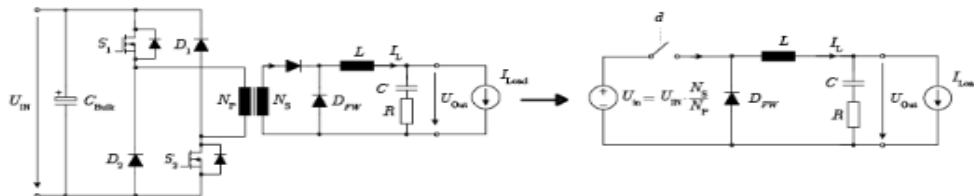


EXAMPLE: Power Electronics Control – PLECS® by PLEXIM

Introduction - Control of a DC/DC Converter

An important field of application for model predictive control are power electronic systems. In this example a typical DC/DC converter which supplies an isolated DC voltage to a telecom system is considered. Assume that the input voltage of the two-transistor forward converter, depicted below on the left, is a constant voltage U_{IN} delivered by a previous PFC rectifier stage. The load attached to the converter has an ohmic-capacitive characteristic.

This two-transistor forward converter can be modelled as a buck converter from which it is more convenient to derive a control oriented model. The buck converter has only one switch and the input voltage U_{in} is the actual input voltage scaled by the transformer turn ratio. The equivalent circuit is depicted on the right in the figure below.



Based on the lecture material *Power Electronic Systems II*, Institute for Power Electronic Systems, ETH Zurich

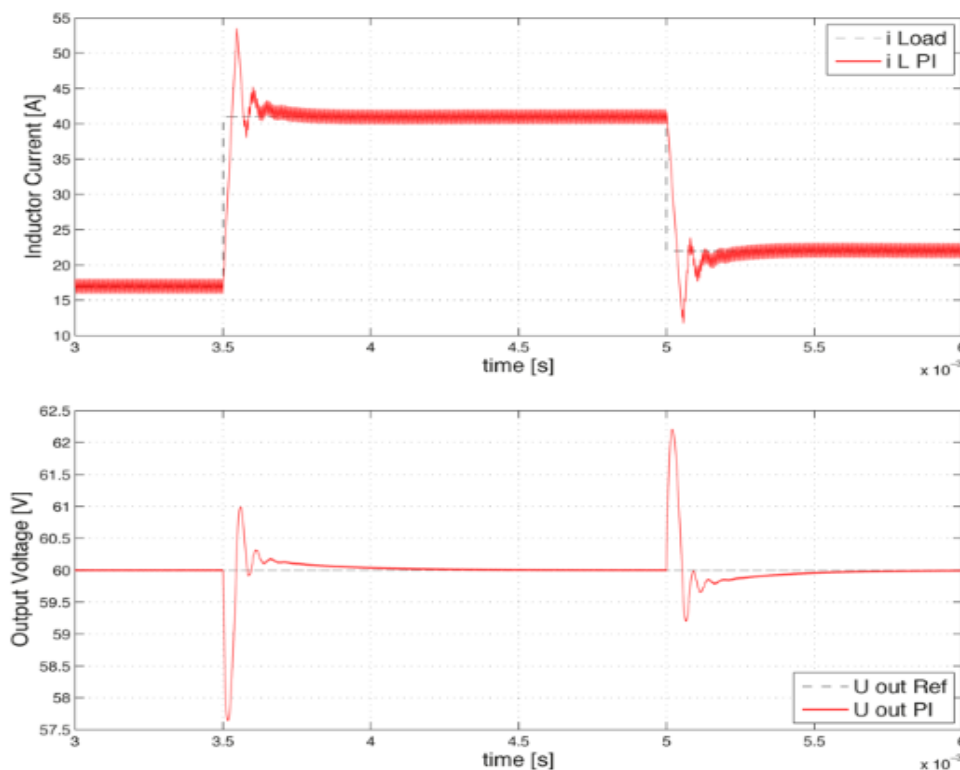
The states of the control oriented model, which is used as a model for the predictive controller, are the inductor current \hat{i}_L and the capacitor voltage \hat{u}_C . Further there are the input signal \hat{d} and the disturbances in the input voltage and the load current $\hat{w} = [\hat{u}_{in} \quad \hat{i}_{Load}]^T$. As an output signal the states \hat{i}_L and \hat{u}_C as well as the output voltage \hat{u}_{out} are considered. The small signal model (small signals are marked with a hat) in state-space form reads as:

$$\begin{aligned} \frac{d}{dt} \hat{x} &= \begin{bmatrix} -\frac{R}{L} & -\frac{1}{L} \\ \frac{1}{C} & 0 \end{bmatrix} \hat{x} + \begin{bmatrix} \frac{U_{in}}{L} \\ 0 \end{bmatrix} \hat{d} + \begin{bmatrix} \frac{D}{L} & -\frac{R}{L} \\ 0 & -\frac{1}{C} \end{bmatrix} \hat{w} \\ \hat{y} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ R & 1 \end{bmatrix} \hat{x} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & -R \end{bmatrix} \hat{w} \end{aligned} \quad \Leftrightarrow \quad \begin{aligned} \frac{d}{dt} \hat{x} &= A \cdot \hat{x} + B1 \cdot u + B2 \cdot \hat{w} \\ \hat{y} &= C \cdot \hat{x} + \begin{bmatrix} D2 \\ D4 \end{bmatrix} \cdot \hat{w} \end{aligned}$$

Control Objective by Using Model Predictive Control

The converter should provide a constant output voltage U_{out} of 60 V while delivering the power required by the load. The nominal load current I_{Load} is 22 A. The input voltage U_{in} is constant at level 144 V, while the load resistance varies in the range [1.5, 5] Ω .

Conventionally the output voltage of the Buck Converter was controlled by a PI controller. In the first plot below, the current \hat{i}_L in the inductor is shown, when the resistance in the load is reduced from 5 Ω to 1.5 Ω , i. e. from upper bound to the lower bound of the possibly required load resistance. The red curve represents the current in the inductor. Also the change in the output voltage is depicted when changing the load resistance.



From the figures above one can see that the current in the inductor has a high overshoot and the output voltage has a relatively long settling time when a change in the load resistance occurs.

Potential Benefits of Model Predictive Control

- Below it is shown that the size of the converter can be reduced by using a MPC controller designed with FORCES Pro. With the MPC controller it will be possible to limit the current in the inductor. With the warranty that the current does not exceed a certain upper bound, a smaller inductor can be built in and the costs are reduced.
- Also the controller designed with FORCES Pro will calculate the optimal input at every time step. The performance of the system is increased, i. e. less overshoot and faster settling time.

Model Predictive Control Design via FORCES Pro MATLAB® Interface

To design the FORCES Pro controller, the MPC setup has to be defined first. Below the requirements are shown. A prediction horizon of 25 is chosen. In the cost function R penalizes the deviation of the input signal from its reference value. The matrix Q penalizes the deviation of the states from its reference values. Notice that Q is defined such that a deviation of the inductor current to its reference value is less penalized than a deviation of the output voltage to its reference value. The input signal d to the PWM is limited to $[0, 1]$, while the inductor current should not exceed a current of 42 A. This overshoot limitation concerns the average inductor current. Below one can see, that this limit is exceeded by half of the currents peak-to-peak value. The constraints are consistently defined with the model, i. e. a current reduction by -20 A and a current enhancement by 20 A is allowed at most. This is equivalent to a current in the inductor in the range of $[2, 42]$ A.

MPC Setup and Constraints

```
% MPC Setup
N = 25;
Q = [.01, 0; 0, 10];
R = 1;
nx = 2;
nu = 1;

% Constraints
umin = 0;
umax = 1;
xmin = -20;
xmax = 20;
```

Next, the multistage problem is formulated. In this example, there exists a linear term f in the cost function due to the variable load, i. e. the steady-state inductor current changes. The cost function therefore reads as

$$(x^+ - x_{\text{ref}})^T Q (x^+ - x_{\text{ref}}) + (u - u_{\text{ref}})^T R (u - u_{\text{ref}})$$

To solve the optimization problem, the reference values need to be re-calculated at every time step. Below the parameters of the problem are marked red. The optimization variable of the multistage problem is $z_i = [u_i \quad x_{i+1}]^T$, where u is the input signal given to the system.

$$\begin{aligned} & \text{minimize} \quad \sum_{i=1}^N \frac{1}{2} z_i^T H_i z_i + \textcolor{red}{f}_i^T z_i && \text{(separable objective)} \\ & \text{subject to} \quad D_1 z_1 = \textcolor{red}{c}_1 && \text{(initial equality)} \\ & \quad \quad \quad C_{i-1} z_{i-1} + D_i z_i = \textcolor{red}{c}_i && \text{(inter-stage equality)} \\ & \quad \quad \quad \underline{z}_i \leq z_i \leq \bar{z}_i && \text{(bounds)} \end{aligned}$$

In this example three parameters have to be given to the solver.

- parameter(1): Represents the right hand side of the initial equality of the problem in standard form above.
- parameter(2): The linear term f of the cost function. This term contains the reference values of the states which are calculated based on the resistance of the load.
- parameter(3): Represents the right hand side of the inter-stage equality constraint for the stages $i = 2:N$ of the problem.

Next to the parameters, the dimensions of the variables, the equality constraints and the bounds have to be defined. The values defined in the MPC setup are added to the multistage problem in the section 'cost'. The terms in the equality constraints which are constant over all stages are defined in the section 'equality constraints'. After defining the output of the solver and the solver settings, the code for the controller can be generated.

Multistage Problem

```

% get stages struct of length N
stages = MultistageProblem(N);

% RHS of first eq. constr. is a parameter: stages(1).eq.c = -A*x0 - B2*w
parameter(1) = newParam('minusAx0_minusB2w',1,'eq.c');

% Linear Term depends on x_ref and u_ref
parameter(2) = newParam('Linear_Term',1:N,'cost.f');

% RHS of equality constraints for remaining stages: stages(i).eq.c = - B2*w
parameter(3) = newParam('minusB2w',2:N,'eq.c');

for i = 1:N

    % dimension
    stages(i).dims.n = nx+nu; % number of stage variables
    stages(i).dims.r = nx; % number of equality constraints
    stages(i).dims.l = 2; % number of lower bounds
    stages(i).dims.u = 2; % number of upper bounds

    % cost
    stages(i).cost.H = blkdiag(R,Q);

    % lower bounds
    stages(i).ineq.b.lbidx = 1:2; % lower bound acts on these indices
    stages(i).ineq.b.lb = [umin; xmin]; % lower bound on input u and state il

    % upper bounds
    stages(i).ineq.b.ubidx = 1:2; % upper bound acts on these indices
    stages(i).ineq.b.ub = [umax; xmax]; % upper bound on input u and state il

    % equality constraints
    if( i < N )
        stages(i).eq.C = [zeros(nx,nu), Ad];
    end
    stages(i).eq.D = [Bd1, -eye(nx)];
end

% define outputs of the solver
outputs(1) = newOutput('u0',1,1);

% solver settings
codeoptions = getOptions('DCDC_FORCES_Pro_Controller');

% generate code
generateCode(stages,parameter,codeoptions,outputs);

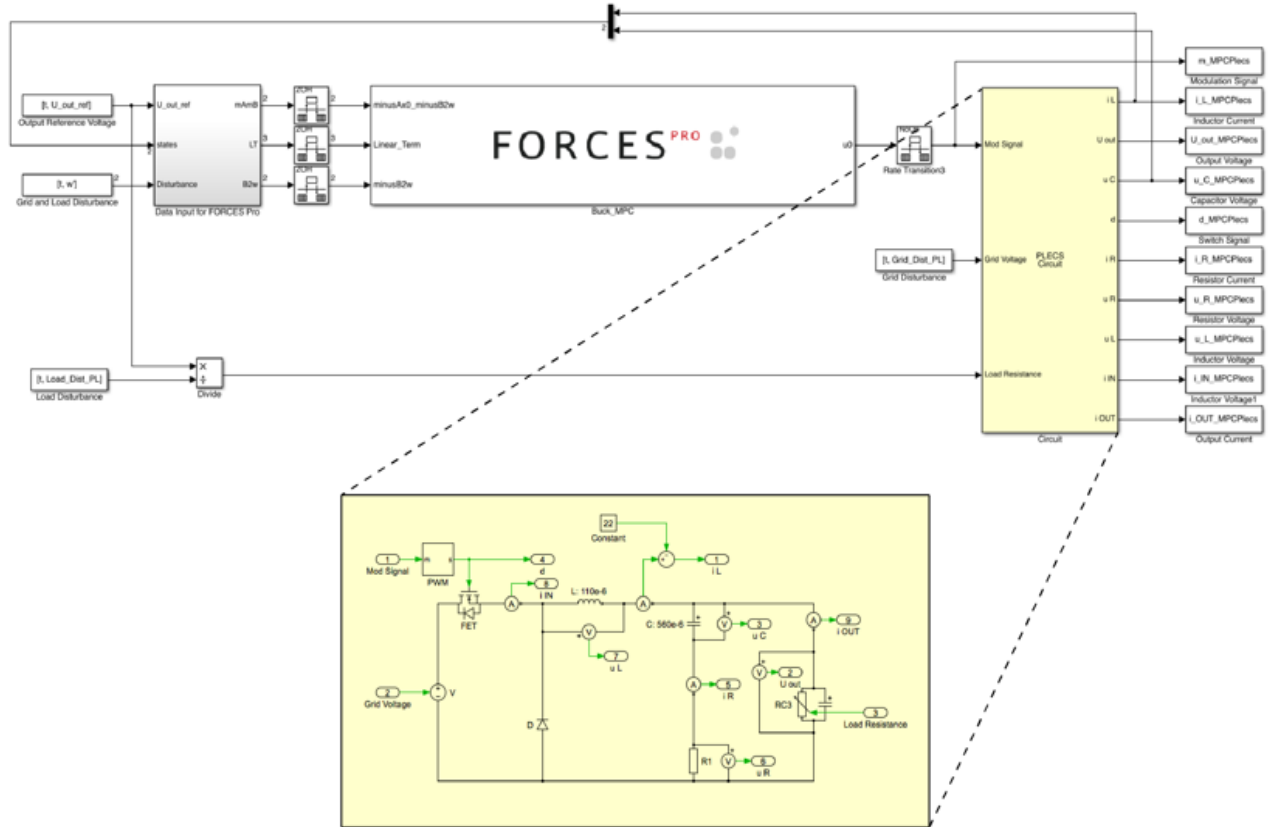
```

[Download code](#) with example simulation

Simulation of the PLECS® Model with Model Predictive Control

After the code is generated, the FORCES Pro Simulink® block can be added to the model DCDC_FORCES_Pro_viewer.slx as shown in the figure below (copy/past it from the file DCDC_FORCES_Pro_Controllercompact_lib.mdl in the folder 'DCDC_FORCES_Pro_Controller/Interface' generated by FORCES Pro).

The controller has a frequency of 100 kHz. To simulate the system with a time step of 1e-7 s, rate transition blocks are used. Below the Simulink® model DC_DC_FORCES_Pro.slx with the PLECS® circuit and the FORCES Pro controller is depicted.



In the grey box in the model depicted above, the three parameters which are the input to the FORCES PRO controller, are calculated.

- parameter(1): The right hand side of the initial equality constraint is $-Ad \cdot x - Bd2 \cdot w$.
- parameter(2): For the linear term of the cost function the reference values for the states and the input signal need to be calculated. The reference values are calculated by solving the linear system

$$\begin{bmatrix} Ad - I & Bd1 \\ Cd2 & Dd3 \end{bmatrix} \cdot \begin{bmatrix} x_{ref} \\ u_{ref} \end{bmatrix} = \begin{bmatrix} -Bd2 \cdot w \\ U_{out,ref} - Dd4 \cdot w \end{bmatrix}$$

which follows from the system equations in steady-state. To calculate the linear term f the reference values are plugged into the linear term $f = [-u_{ref} \cdot R \quad -x_{ref}^T \cdot Q]^T$, which is equal to

$$f = \begin{bmatrix} Ad - I & Bd1 \\ Cd2 & Dd3 \end{bmatrix}^{-1} \cdot \begin{bmatrix} 0 & -Bd2 \\ 1 & -Dd4 \end{bmatrix} \cdot \begin{bmatrix} U_{out,ref} \\ w \end{bmatrix} \cdot \begin{bmatrix} 0 & -R \\ -Q & 0 \end{bmatrix}$$

The matrices in the derivation above are explained in more detail in the system presented in the code available for this example. For more detail on 'reference tracking problems' with FORCES Pro click [here](#).

- parameter(3) is equal to $-Bd2 \cdot w$.

Comparison of Model Predictive Control and PI Control

In the figures below the evolution of the inductor current and the output voltage are compared when controlling the system with PI and with the MPC controller designed using FORCES Pro. It can be seen that the MPC controller is able to keep the inductor current within the limits defined above. However, this limits the tracking speed of the output voltage in the corresponding time interval. Overall, the tracking performance of the output voltage is increased compared to the baseline PI controller.

