An Efficient Nonlinear Model-Predictive Eco-Cruise Control for Electric Vehicles

Tim Schwickart, Holger Voos and Jean-Régis Hadji-Minaglou
University of Luxembourg
L-1359 Luxembourg, Luxembourg
Email: tim.schwickart@uni.lu

Mohamed Darouach
CRAN UMR-CNRS 7039
Université de Lorraine, IUT de Longwy
F-54400 Cosnes et Romain
Email: mohamed.darouach@univ-lorraine.fr

Abstract—**A nonlinear problem formulation of an energy-saving model-predictive eco-cruise controller for electric vehicles is presented. With regard to the intended application in real-world tests, the model has to include the specific properties of a serial electric vehicle such as energy-recovery and a discontinuous accelerator input giving rise to a binary control variable. These specific features and the nonlinearity of the system dynamics make it a challenging task to formulate the optimisation problem in a way that allows a fast computation in real-time application. The challenges are addressed by using a model of the vehicle dynamics that is formulated in terms of the vehicle position instead of time and by considering the kinetic energy instead of the velocity. Furthermore, various constraints on the input and state variables are introduced for a realistic representation of the vehicle characteristics. A special focus is put on the treatment of a binary input variable in the optimisation. Here, in order to avoid a mixed-integer formulation of the problem, a continuous variable is introduced which is forced to take only discrete values by a penalty term. Finally, first simulation results underline the feasibility of this control approach.**

I. Introduction

Besides the vehicle technology and the environmental conditions, the driving style has a huge influence on the energy consumption of a vehicle. For fuel-powered cars, this has been investigated in [1] and [2]. Due to possible improvements of 58 % (compared to a sportive driving style) [1] and the fact that improvements of this magnitude cannot be expected by improving the vehicle technology, it is a promising approach to improve the driving style in order to save energy.

A sophisticated way to address this problem is controlling the driving speed automatically by a driver-assistant system (eco-cruise control). Eco-cruise control can be described as an optimal control problem (see for example the preliminary work [3] and the up-to-date reference [4] for an explanation). The accelerator and brake pedal positions are the inputs of the system, while the driving speed and the energy consumption are the outputs given by a dynamic vehicle model (based on previous knowledge of the speed limits and the road slope). The inputs and outputs are optimised in a way that a cost function (containing for instance a compromise between driving speed and energy consumption) is minimised. As the car is running under changing traffic and environment conditions, it is hardly possible to calculate the complete optimal driving strategy in advance. A common approach to overcome this is to apply model-predictive control (MPC) in a receding horizon (RHC) fashion, where the optimisation is carried out for a finite prediction horizon and is repeated at every time step. This control strategy has been considered as the tool of choice for the eco-cruise control of fuel-powered cars in several works [4]–[9]. The biggest challenge in the application of MPC is that it requires a fast online-optimisation of the (in this case nonlinear) system behaviour, which is hampering a real-world implementation. Therefore, the formulation of the optimal control problem is decisively important in order to achieve a sufficiently fast solution.

In addition, almost all recent works dealing with eco-cruise control neglect electrical vehicles, where such a solution would also dramatically increase the driving range. The controllers developed so far are not fully suitable for electric vehicles due to their different specific technical properties and abilities. For instance, fuel-powered engines tend to be very inefficient at low partial load whereas electrical engines remain efficient in these operating points. Another example is that electrical vehicles can recover energy by electrical braking.

This paper contributes an efficient eco-cruise controller especially for electric vehicles including an underlying vehicle model, a cost function and constraints, which will be implemented in a serial electric car. Hence, all significant properties of this vehicle such as energy consumption and recovery, battery state-of-charge, actuator limits, a discontinuous control input as well as real-time requirements need to be considered.

The paper is organized as follows. In section II, the derivation of the underlying dynamic model is described. Section III explains the constraints and cost function terms that are used in the problem formulation including the treatment of a binary input variable. This is followed by a presentation of first simulation results in section IV.

II. Vehicle Model

The controller derived in this paper will be implemented in a Smart Electric Drive (model 2012) and hence its characteristics must be met by the problem formulation. The centrepiece of the Smart Electric Drive is a permanent magnet synchronous machine. This is a three-phase AC machine that is able to work as motor or generator. Thus, it is possible to feed back energy when decelerating. A lithium-ion battery serves as accumulator and supplies the synchronous machine via a DC/AC converter. The wheels are driven by the motor through a gear box with one fixed transmission ratio. This is possible as the characteristics of the electric machine do not require gear shifting.

In first experiments, only the accelerator shall be used as control input. The Smart recovers energy as soon as the accelerator pedal is released wherein the level of electrical braking cannot be influenced by the driver and is for simplicity reasons assumed to be a constant brake force. This leads to a binary control variable and will be discussed later on.

In the following, a model of the vehicle motion and its energy consumption in dependence on time is derived. Symbols can be found in Tab. I.

A one-dimensional vehicle motion based on Newton's second law is assumed in (1a), wherein the vehicle is considered as a mass point and tyre slip is disregarded. The vehicle acceleration is proportional to the difference between the traction force F_{trac} (or the brake force F_{brake}, respectively) and the driving resistance forces.

$$\frac{dv}{dt} = (F_{trac} + F_{brake} - F_r - F_{gr} - F_d)/m_{eq} \tag{1a}$$

The electrical power consumption is modeled as:

$$\frac{dE_{el}}{dt} = (F_{trac} + \eta_{rec} \cdot F_{brake}) \cdot (\alpha_1 \cdot v + \alpha_2 \cdot v^3) \tag{1b}$$

Polynomial (1b) is an approximation of the power consumption (if a traction force F_{trac} is acting) and the power recovery (if a brake force F_{brake} is acting) of the electric vehicle. The polynomial coefficients α are obtained by approximating a measured power consumption map.

The driving resistance forces are given as:

Rolling resistance force:

$$F_r \approx (m_v + m_l) \cdot g \cdot c_r \tag{2a}$$

Grade resistance force:

$$F_{gr} \approx (m_v + m_l) \cdot g \cdot \alpha_{sl} \tag{2b}$$

Air drag resistance force:

$$F_d = \frac{1}{2} \cdot \rho_a \cdot c_d \cdot A_v \cdot v^2 \tag{2c}$$

Equations (2) give the driving resistance forces as commonly modeled (see [10] for an overview). Note that trigonometric functions of the road slope angle α_{sl} have been replaced by their small-angle-approximations in (2a) and (2b).

For the application of a predictive cruise controller, it is useful to describe the model as a function of the position instead of time, since the inputs related to the road ahead (slope angle and speed limits) are given as functions of the position. The model can be reformulated in terms of position by multiplying the equations by the differential quotient dt/ds, which is just the inverse of the velocity v:

$$\frac{d}{dt} \cdot \frac{dt}{ds} = \frac{d}{dt} \cdot \frac{1}{v} = \frac{d}{ds} \tag{3}$$

The reformulation (3) consequently leads to a motion equation depending on the inverse of a state variable (the velocity v). This is disadvantageous for a fast and reliable solution of the optimisation problem. Following the idea in [9], a second reformulation is applied to calculate the kinetic energy e of the moving vehicle instead of the driving speed v. As the kinetic energy is a unique function of the velocity at a given vehicle mass, the speed can be calculated from the kinetic energy values after the optimisation.

$$\begin{gathered} e = \tfrac{1}{2} \cdot m_{eq} \cdot v^2 \\ \tfrac{dv}{dt} = v \cdot \tfrac{dv}{ds} = \tfrac{1}{2} \tfrac{d}{ds} v^2 = \tfrac{1}{m_{eq}} \cdot \tfrac{de}{ds} \end{gathered} \tag{4}$$

Reformulation (4) provides a simple relation between the derivative of the speed with respect to time and the derivative of the kinetic energy with respect to the position s. Applying (4) to the motion equation (1a) and applying (3) to the energy consumption expression (1b), the linear motion equation (5b) and the bilinear energy consumption equation (5c) are obtained.

Traction force:

$$\frac{dF_{trac}}{ds} = u_1 \tag{5a}$$

Kinetic energy of the moving vehicle:

$$\frac{de}{ds} = F_{trac} + F_{brake} - F_r - F_{gr} - F_d(e);\ e > 0 \tag{5b}$$

Where:

$$F_d(e) = \frac{1}{m_{eq}} \cdot \rho_a \cdot c_d \cdot A_v \cdot e$$

Electrical energy consumption:

$$\frac{dE_{el}}{ds} = (F_{trac} + \eta_{rec} \cdot F_{brake}) \cdot (\alpha_1 + \alpha_2 \cdot \frac{2}{m_{eq}} \cdot e) \tag{5c}$$

Equation (5a) provides the system input u_1 representing the change rate of the traction force. This differential equation is necessary to be able to limit the change rate of the traction force, see section III-A.

By introducing the assignment (6), the model can be given in the state space representation (7) with the states x, the control inputs u, the given input w and the parameters a.

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} F_{trac} \\ e \\ E_{el} \end{pmatrix};\quad \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} u_1 \\ F_{brake} \end{pmatrix};\quad (w_1) = (\alpha_{sl}) \tag{6}$$

$$\begin{aligned} \frac{dx_1}{ds} &= u_1 \\ \frac{dx_2}{ds} &= x_1 - a_1 \cdot x_2 + u_2 - a_2 \cdot w_1 - a_3 \\ \frac{dx_3}{ds} &= a_4 \cdot x_1 + a5 \cdot x_1 \cdot x_2 + a_6 \cdot u_2 + a7 \cdot u_2 \cdot x_2 \end{aligned} \tag{7}$$

III. Optimal Control Problem Formulation

Based on the model (5), the optimal control problem (OCP) of the eco-cruise controller is formulated.

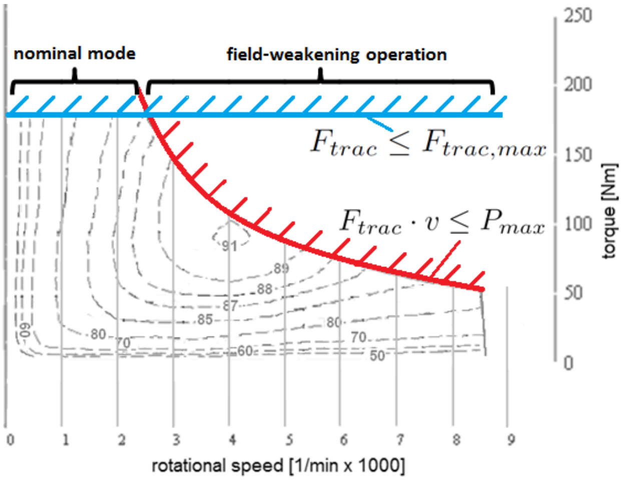

Fig. 1. Typical efficiency map of a permanent magnet synchronous machine together with the constraints guaranteeing a solution in the feasible area below the full-load curve.

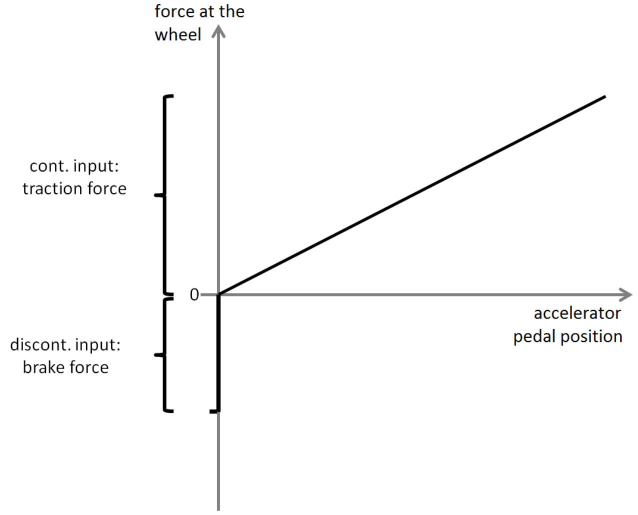
force at the wheel
cont. input: traction force
0
discont. input: brake force
accelerator pedal position

Fig. 2. Discontinuous control input: as soon as the accelerator pedal is released, the force at the wheels steps to a certain value.

A. Introduction of Constraints

To force the solution to provide realistic and realisable results, it is necessary to introduce constraints on the optimisation variables. To satisfy the actuator and vehicle limitations, the absolute values of the traction force F_{trac} and the brake force F_{brake} are constrained as well as the change rate r of the traction force:

$$\begin{aligned} 0 \leq F_{trac} &\leq F_{trac,max} \\ F_{brake,min} \leq F_{brake} &\leq 0 \\ r_{min} \leq u_1 &\leq r_{max} \end{aligned} \tag{8}$$

Limiting the traction force ensures that the solution of the optimisation problem will not include values above the maximum possible torque in nominal operating mode of the motor (Fig. 1, left). In addition, the solution must also be prevented from taking values above the speed-dependent maximum torque in field-weakening operation (Fig. 1, right). In this operating mode, the magnetic field and thus the torque is reduced in order to achieve a higher rotational speed. Here, the mechanical power output of the machine is kept constant. Consequently, an additional constraint limiting the maximum mechanical power P_{mech} is introduced. The two constraints together ensure that the solution of the optimisation problem only includes feasible traction force values.

However, the constraint on the motor power needs to be formulated such that it complies with the model (5). In terms of the kinetic energy, the following constraint is obtained:

$$P_{mech} = F_{trac} \cdot v = F_{trac} \cdot \sqrt{\frac{2}{m_{eq}} \cdot e} \leq P_{max} \tag{9}$$

Using a square root function is problematic regarding a fast and reliable computation of the solution. Thus, both sides of the inequality are squared in order to obtain a quadratic constraint. As the traction force F_{trac} and the kinetic energy e only take positive values, the reformulated constraint (10) provides the same results.

$$F_{trac}^2 \cdot \frac{2}{m_{eq}} \cdot e \leq P_{max}^2 \tag{10}$$

B. Handling of a Binary Control Variable

The introduction of a binary control variable is necessary to describe the characteristics of the accelerator pedal of the Smart Electric Drive. The Smart recovers electrical energy as soon as the accelerator pedal is released. This causes a brake force at the wheels during recuperation. The amount of this brake force cannot be influenced by the driver and is for simplicity reasons assumed to be constant. This results in a discontinuous control input: if the accelerator pedal is slightly pushed, a small traction force is generated but as soon as the pedal is released, a certain constant amount of brake force is acting, see Fig. 2. To address this, the input force is given in two parts: a continuous traction force F_{trac} that is constrained to be positive and a binary brake force F_{brake} that is constrained to take negative values, see (8).

With the brake force as a binary control variable, the optimal control problem becomes a Mixed-Integer Nonlinear Program (MINLP), which is in general computationally exhaustive to solve.

One approach to handle the problem is to solve it by a Dynamic Programming (DP) algorithm [11]. This method requires a discretisation of the inputs and states, thus a binary variable would only be a special case of discretisation. DP will find the solution to the MINLP if it exists. The biggest drawback of this method is the computation time, growing exponentially with the problem dimensionality. Another way to deal with integer variables are the so-called branching techniques [8]. The disadvantage here is again the computation time since these methods require to solve several nonlinear subproblems of the original problem. The computation time disqualifies the mentioned methods in most cases of fast real-time optimisation. A MINLP for eco-cruise control was first solved in real-time by Kirches in 2010 [8] by applying a Multiple Shooting discretisation and a rounding strategy.

In this paper, another approach is presented that is to the best of the knowlegde of the authors new in the application of eco-cruise control. Following the basic idea of [12], an additional cost function term is introduced that penalises any value of the brake force that differs from the two allowed binary numbers. Here, a quadratic penalty is used with zeros at the values 0 and $F_{brake,min}$. Together with the constraint (8), this cost function term will force the variable F_{brake} to take one of these two values.

$$p_{brake} = -F_{brake}^2 - F_{brake,min} \cdot F_{brake} \tag{11}$$

One advantage of this method is that efficient discretisation techniques in combination with efficient NLP solvers like Sequential Quadratic Programming (SQP) can be used. A further benefit is that only one NLP needs to be solved in contrast to the application of branching techniques. This leads to a high computational efficiency.

In practise, there are several difficulties in the application of the penalty method. On the one hand, if the weight on the related penalty term is large enough to make the binary optimisation variable take one of the values at the boundaries of the allowed interval, the problem can become ill-conditioned and the solver might not converge to the solution. On the other hand, if the penalty weighting is too low, violations of the discrete solutions occur and rounding becomes necessary.

Second, the penalty term changes the cost function such that several local minimum values can occur. Since most of the descent based solvers only converge to a local minimum, the global optimum might not be found. This means that a proper initialisation of the optimisation variables is important.

Despite of these practical problems, good results are achieved in this paper as long as the penalty weighting is tuned properly. It is worth mentioning that the penalty term (11) can also be introduced as a hard constraint. In this case, exact discrete solutions are guaranteed. However, the hard constraint is not used here, as it leads to convergence problems.

Finally, it must be prevented that traction force and brake force are simultaneously applied, since this is not possible in the real application. Introducing the following term as a hard constraint can solve this problem since its value will only be zero if at least one of the two input forces is zero.

$$-F_{trac} \cdot F_{brake} = 0 \tag{12}$$

However here, to achieve a better convergence and a faster computation, the left hand side of (12) is added to the cost function as a penalty, instead. This forces the solution to avoid the simultaneous use of both forces.

C. Cost Function

The next step is the set-up of the cost function. For the ease of implementation, a least-squares formulation (see (13a)) of the cost function is used. Hence, the absolute value of each term is minimised.

To consider the energy consumption of the vehicle, the expression $E_{el}(s) + E_{cap}(0)$ is added to the cost function. $E_{cap}(0)$ represents the free capacity of the battery at position s_0 of the prediction horizon. Thus, the controller tries to keep the state of charge as high as possible and considers the maximum possible charge.

The driving speed is taken into account by tracking a kinetic energy set-point. Hence, the least squares argument $e(s) - e_{ref}(s)$ is added to the cost function. No hard constraints regarding the kinetic energy are included because the problem might become infeasible if the vehicle exceeds the limits due to system disturbance.

Next, the traction force input is required to be smooth and thus oscillating solutions shall be suppressed. This is realised by adding the term u to the least squares objective. It also improves the convergence behaviour.

Finally, the quasi-binary behaviour of the brake force input is realised by adding (11) to the cost function. As already discussed, expression (12) is additionally introduced to avoid a simultaneous use of the traction force and the brake force.

The complete cost function is given in (13a). The weightings used for the calculations were obtained experimentally and can be found in Tab. I.

D. Overall Problem Formulation

The complete optimisation problem is given in (13). In this formulation, the eco-cruise control problem is a nonlinear problem with nonlinear constraints.

$$\begin{aligned} min \quad & \int_{s_0}^{s_{end}} (t_1 + t_2 + t_3 + t_4 + t_5) \cdot ds \\ & t_1 = Q_1 \cdot (E_{el}(s) + E_{cap}(0))^2 \\ & t_2 = Q_2 \cdot (e(s) - e_{ref}(s))^2 \\ & t_3 = Q_3 \cdot (u)^2 \\ & t_4 = Q_4 \cdot (-F_{brake}(s)^2 + F_{brake,min} \cdot F_{brake}(s))^2 \\ & t_5 = Q_5 \cdot (-F_{trac}(s) \cdot F_{brake}(s))^2 \end{aligned} \tag{13a}$$

subject to the model:

$$\begin{aligned} \frac{dF_{trac}}{ds} &= u \\ \frac{de}{ds} &= F_{trac} + F_{brake} - F_r - F_{gr} - F_d(e) \\ \frac{dE_{el}}{ds} &= (F_{trac} + \eta_{rec} \cdot F_{brake}) \cdot (\alpha_1 + \alpha_2 \cdot \frac{2}{m_{eq}} \cdot e) \end{aligned} \tag{13b}$$

subject to the equality constraints:

$$F_{trac}(0) = 0; \quad e(0) = 0; \quad E_{el}(0) = 0 \tag{13c}$$

subject to the inequality constraints:

$$\begin{aligned} 0 &\leq e \\ r_{min} \leq \frac{dF_{trac}}{ds} &\leq r_{max} \\ 0 \leq F_{trac} &\leq F_{trac,max} \\ F_{brake,min} \leq F_{brake} &\leq 0 \\ F_{trac}^2 \cdot (\frac{2}{m_{eq}} \cdot e) &\leq P_{max}^2 \end{aligned} \tag{13d}$$

TABLE I. SYMBOLS AND PARAMETERS

symbol	value	unit	description
A_v	2.17	m^2	projected front surface area
α_1	1.34	$-$	polynomial coefficient
α_2	3.87 e-5	$-$	polynomial coefficient
c_d	0.24	$-$	air drag coefficient
c_r	0.01	$-$	rolling resistance coefficient
η_{batt}	0.85	$-$	battery efficiency factor
η_{motor}	variable	$-$	motor and electronics efficiency
η_{mech}	0.92	$-$	transmission efficiency factor
η_{rec}	0.85	$-$	energy recovery efficiency factor
$F_{brake,min}$	-700	N	minimum brake force
$F_{trac,max}$	3613	N	maximum traction force
g	9.81	m/s^2	gravitational constant
m_{eq}	1197	kg	equivalent vehicle mass
m_l	75	kg	vehicle payload mass
m_v	1110	kg	empty vehicle mass
P_{max}	47	kW	maximum motor power output
Q_1	6.5e-3	$-$	weighting factor
Q_2	15	$-$	weighting factor
Q_3	1e7	$-$	weighting factor
Q_4	0.65	$-$	weighting factor
Q_5	0.15	$-$	weighting factor
r_{min}	-200	N/m	min. change rate of traction force
r_{max}	200	N/m	max. change rate of traction force
ρ_a	1.2	kg/m^3	density of the air

IV. COMPUTED RESULTS

In this section, the results obtained by solving the problem given in (13) are presented.

A distance horizon of 300 m is chosen and subdivided into 20 steps of 15 m as a compromise between computational speed and accuracy. The driving speed set-point and the initial speed are fixed to 90 km/h (375 kJ of kinetic energy, respectively). The road gradient profile includes a down-slope and is given in Fig. 3 (bottom).

The model parameters are chosen to represent a light electric vehicle in a realistic way, see Tab. I.

To solve the aforementioned problem, the MATLAB interface of the ACADO Toolkit [13] is used. ACADO is a software environment for solving dynamic optimisation and optimal control problems. It is chosen in this work since it provides efficient discretisation techniques (shooting methods) together with efficient solvers (sequential quadratic programming). Apart from the standard settings, a single-shooting discretisation and a Gauss-Newton type Hessian approximation are chosen.

The solution for a zero weighting of the energy consumption term is given in Fig. 3a (set-point tracking). It shows an overall energy consumption of 5.8 kJ for an average speed of 90 km/h. Fig. 3b shows the solution of the same problem with a non-zero weighting on the energy consumption (eco-cruise control). It gives an overall energy recovery of -24.6 kJ at an average speed of 88.4 km/h (a reduction of 1.8 %). According to this, a significant energy saving was achieved at a slightly reduced average speed, which is a remarkable result.

A closer look at Fig. 3a shows that the set-point-tracking controller starts out with a slight deceleration. The reason for this is the (within the actuator limits) inevitable speed overshoot during the down-slope. This shows the predictive behaviour of the controller. At the beginning of the slope, the controller starts braking. The traction force is put to zero and the brake force switches from -6 N to -700 N within one step, which illustrates the quasi-binary behaviour. After the down-slope, the velocity is regulated to the speed set-point of 90 km/h.
In contrast to the behaviour described before, the eco-cruise controller is trying to achieve a compromise between energy consumption and speed set-point following (Fig. 3b). It starts out with a zero traction force since this leads to a zero energy consumption according to (5c). Significant braking starts at the beginning of the down-slope and is stopped one step before the end of the down-slope in order to continue driving at a zero traction force. At the end of the calculated horizon, the traction force is increased to regulate the driving speed to a static compromise between set-point following and energy-saving of 87.7 km/h.

The simulation results show that the penalty method forces a quasi-binary behaviour of the brake force input. Nevertheless, rounding is necessary after solving because of small violations. However, the gain in computational speed is worth too accept this rather small inaccuracy.

All calculations were done in approximately 0.12 seconds on an Intel 2640M Quadcore running at 2.80 Ghz with 4 GB RAM. By directly executing the C-Code generated by ACADO using an appropriate initialisation of the optimisation variables, the calculation is expected to be fast enough for real-time applications (maximum calculation time 0.1 seconds).

V. CONCLUSION

An efficient problem formulation is presented that allows the solution of a nonlinear eco-cruise control problem for electric vehicles fast enough for a real-time application. The efficiency is based on a system model that is reformulated in dependence on kinetic energy (instead of speed) and the vehicle position. The model indirectly describes the vehicle speed by means of the kinetic energy, the vehicle position (it is the integration variable) and the energy consumption/recovery in two equations - a linear motion equation and a bilinear energy consumption equation.

Several linear and nonlinear constraints assure that practical limitations (e.g. the full-load curve) are satisfied.

The cost function itself consists of terms related to the compromise between energy savings and speed set-point tracking as well terms forcing an expected system behaviour. The penalty related to the energy consumption considers the free battery capacity. Hence, an explicit battery model is not needed. The kinetic energy set-point is predefined such that the speed limits are considered and no additional speed constraints are necessary.

Additionally, a binary control variable is introduced by penalising binary violations. By doing so, an efficient methodology to avoid a computationally extensive MINLP formulation is found.

First simulations using the ACADO Toolkit provide promising results. The next steps will be to set up a real-time model-predictive feedback loop solving the presented optimal control problem at every time step and to investigate the control behaviour. In this context, the convergence behaviour of the nonlinear optimisation needs to be examined as well.

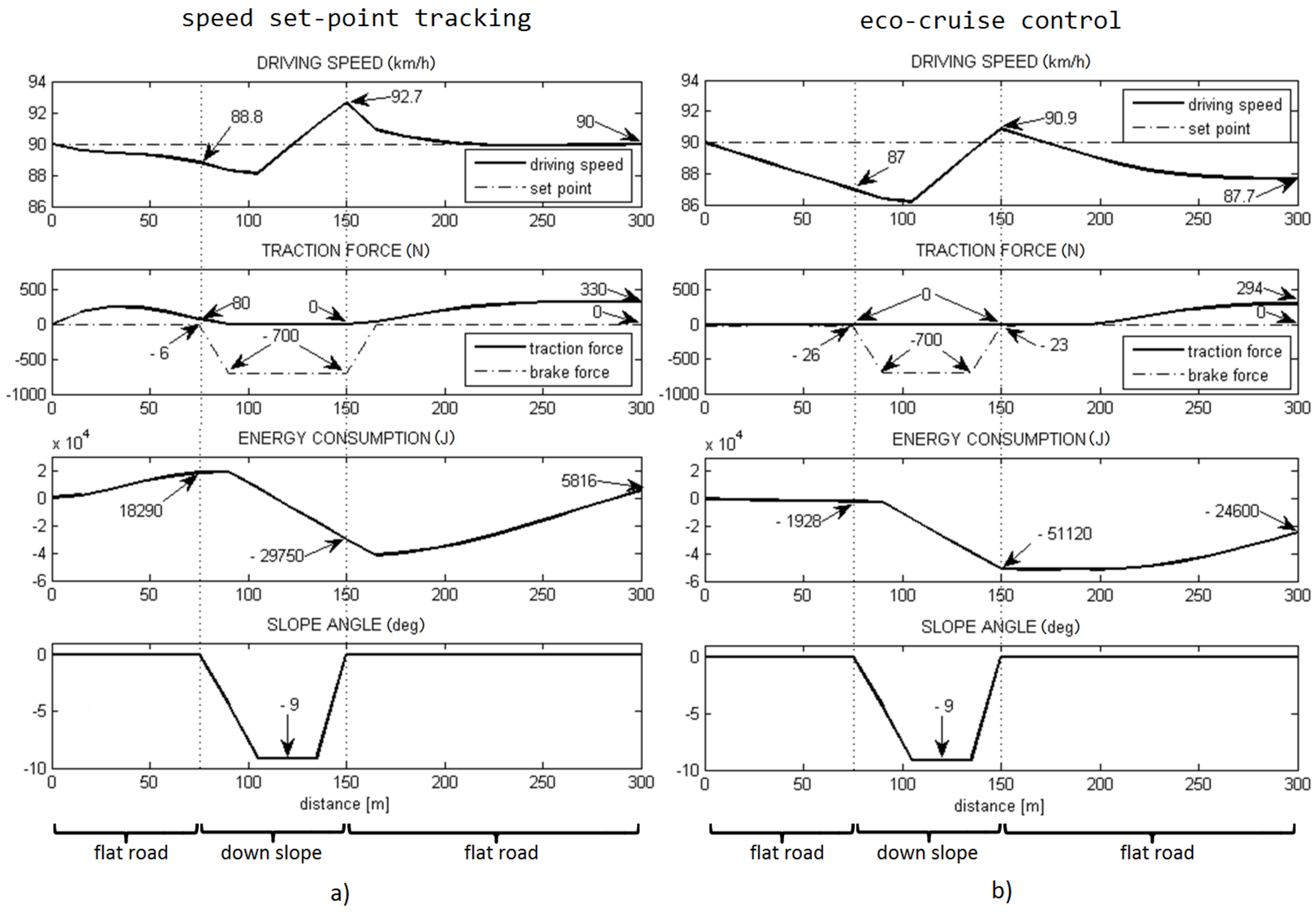

Fig. 3. Solution to the optimal control problem including a down-slope. a) gives the results for pure set point tracking. The results for eco-cruise control are depicted in b). Note that kinetic energy values are already converted into driving speed.

Finally, the control system will be tested in real-world scenarios with the Smart Electric Drive.

REFERENCES

[1] C. Dorrer, *Effizienzbestimmung von Fahrweisen und Fahrerassistenz zur Reduzierung des Kraftstoffverbrauchs unter Nutzung telematischer Informationen*. Expert-Verlag, 2004.

[2] J. N. Barkenbus, “Eco-driving: An overlooked climate change initiative,” *Energy Policy*, vol. 38, no. 2, pp. 762 – 769, 2010.

[3] A. Schwarzkopf and R. Leipnik, “Control of highway vehicles for minimum fuel consumption over varying terrain,” *Transportation Research*, vol. 11, no. 4, pp. 279 – 286, 1977.

[4] B. Saerens, “Optimal control based eco-driving - theoretical approach and practical applications,” Ph.D. dissertation, KU Leuven, 2012.

[5] W. Huang, D. Bevly, S. Schnick, and X. Li, “Using 3d road geometry to optimize heavy truck fuel efficiency,” in *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*, oct. 2008, pp. 334 –339.

[6] F. Lattemann, K. Neiss, S. Terwen, , and T. Connolly, “The predictive cruise control a system to reduce fuel consumption of heavy duty trucks,” SAE, Tech. Rep. 2004-01-2616, 2004.

[7] M. Kamal, M. Mukai, J. Murata, and T. Kawabe, “Ecological vehicle control on roads with up-down slopes,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, no. 3, pp. 783 –794, sept. 2011.

[8] C. Kirches, “Fast numerical methods for mixed-integer nonlinear model-predicitve control,” Ph.D. dissertation, Ruprecht-Karls-Universität Heidelberg, 2010.

[9] E. Hellstroem, “Look-ahead control of heavy vehicles,” Ph.D. dissertation, Linköping University, 2010.

[10] B. Heissling and M. Ersoy, *Chassis Handbook*. Vieweg + Teubner Verlag, 2011.

[11] R. E. Bellmann and S. E. Dreyfus, *Applied Dynamic Programming*. Princeton University Press, 1962.

[12] J.-F. Fu, R. G. Fenton, and W. L. Cleghorn, “A mixed integer-discrete-continuous programming method and its application to engineering design optimization,” *Engineering Optimization*, vol. 17, no. 4, pp. 263–280, 1991.

[13] B. Houska, H. J. Ferreau, and M. Diehl, “Acado toolkit - an open-source framework for automatic control and dynamic optimization,” *Optimal Control Applications and Methods*, vol. 32, no. 3, pp. 298–312, 2011.