9th International Scientific – Technical Conference – PROCESS CONTROL 2010
June 7 – 10, 2010, Kouty nad Desnou, Czech Republic

DESIGN OF A MPC CONTROLLER RUNNING ON DSPACE HARDWARE USING ACADO TOOLKIT

KOPACKA MATUS, SAERENS BART†, FERREAU HANS JOACHIM‡, HOUSKA BORIS‡, DIEHL MORITZ‡, ROHAL-ILKIV BORIS

Slovak University of Technology in Bratislava, Faculty of Mechanical Engineering
Institute of Automation, Measuring and Applied informatics
Namestie slobody 17, 812 31 Bratislava, Slovakia
tel: +421 2 5249 5392, e-mail: matus.kopacka@stuba.sk
† Katholieke Universiteit Leuven, Department of Mechanical Engineering
Celestijnenlaan 300A, 3001 Leuven-Heverlee, Belgium
tel: +32 16 322548, e-mail: bart.saerens@mech.kuleuven.be
‡Katholieke Universiteit Leuven, Electrical Engineering Department (ESAT-SCD)
Kasteelpark Arenberg 10, 3001 Leuven-Heverlee, Belgium
tel: +32 16 321884, e-mail: moritz.diehl@esat.kuleuven.be.

Abstract: The following paper describes the design of a predictive controller using the ACADO toolkit (an open-source framework for automatic control and dynamic optimization) dedicated for a real-time run on dSpace hardware. The hereinbefore mentioned newly programmed toolkit offers a controller build-up with a user friendly and intuitive C++ coding environment, computational inexpensiveness and real-time operation. Velocity control of a personal car has been chosen as a case study. This paper presents the necessary steps of a controller build-up, including its coding and simulation.

Keywords: MPC, ACADO, optimal control, vehicle dynamics, dSpace

1 INTRODUCTION

Control of vehicle dynamics is a complex task, considering different phenomena acting onto the vehicle arising from dynamic nature of car parts; or outer disturbances caused by the road state, or weather conditions. To maintain the vehicle in a prescribed trajectory or velocity, the controller has to have access to the engine and brake management. Another condition for proper functionality is a fast controller able to react on a millisecond level, making it a typical example of a system with fast dynamics. Such a controller then has to be equipped with a solver, able to solve a minimization problem within that short control period, as well.

The software package ACADO Toolkit is used in this work. This package is developed for the model predictive control (MPC) design, disposing of the above mentioned features and intuitive coding.

2 VELOCITY CONTROL

2.1 Motivation

Velocity control and its derivatives (torque control, revolutions control) has a variety of applications, e.g. in electronic cruise control, in maintaining the constant idle speed of an engine, in eco-drive assist systems (Saerens et al. [2010]) or in holding a constant power output of a cogeneration unit.

Cruise control maintains a desired reference velocity, so that the driver can relax his foot from the gas pedal, when driving long distances on a sparsely populated highway; or to avoid speed limit violations. Engagement of a cruise control also results in a lower fuel consumption on non-hilly tracks. Disadvantages might be a delayed deceleration of the vehicle, when needed or an improper velocity regarding the traffic conditions or the road state.

Another field, where the tracking of a velocity reference is desired, is fuel consumption testing. There exist normalized test cycles imposed by international authorities that are obligatory for all car manufacturers. They basically consist of predefined time-points, at which the vehicle has to have a particular prescribed velocity (Fig. 1). If the tracking of the mentioned reference velocity is not accurate, the official fuel consumption and emissions are misrepresented. This is a motivation to use electronic velocity control, as well.

There exist many test cycles, different in the area of application, country of validity, etc., such as:

EUDC	- Extra Urban Driving Cycle, simulates the highway,
ECE	- steady state cycle for heavy duty truck engines,
NEDC	- New European Driving Cycle, since 2000; NEDC is ECE plus EUDC,
15 MODE	- urban driving cycle, Japan,
FTP 75	- a transient test cycle for cars and light duty trucks, US.

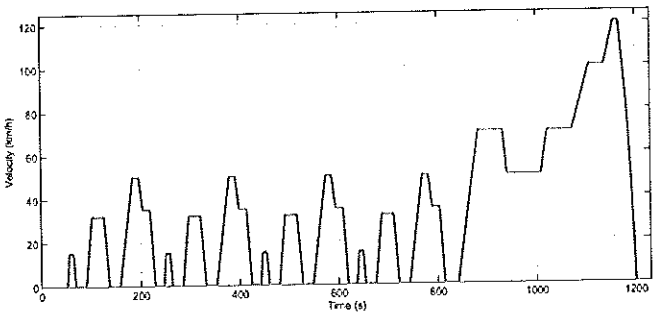


Figure 1 – NEDC test cycle

A detailed list of test cycles can be found at (tes [2010]).

2.2 System Description

The task in this case study is to track a predefined velocity profile. The vehicle dynamics are simplified, only taking into account the longitudinal dynamics and considering the vehicle as a point mass. The considered vehicle system consists of three, mutually coupled subsystems:

- The vehicle longitudinal dynamics (Hellström [2005]), including wind resistance, road slope, inertia effects of wheels and engine, transmission ratios and efficiency of a gearbox and the final drive, etc.:

$$\begin{aligned}\dot{v} = & \frac{r_\mathrm{w}}{J_\mathrm{w} + m r_\mathrm{w}^2 + \eta_\mathrm{D} i_\mathrm{D}^2 \eta_\mathrm{G} i_\mathrm{G}^2 J_\mathrm{E}} \Big(\eta_\mathrm{D} i_\mathrm{D} \eta_\mathrm{G} i_\mathrm{G} T_\mathrm{e} - k_\mathrm{b} B_\mathrm{b} - \frac{1}{2} c_\mathrm{w} A_\mathrm{a} \rho_\mathrm{air} r_\mathrm{w} v^2 \\ & - m g r_\mathrm{w} (c_\mathrm{r} \cos\theta + \sin\theta) \Big).\end{aligned} \tag{1}$$

- The dynamics of the air in the intake manifold of the engine (Saerens et al. [2009]), considering sonic choking:

$$\dot{p}_\mathrm{m} = \frac{r T_\mathrm{m}}{V_\mathrm{m}} \frac{p_\mathrm{a}}{\sqrt{r T_\mathrm{a}}} A_\mathrm{TV}(\alpha) c_\mathrm{d} f\left(\frac{p_\mathrm{m}}{p_\mathrm{a}}\right) - \frac{\eta_\mathrm{v} V_\mathrm{tot}}{2 V_\mathrm{m}} n p_\mathrm{m}. \tag{2}$$

- The throttle valve, throttling the air sucked into the engine:

$$\dot{x} = -43.55x + u, \tag{3a}$$
$$\alpha = 27.06x + 0.38u. \tag{3b}$$

Where the symbols' meanings are:

α	throttle valve plate angle	J_w	wheel inertia
η_D	efficiency of final drive (differential)	k_b	weighting brake parameter
η_G	efficiency of gearbox	m	mass of vehicle
η_v	volumetric efficiency	$\dot{m}_f$	fuel mass flow rate
κ	isentropic exponent	n	engine speed
ρ_{air}	air density	p_a	ambient pressure
θ	road slope	p_m	intake manifold pressure
A	frontal cross-sectional area	r	specific gas constant
B_b	maximal brake force	r_w	wheel radius
c_d	throttle body discharge coefficient	T_a	ambient air temperature
c_r	rolling friction coefficient	T_e	engine torque
c_w	aerodynamic drag coefficient	T_m	air temperature in the intake manifold
f	f. taking into account sonic choking	u	control input
g	earth gravitational constant	v	velocity
i_D	reduction ratio of final drive (differential)	V_{tot}	total cylinder volume
i_G	reduction ratio of the gearbox	V_m	intake manifold volume
J_e	engine inertia	x	identified system state

The input into the entire system is a control input u (voltage), scaled to $0 \leq u \leq 1$, forcing the throttle valve to change its position between $0 \leq \alpha \leq \frac{\pi}{2}$ rad and so to let the air into the engine. The output of the system is the vehicle velocity v. The cost function for the optimal control $\mathcal{J}$ itself is of a simple form:

$$\mathcal{J} = \|v - v_{ref}(t)\|_2^2, \tag{4}$$

where $v_{ref}(t)$ is the reference velocity of the vehicle, in the simulation set to a constant value.

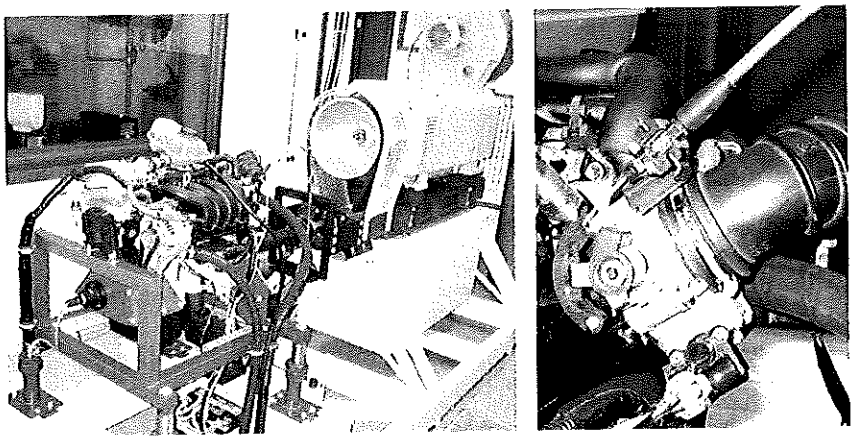
Due to the technical restrictions, the equations of air intake dynamics and the throttle valve behavior are describing the real systems present in the laboratory, while the vehicle dynamics are simulated by an electric brake connected to the engine (Fig. 2).

3 ACADO

3.1 Software description

The *ACADO Toolkit* (aca [2010]) is an open source software environment and algorithm collection for automatic control and dynamic optimization. It provides a general framework for employing a great variety of algorithms for:

- direct optimal control,
- model predictive control,
- state and parameter estimation,
- multi-objective optimization.



(a) Combustion engine connected to an electrical brake (b) Detail of the throttle valve body

Figure 2 – Electric motor connected to a throttle valve

It is implemented as a self-contained C++ code and comes along with user-friendly Matlab interfaces. The object-oriented design allows for convenient coupling of existing optimization packages and for an extension with user-written optimization routines. Table 1 the solvers that can be used in ACACO.
It has been chosen for its speed crucial for the control of fast dynamic systems and other features listed above.

Table 1 – Available optimization solvers

Application area	Software
Linear Programming	ILOG CPLEX, SoPlex, lp_solve, LINGO, linprog (MATLAB)
Quadratic Programming	MOSEC, quadprog (MATLAB), qpOASES
Mixed-Integer Linear Programming	ILOG CPLEX, lp_solve
Mixed-Integer Quadratic Programming	TOMLAB (MATLAB)

3.2 Source code of a controller

The code of the designed controller will be based on all three differential equations and state representation of the vehicle, as described in subsection 2.2. The explanation of the code structure will be done considering an extra MPC controller, only taking into account the throttle valve subsystem (Eq. 3).
Please note that the structure of a controller for the full system is the same and follows the following logical parts:

1. Definition of differential states, variables, control inputs, differential equations, constants, ...
2. Definition of a cost function in the form:

$$\mathcal{J} = Q\|h - s\|_2^2, \tag{5}$$

where:
Q is a weighting matrix,
h is a vector of states,
s is a vector of setpoints.

3. Definition of an optimal control problem with corresponding algorithm, constraints, initial values, settings, ...

4. Plotting of results.

5. MPC initialization, if the code is ready to be compiled.

— Start of a sample code —

```
#include <acado_toolkit.hpp>
#include <gnuplot/acado2gnuplot.hpp>

int main( ){

USING_NAMESPACE_ACADO

DifferentialState       x;      // initial definition of differential states,
IntermediateState       alpha;  // variables,
Control                 u;      // control inputs,
DifferentialEquation    f;      // differential equations, constants...

Matrix A(1,1);
A.setZero();

Matrix B(1,1);
B.setZero();

Matrix C(1,1);
C.setZero();

Matrix D(1,1);
D.setZero();

double t_start  = 0.0;
double t_end    = 1.0;
double alphaD   = 1.0;          // alpha Desired - static reference

A(0,0)  = -43.5489;
B(0,0)  =   1.0;
C(0,0)  =  27.0625;
D(0,0)  =   0.3775;

f << dot(x) == A(0,0)*x +   b(0)*u;
     alpha  =  C(0,0)*x + D(0,0)*u;

Function h;                     // h,Q,r are arguments of a cost function
h << alpha;
h << u;

Matrix Q(2,2);
Q.setIdentity();
Q(0,0) = 10.0;
```

```
Q(1,1) = 0.01;

Vector r(2);
r(0) = alphaD;
r(1) = 0.0;

OCP ocp( t_start, t_end, 10 );  // definition of Optimal Control Problem (OCP)
ocp.minimizeLSQ( Q,h,r );
ocp.subjectTo( f );

ocp.subjectTo( 0.0 <=   u   <= 1.0  );       // constraints
ocp.subjectTo( 0.0 <= alpha <= 1.0  );
//ocp.subjectTo( AT_START, u==0       );

GnuplotWindow window;                        // results plotting
window.addSubplot( x,          "State x" );
window.addSubplot( alpha, "Output alpha" );
window.addSubplot( u,        "Control u" );

RealTimeAlgorithm algorithm(ocp,0.05);       // OCP algorithm and settings
algorithm << window;
algorithm.set( "MaxNumIterations",3 );
algorithm.set( "HessianApproximation", GAUSS_NEWTON );

Vector x0(1);                                //initial conditions of states
x0(0) = 0.0;

algorithm.solve( x0 );

                      // MPC controller, uncommented during compilation
// Controller controller;
// controller.setControlLaw( algorithm );
// controller.init( x0 );
// controller.step( 0.0,x0 );
return 0;
}
```

— End of code —

4 SIMULATION RESULTS

The simulation has been done using the ACADO toolkit. The goal is to accelerate a vehicle from 4 m/s to 10 m/s, while respecting the physical constraints of the system (Fig. 3).

It shows how the system states and quantities are changing, particularly the control input, desired velocity, engine revolutions, instantaneous fuel and air inflow, etc.

5 CONCLUSION

The paper introduced a field of vehicle longitudinal dynamics modeling and velocity control using a MPC (model predictive control) controller by using the ACADO toolkit as a solver for the minimization problems within the controller.

A future augmentation of this setup shall be a real-time run on a dSpace system, where the electric brake connected to the engine shall simulate the dynamics of a vehicle. Another improvement could be an additional utilization of the brake force described in this case by a constant member $k_b B_b$ in (Eq. 1) and a decentralized multi loop control of the whole plant.

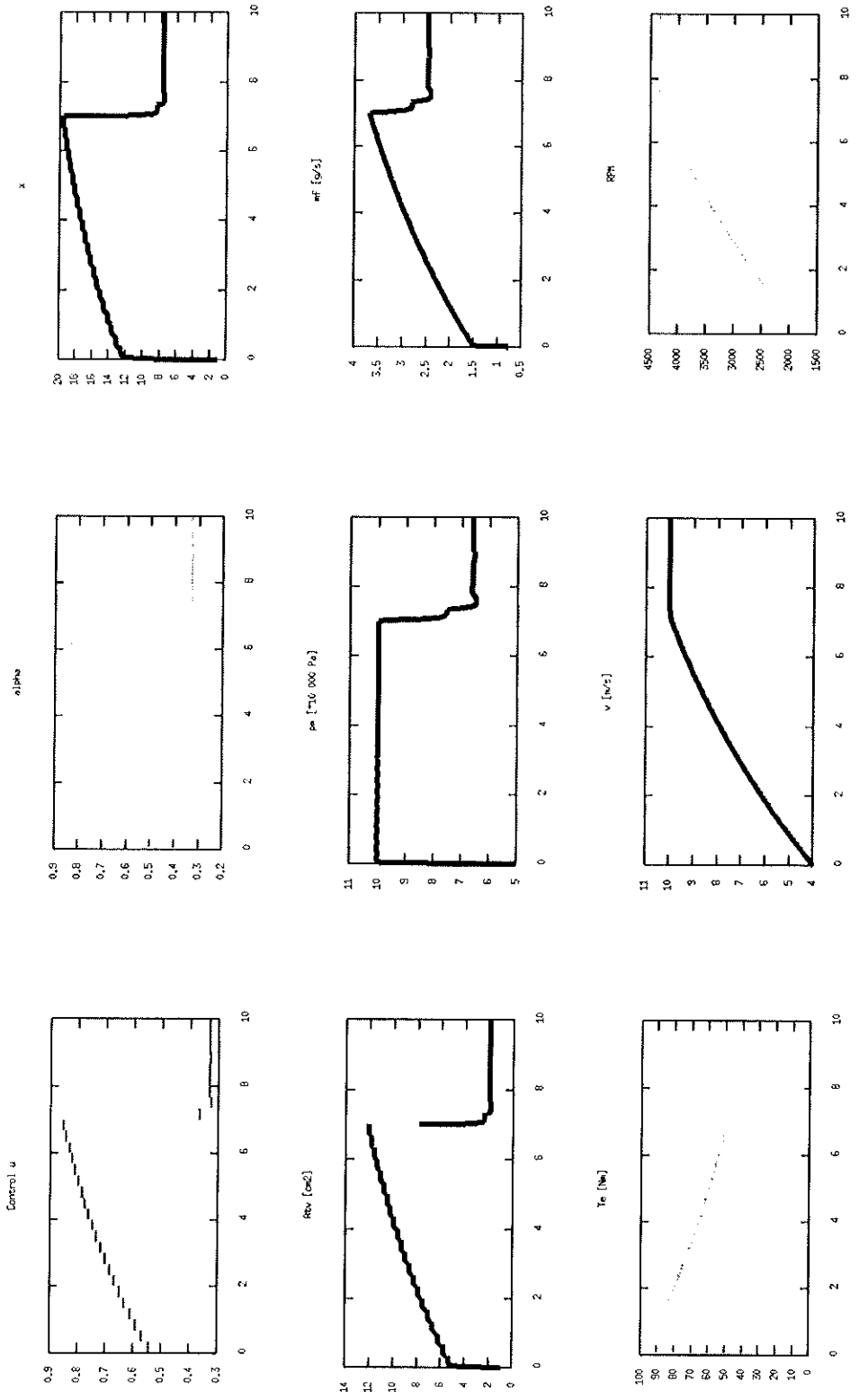


Figure 3 – ACADO simulation of a vehicle MPC (x-axis: Time (s))

ACKNOWLEDGEMENT

I would like to thank to Lieboud Van den Broeck from KUL, who helped me with the identification issues at the engine.

The work has been supported by the Slovak Research and Development Agency under grant APVV-0280-06 and LPP-0096-07. This support is very gratefully acknowledged.

This research has been supported by KUL: Optimization in Engineering Center (OPTEC), IOF-SCORES4CHEM, GOA MaNet, GOA Real-time global optimal control of robots and mechatronic systems; by IUAP P6/04 (DYSCO-Dynamical systems, control and optimization, 2007-2011), by FWO via projects and fellowships (the third author holds an FWO scholarship) and via research communities; the IWT via PhD and SBO grants (LeCoPro,Climaqs,POM); ACCM; the EU via ERNSI; FP7-HD-MPC (INFSO-ICT-223854), COST intelliCIS, FP7-EMBOCON (ICT-248940).

Research funded by a Ph.D grant of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen).

References

2010. Emission test cycles. `http://www.dieselnet.com/standards/cycles`.

2010. Key features of acado. `http://www.acadotoolkit.org`.

HELLSTRÖM, E. 2005. *Explicit use of road topography for model predictive cruise control in heavy trucks*. Master's thesis, Dept. of Electrical Engineering, Linköpings universitet.

HOUSKA, B.; FERREAU, H. J.; DIEHL, M. 2010. Acado toolkit - an open-source framework for automatic control and dynamic optimization. *Optimal Control Methods and Application*. (accepted).

LJUNG, L. 1999. *System identification. Theory for the user*. Prentice Hall PTR.

SAERENS, B.; DIEHL, M.; VAN DEN BULCK, E. 2010. *Automotive Model Predicitive Control*, volume 402/2010 of *Lecture Notes in Control and Information Sciences*, chapter Optimal Control Using Pontryagin's Maximum Principle and Dynamic Programming, 119 – 138. Berlin / Heidelberg: Springer. doi:10.1007/978-1-84996-071-7_8.

SAERENS, B.; VANDERSTEEN, J.; PERSOONS, T.; SWEVERS, J.; DIEHL, M.; VAN DEN BULCK, E. 2009. Minimization of the fuel consumption of a gasoline engine using dynamic optimization. *Applied Energy*, 86, 9.