and are provided in Appendix A. The requirement on repetition of the state space matrices means that the turning speed of the engine has to behave repetitively after n_{L} sampling steps. This practically means that the turning speed has to stay

constant since significant turning speed oscillations within one segment are undesirable.

At this stage, we would like to point out, that for practically every constant idle speed $\overline{\omega}$, a lifting system description can be found by determining the common integer multiple of the sampling time T_S and the segment time $\varphi_{\Delta\mathrm{seg}}/\overline{\omega}$. The resulting switching sequences may unfortunately become quite large compared to the original system description. For the purpose of illustration in this paper we therefore consider those cases where the least common multiple function $lcm(\cdot,\cdot)$ of sampling time and segment time gives the segment time:

$$lcm\left(T_\mathrm{S}, \frac{\varphi_{\Delta\mathrm{seg}}}{\overline{\omega}}\right) = \frac{\varphi_{\Delta\mathrm{seg}}}{\overline{\omega}}, \tag{21}$$

only. There is an upper limit for the turning speed in this description, caused by the sampling time that may not become larger than the segment time. Assuming a sampling rate of $T_\mathrm{S} = 20$ms the maximum turning speed is 1500rpm while the range of interest in idle-speed control is typically 500..1000rpm, see Hrovat and Sun (1997).

3.2 Controller scheme

The control approach is based on the lifted system description. An output-based LQR controller in the lifted system environment is used with the cost function:

$$J_\mathrm{L}(l) = \sum_{l=1}^{\infty} \boldsymbol{y}_\mathrm{L}(l)^\mathrm{T}\boldsymbol{Q}_\mathrm{L}\boldsymbol{y}_\mathrm{L}(l) + \boldsymbol{u}_\mathrm{L}(l)^\mathrm{T}\boldsymbol{R}_\mathrm{L}\boldsymbol{u}_\mathrm{L}(l) \tag{22}$$

where the cost parameters are chosen to fulfill $\boldsymbol{Q}_\mathrm{L} > 0$ and $\boldsymbol{R}_\mathrm{L} > 0$. This leads to a state-feedback-controller using $\boldsymbol{u}_\mathrm{L} = \boldsymbol{K}_\mathrm{L}\boldsymbol{x}_\mathrm{L}$ which causes two challenges that have to be considered to enable a practical implementation.

Firstly, this controller requires an observer to calculate an estimate $\hat{\boldsymbol{x}}_\mathrm{L}(l)$ of the full state vector. This can be achieved for the lifted system with the standard Luenberger observer method. Secondly, the lifting technique in the form described in section 3.1 has to be extended to also include measurement information during the lifting-timespan. This is particularly important, if n_L is large. To achieve an optimal estimation of states and disturbances, the information contained in the measurements $y(l\,n_\mathrm{L}+1),...,y\left((l+1)\,n_\mathrm{L}-1\right)$, should be included in each sampling instant. To this end, we propose two strategies:

(a) the estimate of states and disturbances at time instant $(l\,n_\mathrm{L})$ is updated at time instants $(l\,n_\mathrm{L}+1)$,...,$((l+1)\,n_\mathrm{L}-1)$, or
(b) the underlying switching sequence is shifted for one position at each time instant e.g. $\{0,1,0,0\} \rightarrow \{1,0,0,0\} \rightarrow$ and so forth.

Both strategies require an implementation, similar to the one described in Alessandri and Coletta (2001).

3.3 Stability

Through the application of a linear quadratic controller to the lifted system description in (20), the stability analysis can be carried out using well-known tools of linear control theory. The weighting matrices $\boldsymbol{Q}_\mathrm{L}, \boldsymbol{R}_\mathrm{L}$ are chosen to be positive definite. Therefore an observer-based stabilizing state-feedback-controller for the lifted system-description can be found if the system is detectable and stabilizable.

4. RESULTS

To illustrate the features that arise through the shown model and controller, we present some simulation results. These simulations show that the dependencies of the system behavior on the crankshaft angle are incorporated to the actuator signal by the proposed modeling and controller layout technique.

For our simulations we used the weighting matrices $Q = diag(0.2,...,0.2)$ and $R_\mathrm{SSD} = diag\left(100,1,1,...,100,1,1\right)$, $R_\mathrm{MVM} = diag\left(\frac{100}{4},\frac{1}{4},1,...,\frac{100}{4},\frac{1}{4},1\right)$, as well as the parameters $\Theta = 0.15, n_\mathrm{d} = 3, \tau_\mathrm{e} = 0.07, \varphi_{\Delta\mathrm{seg}} = \pi$ and the sampling time $T_\mathrm{S} = 0.01$. Different values are chosen for R_SSD and R_MVM to cope with the alternating behavior of $u_\mathrm{i}, u_\mathrm{c}$ in Fig. 6. This behavior results directly from the model assumptions illustrated by the ZOH-element in Fig. 4. Regarding the signal u_i this behavior is consistent with the real application since the corresponding actuator, namely the ignition angle can only influence the produced torque when the beginning of a power-stroke is taking place in the upcoming sampling interval. For the signal u_c the assumption that the air mass flow into the cylinder is continuous, which is used for the derivation of the initial MVM, is bypassed by this model feature. For the implementation this behavior has to be corrected by either post-processing the controller output signals or augmenting the system model to include the accumulation of the air mass flow in the cylinder.

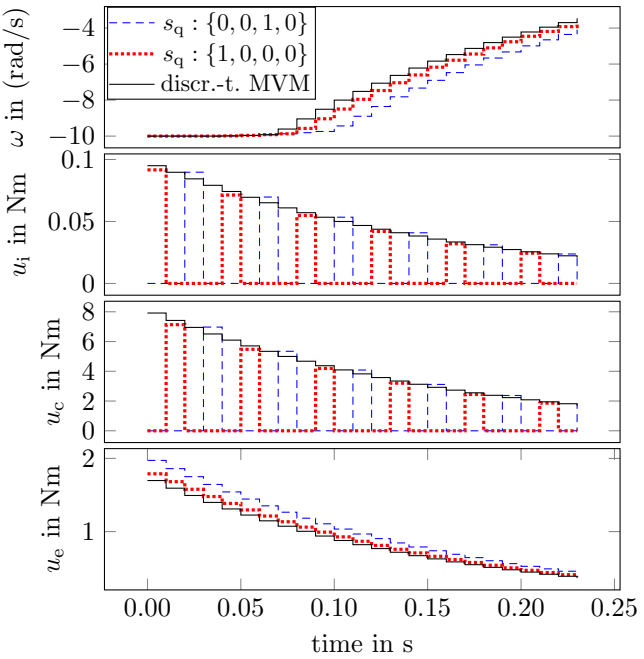

Fig. 6. SSD plant and controller output for different initial conditions in comparison with MVM

To sum up, the simulation results show that the system model tracks relevant properties which are traditionally covered either by the discrete-event or by the mean-value modeling approaches. Especially the discontinuous operation of the reciprocating engine is covered by the model under the previously described simplifications and assumptions. This emerges in the differences that appear in the actuator signals in Fig. 6. Here it can be seen, that

a different starting angle results in different shapes of the actuator signals. To put it simple the simulations show a behavior where the controller requests more torque from the electric motor whenever the combustion engine will need significantly longer to create the desired torque.

5. CONCLUSION

In this work we have presented an alternative solution to handle two challenging aspects regarding the control of state-of-technology hybrid powertrains in idle mode. These are firstly, the control allocation problem, which is to decide how to coordinate the electric motor and multiple control paths on the combustion engine. And secondly the system behavior consisting of continuous and discrete components, where typically digital controller structures with constant sampling time are available, only. We address these aspects by developing a switching system model in the discrete time environment. Thereby we provide the necessary system description to use the lifting technique for the determination of an optimal control law at constant turning speeds. We illustrate the previously described aspects by presenting selected simulation results.

The practical implementation and the adjustment of possible additional dynamical effects that improve the presented system description are mentioned and considered as the upcoming steps in the work of the authors. Measurements on a real system are considered necessary to incorporate practical aspects as e.g. the effects of (elastic) mechanical coupling, further actuator dynamics, signal transportation delays and others.

ACKNOWLEDGEMENTS

The authors would like to thank Jan Klöck as well as Alexander Wischnewski, Caroline Zhu and Changsheng Hua for revisions and discussions.

REFERENCES

Alessandri, A. and Coletta, P. (2001). Switching observers for continuous-time and discrete-time linear systems. In *Proceedings of the 2001 American Control Conference.*, volume 3, 2516–2521.

Cook, J.A. and Powell, B.K. (1988). Modeling of an internal combustion engine for control analysis. *IEEE Control Systems Magazine*, 8(4), 20–26.

DeSantis, E., DiBenedetto, M.D., and Berardi, L. (2004). Computation of maximal safe sets for switching systems. *IEEE Trans. on Autom. Control*, 49(2), 184–195.

DeSantis, E., DiBenedetto, M.D., and Pola, G. (2006). Digital idle speed control of automotive engines: A safety problem for hybrid systems. *Nonlinear Analysis: Theory, Methods & Applications*, 65(9), 1705 – 1724.

Guzzella, L. and Onder, C.H. (2010). *Introduction to Modeling and Control of Internal Combustion Engine Systems*. Springer, Berlin Heidelberg, 2nd edition.

Hrovat, D. and Sun, J. (1997). Models and control methodologies for ic engine idle speed control design. *Control Engineering Practice*, 5(8), 1093 – 1100.

Kandler, C., Koenings, T., Ding, S.X., Weinhold, N., and Schultalbers, M. (2015). Stability investigation of an idle speed control loop for a hybrid electric vehicle. *IEEE Trans. on Control Syst. Technol.*, 23(3), 1189–1196.

Kandler, C. (2014). *Methoden zur modellbasierten Mehrgrößenleerlaufregelung eines Hybridantriebsstrangs.* Berichte aus der Steuerungs- und Regelungstechnik. Shaker, Aachen.

Kessel, J.A., Schmidt, M., and Isermann, R. (1998). Modellbasierte motorsteuerung, -regelung und -überwachung. *MTZ - Motortechnische Zeitschrift*, 59(4), 240–246.

Kranc, G. (1957). Input-output analysis of multirate feedback systems. *IRE Transactions on Automatic Control*, 3(1), 21–28.

Liberzon, D. (2003). *Switching in systems and control*. Systems & control. Birkhäuser, Boston.

Lunze, J. and Lamnabhi-Lagarrigue, F. (2009). *Handbook of hybrid systems control: Theory, tools, applications*. Cambridge University Press, UK and New York.

Nyquist, H. (1928). Certain topics in telegraph transmission theory. *Transactions of the American Institute of Electrical Engineers*, 47(2), 617–644.

Powell, J. (1987). A review of ic engine models for control system design. *IFAC Proc. Volumes*, 20(5), 235 – 240.

Sciarretta, A. and Guzzella, L. (2007). Control of hybrid electric vehicles. *IEEE Control Systems Magazine*, 27(2), 60–70.

Appendix A. LIFTING MATRICES DEFINITION

In the following we show how to obtain the lifted system matrices used in section 3.1 where we use the capital pi notation for matrix multiplication as:

$$\prod_{i=1}^{n_{\text{end}}} \boldsymbol{X}_i = \boldsymbol{X}_1 \, \boldsymbol{X}_2 \, \cdots \, \boldsymbol{X}_{n_{\text{end}}}, \tag{A.1}$$

as well as the following abbreviations:

$$\mathbf{k}_l^{\dagger} = n_{\text{L}}\, l, \;\; \mathbf{k}_l^{\ddagger} = n_{\text{L}}(l{+}1) = \mathbf{k}_l^{\dagger} + n_{\text{L}} \tag{A.2}$$

We would like to point out that this design procedure does not necessarily result in a minimal realization of the system matrices. Therefore an elimination of unobservable and undetectable states might become necessary to check the detectability and stabilizability in section 3.3.

$$\boldsymbol{A}_{\text{L}}(l) = \prod_{m_{\text{A}}=1}^{n_{\text{L}}} \boldsymbol{A}(\mathbf{k}_l^{\ddagger} - m_{\text{A}}), \tag{A.3}$$

$$\boldsymbol{B}_{\text{L}}(l) = \left[\boldsymbol{B}(\mathbf{k}_l^{\ddagger}{-}1), \ldots, \prod_{m_{\text{B}}=1}^{n_{\text{L}}-1} \boldsymbol{A}(\mathbf{k}_l^{\ddagger} - m_{\text{B}})\, \boldsymbol{B}(\mathbf{k}_l^{\dagger})\right], \tag{A.4}$$

$$\boldsymbol{C}_{\text{L}}(l) = \begin{bmatrix} \boldsymbol{C}(\mathbf{k}_l^{\ddagger} - 1) \prod_{m_{\text{C}}=1}^{n_{\text{L}}-1} \boldsymbol{A}(\mathbf{k}_l^{\ddagger} - m_{\text{C}} - 1) \\ \vdots \\ \boldsymbol{C}(\mathbf{k}_l^{\dagger}) \end{bmatrix}, \tag{A.5}$$

$$\boldsymbol{D}_{\text{L}}(l) = \left[\boldsymbol{D}_{\text{L},1}^{\text{T}}, \boldsymbol{D}_{\text{L},2}^{\text{T}}, \cdots, \boldsymbol{D}_{\text{L},n_{\text{L}}}^{\text{T}}\right]^{\text{T}}, \tag{A.6}$$

where:

$$\begin{aligned} \boldsymbol{D}_{\text{L},1} &= \Big[\boldsymbol{D}(\mathbf{k}_l^{\ddagger} - 1), \boldsymbol{C}(\mathbf{k}_l^{\ddagger} - 1)\boldsymbol{B}(\mathbf{k}_l^{\ddagger} - 2), \cdots, \\ &\qquad \boldsymbol{C}(\mathbf{k}_l^{\ddagger} - 1) \prod_{m_{\text{D}}=1}^{n_{\text{L}}-2} \boldsymbol{A}(\mathbf{k}_l^{\ddagger} - m_{\text{D}})\, \boldsymbol{B}(\mathbf{k}_l^{\dagger})\Big], \\ &\vdots \\ \boldsymbol{D}_{\text{L},n_{\text{L}}-1} &= \left[\boldsymbol{O}, \cdots, \boldsymbol{O}, \; \boldsymbol{D}(\mathbf{k}_l^{\dagger} + 1), \; \boldsymbol{C}(\mathbf{k}_l^{\dagger} + 1)\boldsymbol{B}(\mathbf{k}_l^{\dagger})\right], \\ \boldsymbol{D}_{\text{L},n_{\text{L}}} &= \left[\boldsymbol{O}, \cdots, \boldsymbol{O}, \; \boldsymbol{D}(\mathbf{k}_l^{\dagger})\right]. \end{aligned} \tag{A.7}$$