Simulator for DC-DC Converters

Aleksandra Lekić

Department of Electronics, School of Electrical Engineering,
University of Belgrade, Belgrade, Serbia
lekic.aleksandra@etf.bg.ac.rs

September 5, 2018

Contents

List of Figures

1 Simulator Construction

Simulator is constructed using concepts of circuit theory provided in the thesis [1]. Circuit equations are constructed using Modified Nodal Analysis (MNA) approach [2] and solved in order to obtain state-space model of the converter [3].

In the simulator converter can be controlled using two controlling methods: nonlinear model predictive control (NMPC) [4] and PID control discretized using pole-zero placement algorithm [5]. Simulator gives a possibility of simulation in the open loop with constant frequency and duty-ratio. There is possibility for increasing number of controlling methods, which is our future plan.

Simulator is implemented in Python and C programming languages and it offers a simple and intuitive GUI. Complete simulation can be made on the level of input netlist and provided parameters.

1.1 Determination of the Possible States

For the provided number of switches, denoted as s, number of the possible converter's operating subsystems is 2^s. However, not all of the subsystems are possible, which will be analyzed in what follows.

According to chapter 4 in [6], all switches can be classified by checking which polarity of the current flows through while they are conducting and which polarity of voltage is blocked in the off state. Thus, in on current - off voltage plane switches are classified as one-quadrant, two-quadrant and four-quadrant. In the Fig. 1 one-quadrant switch is a transistor or a diode (Figs. 1a and 1b). Two-quadrant switches are current and voltage bidirectional, made from unidirectional switches from Figs. 1a-b, and depicted in Fig. 1c-d. Finally, four-quadrant switch is a combination of the previous two bidirectional switches.

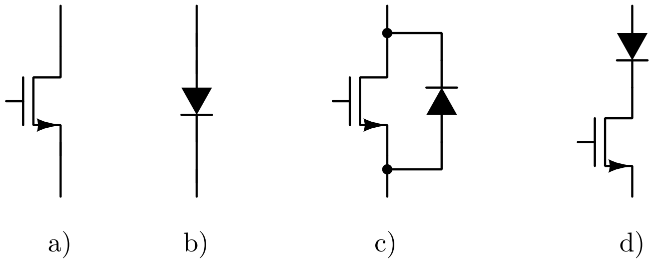


Figure 1: Switches realization: a) transistor, b) a diode, c) current bidirectional and d) voltage bidirectional.

Depending on the control method, switches can be controllable or not. Controllable switches are the ones constructed using a transistor. From the Fig. 1 controllable switches are transistor, current and voltage bidirectional.

Every state formed using switches is checked whether it can occur in the circuit. The states which don't satisfy following bullets are eliminated.

1. It is not in accordance with provided control scheme.

   Controller signal is brought to the number of controlling switches, so all of them are conducting at the same time. States which do not incorporate controlling requirements are eliminated.

2. It causes short connection of the voltage source or open connection of the current source.

3. Cut-sets that can cause DICM with the same inductors have different state of operation. Some of them require that inductors work in discontinuous mode (all switches in that cut-set are off) and some not (not all switches in that cut-set are off).

   Analogue, loops that can cause DCVM with the same capacitors have different state of operation. Some of them require that capacitors work in discontinuous mode (all switches in that loop are on) and some not (not all switches in that loop are on).

   DC-DC converters have number of states of which some are continuous and others are discontinuous. According to [7] there can only be two continuous states and some number of discontinuous states which is not bigger than number of state variables (i.e. inductors and capacitors).

   In [8] is described that discontinuous inductor current mode, DICM, can occur if there is a cut-set consisting off current sources, inductors and switches. Current of that cut-set is

$$\sum_i \pm i_{Li} + \sum_j \pm i_{Sj} + \sum_k \pm i_{Gk} \tag{1}$$

   and DICM happens when the sum of the switches currents falls to zero. Switches than stop conducting. According to classical definition, DICM is related only to current of one inductor and in common representation of the converter there are no current sources. Therefore in DICM inductor current reaches zero. But in general case as shown in [9], during the DICM there is one interval in which the inductor voltage is equal to zero, but the current has some constant value. Condition for that is existence of the loop consisting of inductors, capacitors and possibly voltage source, but without switches.

   Similar case is with discontinuous capacitor voltage mode DCVM. That discontinuous conduction mode can occur when in the converter exists loop with switches, capacitors and independent voltage sources. Sum of the voltages in the loop is

$$\sum_i \pm v_{Ci} + \sum_j \pm v_{Sj} + \sum_k v_{Gk} \tag{2}$$

   and the DCVM happens when the sum of capacitor voltages is equal to sum of independent voltage sources voltages inside the loop. That causes all switches to start conducting. Similar as for the inductors, during discontinuous interval capacitor voltage can be different than zero. It can be concluded (nowhere mentioned before) that it happens if there is cut-set consisting of capacitors, inductors and possibly current sources.

1.2 State-space model generation

Matrix state-space model of the switching converter is the way of solving system equations and one of the most used approaches for simulation of switching converters [3]. State-space model is determined by solving system of equations using slightly updated form of modified nodal analysis (MNA), which will be described in the first subsection.

1.2.1 Formulation of equations

Circuit equations in the state space form are formed automatically using of modified nodal analysis approach [2]. According to MNA, circuit variables are obtained by solving the following equation

$$\mathbf{A}\,\mathbf{x} = \mathbf{z}. \tag{3}$$

where vector **x** consists of node voltages, currents through independent voltage sources and currents through switches, derivatives of inductor currents and capacitor voltages. Vector **z** consists of independent sources values and inductor currents and capacitor voltages, while matrix **A** consists of unknown quantities.

Difference from the standard modified nodal analysis approach is made by adding equation which describes switch when it is conducting or not conducting, being:

$$\begin{aligned} i_S &= 0, && \text{when switch is off,} \\ v_S &= 0, && \text{when switch is on.} \end{aligned} \tag{4}$$

Equations for the inductors and capacitors are similar to equations of the independent current and voltage sources, respectively.

An extra equation that had to be solved for the inductor in the example of buck converter depicted in Fig. 2 is

$$\frac{d\,i_L}{dt} = \frac{v_2}{L} - \frac{v_3}{L}, \tag{5}$$

and for capacitor

$$v_3 = v_C. \tag{6}$$

For the example of the buck converter in state when switch S1 is on and S2 (represents a diode which is non-controllable switch) is off, complete set of system equations is

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{R} & 0 & 0 & 0 & 0 & C \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -\frac{1}{L} & \frac{1}{L} & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ i_{IN} \\ i_{S1} \\ i_{S2} \\ \frac{d\,i_L}{dt} \\ \frac{d\,v_C}{dt} \end{bmatrix} = \begin{bmatrix} 0 \\ -i_L \\ i_L \\ v_{IN} \\ 0 \\ 0 \\ 0 \\ v_C \end{bmatrix} \tag{7}$$

An other difference in MNA method is subsystem in which DICM or DCVM occurs. In the general case sum of inductor currents or capacitor voltages is constant. The discontinuous mode provides system reduction by one equation in terms of cut-set equation/loops equations. Equation for DICM is reduced to

$$\sum_i \pm i_{Li} + \sum_k \pm i_{Gk} = 0, \tag{8}$$

and for DCVM to

$$\sum_i \pm v_{Ci} + \sum_j \pm v_{Sj} = 0. \tag{9}$$

Thus, DCVM and DICM subsystems should provide one extra equation by giving dependence of capacitor voltages or inductor currents. On the other hand one of the unknown state variables is symbolically changed to the combination of sums of inductor currents and current sources currents/capacitor voltages and voltage sources voltages.

Common case for DICM is when inductor current falls to zero. Then the equation is modified by setting inductor voltage equal to zero. Other, more complicated case is when the number of inductors in the DICM cut-set is greater than one. From the Fig. 3 it can be seen that this kind of DICM occurs in Ćuk converter. Equation of inductor currents then becomes

$$i_{L1} - i_{L2} = 0 \tag{10}$$

and therefore, current i_{L1} is changed by $-i_{L2}$ inside vector $\mathbf{z}$. An extra equation is

$$\frac{d\,i_{L1}}{dt} - \frac{d\,i_{L2}}{dt} = 0. \tag{11}$$

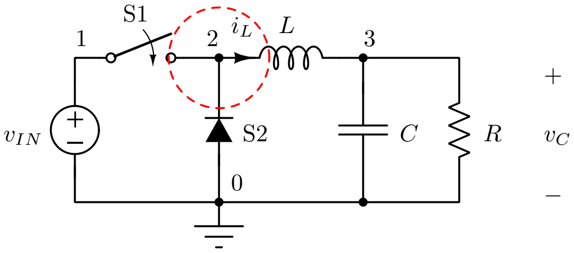


Figure 2: Buck converter.

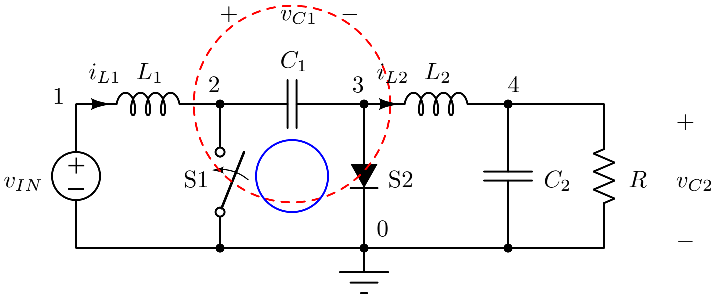


Figure 3: Ćuk converter.

After forming the system of linear equations, equation (3) is solved using SYMPY and reduced row echelon form. Afterwards, equations for the state variables are derived.

1.2.2 State-Space Model Matrices Formulation

First step of simulation is to formulate state-space model for each subsystem. Assuming that $\mathbf{x}$ is the vector that consists of state variables, i.e. inductor currents and capacitor voltages, $\mathbf{u}$ is vector of independent sources and $\mathbf{y}$ is output vector, the state-space matrix representation of each subsystem is

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}_i\,\mathbf{x} + \mathbf{B}_i\,\mathbf{u},\\ \mathbf{y} &= \mathbf{C}_i\,\mathbf{x} + \mathbf{D}_i\,\mathbf{u},\end{aligned} \tag{12}$$

where i stems for i-th subsystem.

In our work we will consider switching converter as a circuit consisting of voltage input, switching cell and the circuit's output. This is approach is valid because it covers most of non-degenerative DC-DC converters [1, 10]. In addition, the special attention will be given to the circuits with the same number of inductors and capacitors, because in that case the output voltage is not any more independent of the load. Switching cell is considered as three-port element made of switches, inductors and capacitors only, but in calculation losses are neglected. This work allows adding losses, switches' threshold voltages and output current source and therefore state-space model can have more than one independent source.

Depending of switch realization and the switching cell configuration, during one switching interval a number of subintervals, denoted as subsystems, may occur. However, that number is equal to $2 + s$, where 2 subintervals are usual CCM intervals and s presents different number of discontinuous modes, as in [7,8]. Standard number of modes is equal to two or three and in that case, state-space model during one switching period is

$$\dot{\mathbf{x}}(t) = \begin{cases} \mathbf{A}_1\,\mathbf{x}(t) + \mathbf{B}_1\,\mathbf{u}(t) & n\,T_S \le t < (n+d)\,T_S \\ \mathbf{A}_2\,\mathbf{x}(t) + \mathbf{B}_2\,\mathbf{u}(t) & (n+d)\,T_S \le t < (n+d+d_2)\,T_S \\ \mathbf{A}_3\,\mathbf{x}(t) + \mathbf{B}_3\,\mathbf{u}(t) & (n+d+d_2)\,T_S \le t < (n+1)\,T_S \end{cases} \tag{13}$$

and

$$\mathbf{y}(t) = \begin{cases} \mathbf{C}_1\,\mathbf{x}(t) + \mathbf{D}_1\,\mathbf{u}(t) & n\,T_S \le t < (n+d)\,T_S \\ \mathbf{C}_2\,\mathbf{x}(t) + \mathbf{D}_2\,\mathbf{u}(t) & (n+d)\,T_S \le t < (n+d+d_2)\,T_S \\ \mathbf{C}_3\,\mathbf{x}(t) + \mathbf{D}_3\,\mathbf{u}(t) & (n+d+d_2)\,T_S \le t < (n+1)\,T_S \end{cases} \tag{14}$$

1.2.3 Solving State-Space Equations Using Eigenvalues

Let m be number of inductors in the circuit, n number of capacitors, p independent sources number and q number of output variables. Previous vectors and matrices are therefore sized as $\mathbf{A}_{(m+n)\times(m+n)}$, $\mathbf{B}_{(m+n)\times p}$, $\mathbf{C}_{q\times(m+n)}$ and $\mathbf{D}_{q\times p}$.

State-space equations can be solved analytically and their solution has the form of [11]

$$\mathbf{x}(t) = \mathbf{x}_p(t) + \mathbf{x}_h(t) \tag{15}$$

where $\mathbf{x}_p$ is particular and $\mathbf{x}_h$ is homogeneous solution. Homogeneous solution is in exponential form with roots that are solution of subsystems' characteristic polynomial. Other way to find these roots is to calculate eigenvalues of the matrix $\mathbf{A}$. Matrix $\mathbf{A}$ has $\underline{\lambda}_j$ eigenvalues, $j \in \{1,\,2,\,\cdots m+n\}$, which are presented as complex. However, eigenvalues are real or in conjugate-complex pairs. Homogeneous solution is in case of distinct eigenvalues presented as

$$\mathbf{x}_h(t) = \sum_{i=1}^{m+n} \underline{\mathbf{x}}_i\,\mathrm{e}^{\underline{\lambda}_i\,(t-t_0)} = \sum_{i=1}^{m+n} \underline{\mathbf{X}}_i\,\mathrm{e}^{\underline{\lambda}_i\,t}, \tag{16}$$

where $t = t_0$ presents the beginning of the operation the current subsystem. Particular solution can be found solving the equation

$$\mathbf{A}\,\mathbf{x}_p(t) + \mathbf{B}\,\mathbf{u}(t) = 0. \tag{17}$$

There are two cases: if matrix $\mathbf{A}$ is invertible, particular solution is equal

$$\mathbf{x}_p(t) = -\mathbf{A}^{-1}\,\mathbf{B}\,\mathbf{u}(t). \tag{18}$$

The second case is when matrix $\mathbf{A}$ is singular and then

$$\mathbf{x}_p^*(t) = (\mathbf{P}^{-1}\,\mathbf{Q}\,\mathbf{u}(t))^T \tag{19}$$

where matrix $\mathbf{P}$ is matrix $\mathbf{A}$ got by eliminating rows and columns corresponding to elements working in discontinuous mode (in matrix $\mathbf{A}$ these rows and columns are originally equal to zero) and $\mathbf{Q}$ is $\mathbf{B}$ vector in which the rows at the same location are avoided (in $\mathbf{B}$ those rows are zero rows). Particular solution is then formed by writing zero or some limiting constant value from the end of previous interval of time (example Ćuk converter, operating in DCM when $i_{L1} + i_{L2}$ falls to zero) at the place of discontinuous element which current or voltage is equal to zero during this interval of time and the other state variables are formed by taking corresponding value from (19). Specially this is the third interval as mentioned above, but in the general case when there are more inductors and capacitors, it can be more than one discontinuous operating state.

Assuming that $t = t_0$ is the end of previous and the beginning of the next time interval, the value of state variables at the beginning of interval is $\mathbf{x}(t_0)$ and it is assumed as known as the vector of state variables at the end of the previous time interval.

Using previously defined system of equations, we get

$$\begin{aligned} &\mathbf{x}(t_0) = \mathbf{x}_p(t_0) + \sum_{i=1}^{m+n} \underline{\mathbf{X}}_i \, \mathrm{e}^{\underline{\lambda}_i \, t_0}, \\ &\dot{\mathbf{x}}(t_0) = \mathbf{A}\,\mathbf{x}(t_0) + \mathbf{B}\,\mathbf{u}(t_0), \\ &\vdots \\ &\mathbf{x}^{(m+n-1)}(t_0) = \mathbf{A}^{m+n-1}\,\mathbf{x}(t_0) + \sum_{j=1}^{m+n-1} \mathbf{A}^{m+n-1-j}\,\mathbf{B}\,\mathbf{u}^{(j-2)}(t_0). \end{aligned} \tag{20}$$

Solution of (20) provides quantities for further calculation. Easy way to do that is to solve matrix system of equations

$$\mathbf{M} \begin{bmatrix} \underline{\mathbf{X}}_1 \\ \underline{\mathbf{X}}_2 \\ \vdots \\ \underline{\mathbf{X}}_{m+n} \end{bmatrix} = \begin{bmatrix} \mathbf{x}(t_0) - \mathbf{x}_p(t_0) \\ \mathbf{A}\,\mathbf{x}(t_0) + \mathbf{B}\,\mathbf{u}(t_0) - \dot{\mathbf{x}}_p(t_0) \\ \vdots \\ \mathbf{A}^{(m+n-1)}\,\mathbf{x}(t_0) + \sum_{j=1}^{m+n-1} \mathbf{A}^{(m+n-1)-j}\,\mathbf{B}\,\mathbf{u}^{(j-2)}(t_0) - \mathbf{x}_p^{(m+n-1)}(t_0) \end{bmatrix} \tag{21}$$

where matrix $\mathbf{M}_{(m+n)^2\,\times\,(m+n)^2} = [\mathbf{M}_{i,\,j}]_{i,j\in[0,m+n),\; i,j\in\mathbb{N}}$ and $(m+n)\times(m+n)$ matrix

$$\mathbf{M}_{i,\,j} = \operatorname{diag}\left\{\underline{\lambda}_j^i \, \mathrm{e}^{\underline{\lambda}_j \, t_0}\right\}. \tag{22}$$

It should be noted that in case of DC-DC converters vector $\mathbf{u}(t) = \mathbf{u}$ is constant. That means that it does not have derivative as well as the partial solution, which gives equation (23) as follows.

$$\mathbf{M} \begin{bmatrix} \underline{\mathbf{X}}_1 \\ \underline{\mathbf{X}}_2 \\ \vdots \\ \underline{\mathbf{X}}_{m+n} \end{bmatrix} = \begin{bmatrix} \mathbf{x}(t_0) - \mathbf{x}_p \\ \mathbf{A}\,\mathbf{x}(t_0) + \mathbf{B}\,\mathbf{u} \\ \vdots \\ \mathbf{A}^{(m+n-1)}\,\mathbf{x}(t_0) + \mathbf{A}^{(m+n-2)}\,\mathbf{B}\,\mathbf{u} \end{bmatrix} \tag{23}$$

After determination of the particular solution, algorithm follows described procedure. After determining coefficients of homogeneous solution and particular solution during the whole interval solution can be determined as

$$\dot{\mathbf{x}}(t) = \mathbf{N} \begin{bmatrix} \underline{\mathbf{X}}_1 \\ \underline{\mathbf{X}}_2 \\ \vdots \\ \underline{\mathbf{X}}_{m+n} \end{bmatrix} + \mathbf{x}_p \tag{24}$$

with matrix $\mathbf{N}_{(m+n)\,\times\,(m+n)^2}$ equal to:

$$\mathbf{N} = \Big[\mathbf{N}_1\,\mathbf{N}_2 \cdots \mathbf{N}_{m+n}\Big], \qquad \mathbf{N}_i = \operatorname{diag}\left\{\mathrm{e}^{\underline{\lambda}_i \, t}\right\}. \tag{25}$$

1.2.4 Solving State-Space Equations Using Different Integration Methods

An other implemented way for solving subsystem equations of the format (12) by its discretizing and integration of the discretized equation. NMPC-codegen [12] offers variety of integration methods which can be used. For this simulator is chosen Runge-Kutta forth integration method.

Besides standard integrators, CASADI [13] can be used for construction of the integrator and generation of its C code.

1.3 Possible transitions

According to [14] there are six possible classes of quasi resonant converters. It can be seen that even in the case of PWM converter, there is possibility of operating in DQRM - discontinuous quasi resonant mode. In [8] there is an example of buck converter with input filter. This circuit has possibility of operating in DQRM and also satisfies the condition for Qn-PWM quasi resonant converter (CcS and $Ll\overline{S}$).

After state determination, possible transitions are formed. For existence some of the quasi-resonant operation modes, it is necessary to satisfy conditions in Fig. 3 in [14]. It shows that the same tests can be performed for both PWM and quasi-resonant converters. Those conditions are summed in table 1.

Table 1: Complete set of quasi-resonant classes with two resonant elements.

	LcS	$Lc\overline{S}$	LlS	$Ll\overline{S}$
ClS	x	ZV	ZV-QSW	
$Cl\overline{S}$	ZC	x		
CcS	ZC-QSW		x	Qn-PWM
$Cc\overline{S}$			Qf-PWM	x

According to the paper [14], those cut-sets and loops are determined at the same time while adding resonant elements to already existing PWM circuit. In our case the circuit is already given with all resonant and nonresonant elements. Thus, the algorithm for determining cut-sets and loops are as following:

- CcS and $Cc\overline{S}$ mean that resonant capacitor forms cut-set with the switch $\mathrm{S}/\overline{\mathrm{S}}$ and a non-empty set of PWM inductors.

  For checking this condition it is important that none of the inductors inside described cut-set participates in so-called DICM cut-sets. That is why the inductors that participate in DICM cut-sets can be taken to be resonant.

- LcS and $Lc\overline{S}$ are cut-sets with the resonant inductor, switch $\mathrm{S}/\overline{\mathrm{S}}$ and a possibly empty set of PWM inductors.

- ClS and $Cl\overline{S}$ are loops with the resonant capacitor, switch $\mathrm{S}/\overline{\mathrm{S}}$ and a possibly empty set of PWM capacitors.

- LlS and $Ll\overline{S}$ mean that resonant inductor forms loop with the switch $\mathrm{S}/\overline{\mathrm{S}}$ and a non-empty set of PWM capacitors.

  For checking this condition it is important that none of the capacitors inside described loop participates in so-called DCVM loops. That is why the capacitors that participate in DCVM loops can be taken to be resonant.

Described checking for loops and cut-sets according to table 1 classify the type of the converter. In case that the converter is none of quasi-resonant classes, it is PWM converter.

For PWM converters in case of impossibility of DQRM, transitions can be (it is represented binary where 2^0 bit presents controllable switch equal to 1 during interval $n\,T_S \leq t < (n+d)\;T_S$ and 2^1 bit is $\overline{\mathrm{S}}$ with possible realisation as a diode).

Q-PWM classes can be seen as classes of PWM converters operating in DQRM. In addition of previous transition it is possible 01 → 11 → 10 → 00.

1.4 Controller Design

1.4.1 Discretized PID Controller

Controller is designed as discrete representation of the corresponding transfer function using zero-pole placement method, which showed good experimental results [5].

General form of controller transfer function in analog domain is given with:

$$G_c(s) = \frac{d(s)}{e(s)} = G_{c0} \frac{\prod\limits_{i=1}^{n} \left(1 + \dfrac{s}{s_{z1}}\right)}{\prod\limits_{j=1}^{m} \left(1 + \dfrac{s}{s_{p1}}\right)} \tag{26}$$

where, n is the order of nominator and m is the order of denominator. Pole-zero matching transformation, $G_c(s) \to G_d(z)$, is done as follows [5]:

- Finite poles and zeros of continuous transfer function $G_c(s)$, $s_{p/z,i} = s_{real,i} + js_{imag,i}$, are transformed into their discrete equivalent by using $z_{p/z,i} = e^{\left(s_{real,i}+js_{imag,i}\right)T_s}$, where T_s represents sampling period.
- Zeros at $s_{z,i} \to \infty$ are mapped into $z_{z,i} = -1$.
- DC gain of discrete transfer function is set to be equal to the DC gain of continuous transfer function $|G_c(s = j \cdot 0)| = |G_d(z = 1 + j \cdot 0)|$

Hence, using the obtained discrete transfer function, difference equation (or recurrent equation) for duty ratio determination is derived in a simple manner:

$$G_d(z) = \frac{d(z)}{e(z)} = \frac{\sum\limits_{l=0}^{i} a_{d,l} z^l}{\sum\limits_{k=0}^{j} b_{d,k} z^k} = \frac{\sum\limits_{l=0}^{i} \dfrac{a_{d,l}}{b_{d,j}} z^{l-j}}{1 + \sum\limits_{k=0}^{j-1} \dfrac{b_{d,k}}{b_{d,j}} z^{k-j}}, \tag{27}$$

$$d[n] = -\left(\sum_{k=0}^{j-1} \frac{b_{d,k}}{b_{d,j}} d[n+k-j]\right) + \sum_{l=0}^{i} \frac{a_{d,i}}{b_{d,j}} e[n+l-j]. \tag{28}$$

For the defined controller, values of errors and duty ratio are updated once during switching period. Converter output values, which are same as the controller input values, at the end of the switching period $t = n\,T_S$ are sufficient for that purpose. Those values are already known, calculated, and thus new value for duty ratio is easily obtained.

1.5 Switching time determination

As in the circuit two switching type are possible: internal, which occurs in case that some DICM cut-set or DCVM loop satisfies condition for discontinuous mode, and external triggered by controller, exact time of the transition and state-variables' and output variables' values have to be known. Algorithm used for internal switching time determination relies on second-order Newton's method. External switching is closely related to controller design. According to the type of switching, two methods are developed.

1.5.1 Internal switching

Function that describes internal switching has a form

$$f(\mathbf{x}) = \alpha\, \mathbf{x} + \beta\, \mathbf{u}, \tag{29}$$

where α and β are vectors that consists of values 0, 1 and -1. Those values are coefficients that contain information of state variables and independent sources taking place in equation (8) or (9) depending of discontinuous mode being checked.

According to second-order Newton's method, new time is

$$t_{n+1} = t_n + \frac{1}{\dfrac{f^{(2)}(\mathbf{x}_n)}{2\,\dot{f}(\mathbf{x}_n)} - \dfrac{\dot{f}(\mathbf{x}_n)}{f(\mathbf{x}_n)}}. \tag{30}$$

State-space model is very useful in this case, because

$$\dot{f}(\mathbf{x}_n) = \alpha\, \mathbf{A}\, \mathbf{x}_n + \alpha\, \mathbf{B}\mathbf{u} \tag{31}$$

and

$$f^{(2)}(\mathbf{x}_n) = \alpha\, \mathbf{A}^2\, \mathbf{x}_n + \alpha\, \mathbf{A}\, \mathbf{B}\mathbf{u}. \tag{32}$$

This method is used because it converges fast, as mentioned in [15].

1.6 NMPC Design

The MPC problem of a DC-DC converter in CCM is formulated as a multistage optimal control problem. One switching period is divided in 2 subsystem intervals $\tau_{i,j} \le t < \tau_{i+1,j}$ for $i = 0, 1$ and $j \in \{0, \dots, N-1\}$. In each interval $\tau_{i,j} \le t < \tau_{i+1,j}$ the system equations of subsystem i, given in (12), are discretized using a nonlinear integrator, so $\mathbf{x}^0_{i,j}$ and $\mathbf{x}^1_{i,j}$ denote the state variables values at the beginning and at the end of operation in the i-th subsystem of the j-th switching period, and $d_{i,j} = \frac{\tau_{i+1,j} - \tau_{i,j}}{T_S}$ denotes the duration of the interval $\tau_{i,j} \le t < \tau_{i+1,j}$.

The MPC problem is then given as

$$\begin{aligned}
\textbf{minimize}\;\; & \ell_N(\mathbf{x}^1_{1,N}) + \sum_{j=0}^{N-1} \left(\psi(d_{0,j}) + \ell(\mathbf{x}^1_{1,j})\right), && (33)\\
\textbf{subject to}\;\; & \mathbf{x}^1_{i,j} = F_i\left(\mathbf{x}^0_{i,j}, d_{i,j}\, T_S\right), \\
& \qquad\qquad i \in \{0, 1\},\; j \in [0, N-1], && (34)\\
& \mathbf{x}^0_{1,j} = \mathbf{x}^1_{0,j},\; j \in \{0, N-1\}, && (35)\\
& d_{0,j} + d_{1,j} = 1,\; j \in [0, N-1], && (36)\\
& d_{i,j} \in [0, 1],\; i = 0, 1, j \in [0, N-1]. && (37)
\end{aligned}$$

2 Program implementation

Implementation of the DC-DC simulator is written in Python programming language. This program offers interface which allows choosing the type of the control. As an input file, it takes netlist of the converter, see section 3.2.

Setup of the circuit model is described in the flowchart from the Fig. 4. It can be seen that from the netlist are constructed subsystems (states) which can occur. For every subsystem are determined system matrices, both in numpy and CASADI [13] format. And at the end a list of all transitions for the input circuit is formed. For this processing are used concepts of DC-DC converter's simulation given in section 1.

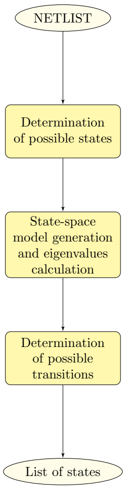

NETLIST
Determination of possible states
State-space model generation and eigenvalues calculation
Determination of possible transitions
List of states

Figure 4: Initialization for transient simulation flowchart.

Simulation, which proceeds, can be done in open loop, in the closed loop using PID controller discretized using pole-zero placement method or closed loop with nonlinear model predictive controller.

2.1 Simulation of PID Controlled Converter

With determined possible states and transitions and also calculated matrices and eigenvalues, simulation proceeds starting at time $t = 0$ with given initial values for state variables. Simulation of PID controller and open loop converter starts with the initial state and continues with the same procedure until it reaches provided stop time. For the step interval by default is taken 1000 points per period, but precision can be given as an input in the program. After each step interval it is checked if there can be changing of state either by controller (limited duty ratio) or by some of discontinuities inside converter. In case that change occurs, by intersect algorithm enough precise moment when the transition occurs is determined. Described procedure is presented in Fig. 5 by flowchart.

According to controller design, duty ratio and error are calculated only once during switching period. On the other hand, condition for state change caused by controller triggering has to be checked after every time interval which is number of times smaller that switching period. Checking is done on basis if the controlled switch or switches turns on or off. In both cases if the switching occurs at the moment which is between current and next time interval, a new time point must be added in which all state variable values must be calculated and in which change state occurs.

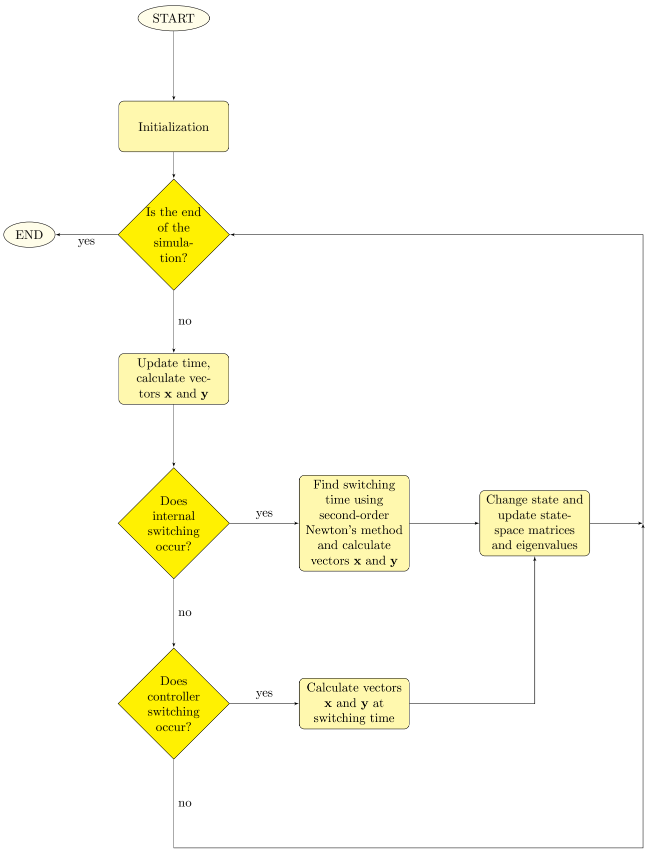

START
Initialization
Is the end of the simulation?
yes
END
no
Update time, calculate vectors **x** and **y**
Does internal switching occur?
yes
Find switching time using second-order Newton's method and calculate vectors **x** and **y**
Change state and update state-space matrices and eigenvalues
no
Does controller switching occur?
yes
Calculate vectors **x** and **y** at switching time
no

Figure 5: Simulation flowchart.

2.2 Simulation of NMPC Converter

Simulation of the PWM converter controlled using NMPC is described with the flowchart from the Fig. 6. The program takes generated state-space model and the states and forms the MPC model. Then, the cost function is produced and the controller is generated. It must be noted that the controller is generated in C and this code can be used for its implementation inside the microcontroller or a DSP. After generation of the code, the transient simulations starts.

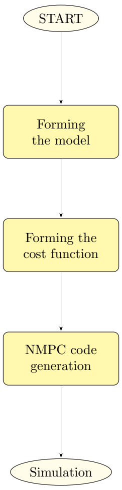


Figure 6: Initialization for transient simulation with NMPC flowchart.

3 Simulator Usage

3.1 Installation

For successfully running the code the following programs/packages have to be installed. Program is written in Python 3.6 using CASADI [?] package. It compiles the controller in C using gcc GNU compiler with cmake, which creates a build system with clang compiler and Cmake.

For more details about installation, please see [12].

3.2 Circuit Components and Forming of the Netlist

Circuit components should be written in the netlist in the following format. Every component is described in only one line.

3.2.1 Voltage source (V)

Voltage source is presented in the format: `V index node_1 node_2 value`.

`V` presents a symbol for a voltage source, `index` is the corresponding voltage source index, `node_1` is positive end and `node_2` negative end of the voltage source, `value` is voltage given as a real number.

3.2.2 Current source (I)

Current source is presented in the format: `I index node_1 node_2 value`.

`I` presents a symbol for a current source, `index` is the corresponding current source index, `node_1` is positive end and `node_2` negative end of the current source, `value` is current given as a real number.

3.2.3 Resistor (R)

Resistor is presented in the format: `R index node_1 node_2 value`.

`R` presents a symbol for a resistor, `index` is the corresponding resistor index, `node_1` is positive end and `node_2` negative end of the resistor, `value` is resistance given as a positive real number.

3.2.4 Capacitor (C)

Capacitor is presented in the format: `C index node_1 node_2 value initial_value`.

`C` presents a symbol for a capacitor, `index` is the corresponding capacitor index, `node_1` is positive end and `node_2` negative end of the capacitor, `value` is capacitance given as a positive real number, and `initial_value` is the capacitor voltage at the beginning of the simulation. Initial value is set to zero as default if it is not provided as an input.

3.2.5 Inductor (L)

Inductor is presented in the format: `L index node_1 node_2 value initial_value`.

`L` presents a symbol for inductor, `index` is the corresponding inductor index, `node_1` is positive end and `node_2` negative end of the inductor, `value` is inductance given as a positive real number, and `initial_value` is the inductor current at the beginning of the simulation. Initial value is set to zero as default if it is not provided as an input.

3.2.6 Switch (SW)

Switch is presented in the from: `SW index type_sw node_1 node_2`.

`SW` is a symbol for switch, `index` is the switch index, `type_sw` denotes the type of the switch: 1 is for transistor, 2 - diode, 3 - current bidirectional, 4 - voltage bidirectional, 5 - four quadrant switch, as in Fig. 1. Nodes `node_1` and `node_2` present positive and negative end of the component, respectively.

3.2.7 Netlist example

One example of the netlist for the standard buck converter depicted in Fig. 2 is as it follows. This example can be found in file: `examples\buck\standard_buck\buck.txt`.

```
V 1 1 0 20.0
SW 1 1 1 2
SW 2 2 2 0
L 1 2 3 200e-6
C 1 3 0 1e-3
R 1 3 0 5.0
```

3.3 PID Controller Definition File

For the desired PID controller transfer function should be added as an input to the simulator. This function is read from the text file in which is written a list containing lists of transfer functions: `[list_1, list_2, ..., list_l]`. Each transfer function list contains three parameters. First parameter is a list containing n nominator coefficients, equation (26). The second parameter presents list of m denominator coefficients and the third is an output desired value. The lists of the transfer function information should be provided in the order in which the output variables are selected.

For the example of the buck converter a PID controller is designed with the respect to the node 3 and its format is:

```
[[[1.05e06, 3.193e09, 3.161e11], [3.011e05, 1.076e10, 0], 5]]
```

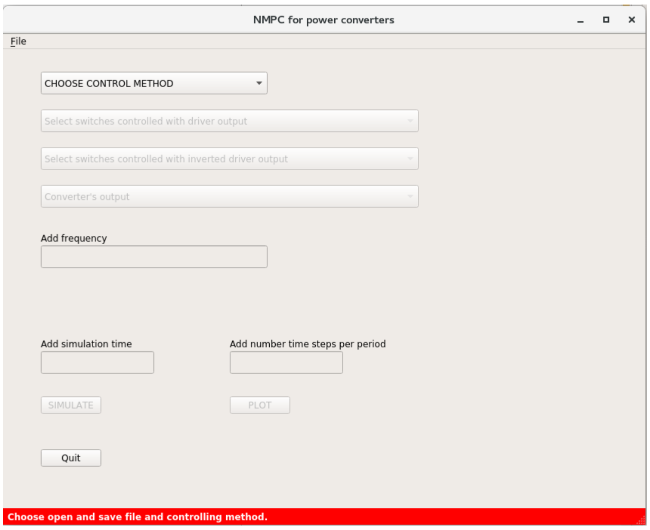
This example can be found in `examples\buck\synchronous_buck\buck_controller.txt`.

3.4 Application Usage

Application starts by running file `run.py` in Python when it open the window from Fig. 7. With the command `File`→`Open file` it can be navigated to the netlist file location (preferably `.txt` file, see section 4). Command `File`→`Save file` provides a location and a name for the simulation output files: a `.npy` file in which are saved transient values of state variables and a figure in pdf format of state variables' transient diagrams.

After choosing input and output files, controlling method can be chosen. There are three different controlling methods available: PWM with constant frequency and duty-ratio, PID controller and NMPC. After choosing controlling method, other lists became enabled.

First falling list contains all switches with their indexes which can be controlled using PWM signal. That means that these switches must be constructed using transistors, i.e. from Fig. 1 these switches are a), c) and d). After choosing all "on state" switches, as they are called in program, because they are controlled using buffered PWM signal, an other list is enabled. The second list contains all controllable switches which are not selected in the first list. These switches can be selected to be controlled with inverted PWM signal. Thus, they are denoted as "off state" switches. In the third falling list an output of the converter has to be chosen.

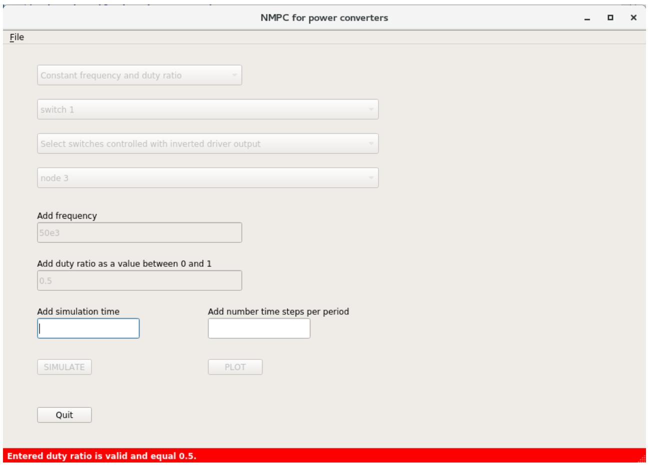

NMPC for power converters
File
CHOOSE CONTROL METHOD
Select switches controlled with driver output
Select switches controlled with inverted driver output
Converter's output
Add frequency
Add simulation time
Add number time steps per period
SIMULATE
PLOT
Quit

Choose open and save file and controlling method.

Figure 7: Application start widow.

Output can be inductor current, capacitor voltage or the node voltage. Multiple outputs can be selected.

Previous procedure is the same for the all controlling methods. For constant frequency/duty-ratio and NMPC, the screen looks like in Fig. 8. Then should be chosen frequency, duty-ratio, simulation time and precision (given as number of points per period). After that the simulation button enables and with pressing it, simulation starts. Simulation results can be plotted by pressing the PLOT button, and then the plot opens in the next dialog window.


NMPC for power converters
File
Select switches controlled with inverted driver output
SIMULATE
PLOT
Quit

Constant frequency and duty ratio

switch 1

node 3

Add frequency: 50e3

Add duty ratio as a value between 0 and 1: 0.5

Add simulation time

Add number time steps per period

Entered duty ratio is valid and equal 0.5.

Figure 8: NMPC and constant frequency/duty-ratio window.

PID controller simulation is a bit different. Besides frequency, number of points per period, simulation time, the coefficients of the controller have to be chosen. Coefficients should be provided in the textual file in the format described in the subsection 3.3. Simulation and plotting procedure are the same as previous.

After running the simulation, dialog window can be closed by pressing QUIT button.

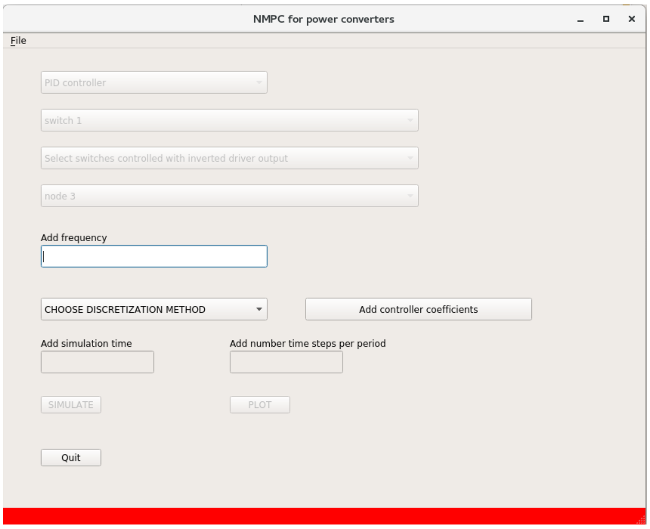

NMPC for power converters
File
PID controller
switch 1
Select switches controlled with inverted driver output
node 3
Add frequency
CHOOSE DISCRETIZATION METHOD
Add controller coefficients
Add simulation time
Add number time steps per period
SIMULATE
PLOT
Quit

Figure 9: PID controller window.

4 Application examples

1. **Synchronous buck converter with controller:** `examples\buck\synchronous_buck`

   For synchronous buck converter with circuit parameters $v_{IN} = 20$ V, $L = 200\,\mu$H, $C = 1$ mF, $R = 5\,\Omega$ and $f_S = 100$ kHz designed to operate with output voltage equal 5 V and with transfer function in Laplace domain

$$H(s) = \frac{1.05 \cdot 10^6\ s^2 + 3.193 \cdot 10^9\ s + 3.161 \cdot 10^{11}}{3.011 \cdot 10^5\ s^2 + 1.076 \cdot 10^{10}\ s}, \tag{38}$$

   the transient simulation diagrams are depicted in Fig. 10.

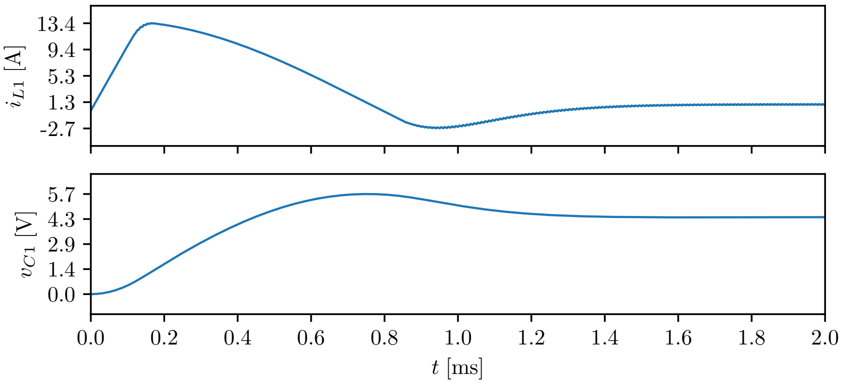


Figure 10: Transient diagrams of state variables for synchronous buck converter shown in Fig. 2 for the first 2 ms.

2. **Standard buck converter with PID control in DICM:** `examples\buck\standard_buck`

   For the standard buck converter designed to operate in DICM with the applied control:

$$H(s) = \frac{0.06\,s + 90}{s},$$

   for the output voltage designed to be equal 5 V, and circuit values: $v_{IN} = 10$ V, $L = 25\,\mu$H, $C = 330\,\mu$F, $R = 15\,\Omega$ and $f_S = 100$ kHz, the transient diagrams are depicted in Fig. 11.

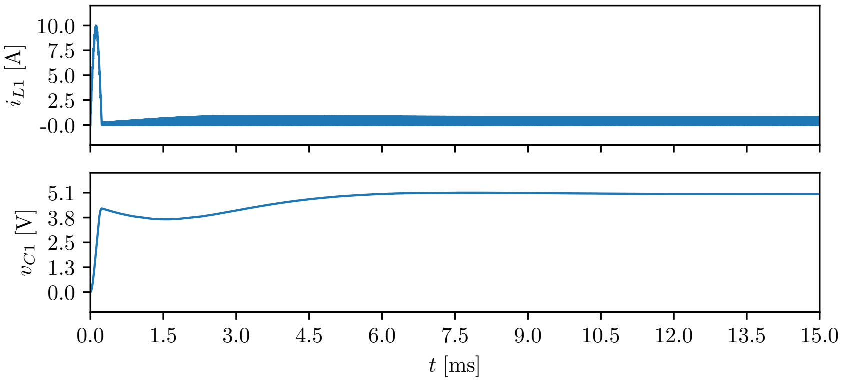


Figure 11: Transient diagrams of state variables for buck converter depicted in Fig. 2 for the first 15 ms.

3. **Standard Ćuk converter in open loop:** `examples\cuk`

Ćuk converter depicted in Fig. 3 is simulated in the open loop with circuit parameters $v_{IN} = 5$ V, $L_1 = 645.4\,\mu$H, $L_2 = 996.3\,\mu$H, $C_1 = 217$ nF, $C_2 = 14.085\,\mu$F and $R = 43\,\Omega$. First simulation is done in the open loop with PWM driven switch S, where switching frequency $f_S = 30$ kHz and duty-ratio $d = 0.2$. Simulation results are depicted in Fig. 12, where is visible that for some intervals of time Ćuk converter operates in DICM.

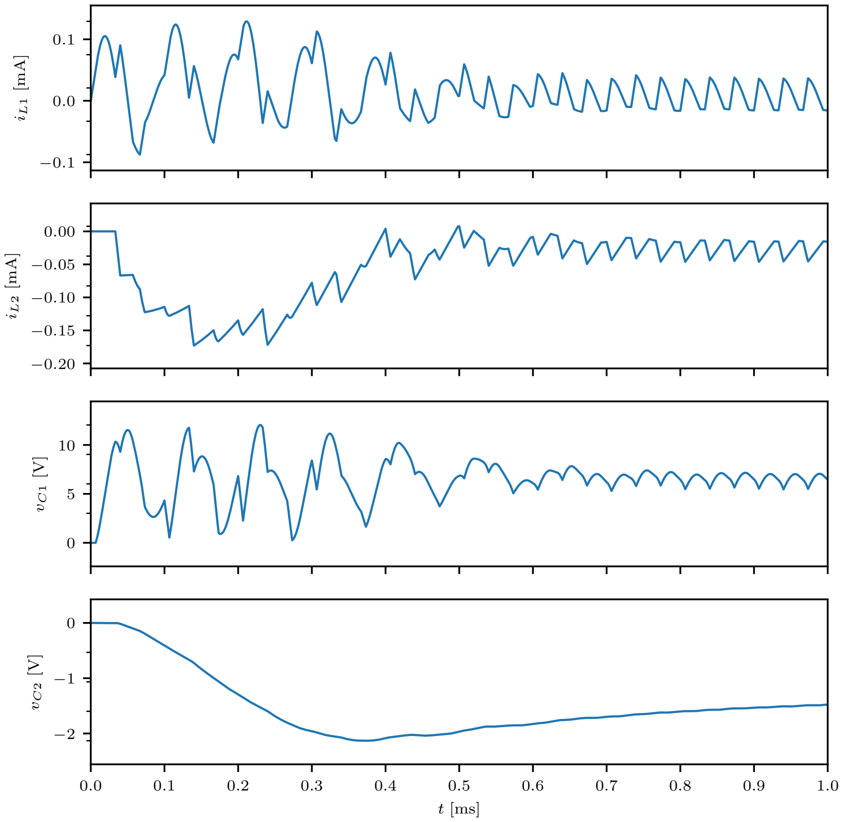


Figure 12: Transient diagrams of state variables for Ćuk converter depicted in Fig. 3 in the open loop with PWM signal, whose frequency is $f_S = 30$ kHz and duty-ratio is $d = 0.2$, for the first 1 ms.

Second simulation is done in the open loop with PWM switching frequency being again $f_S = 30$ kHz, but now duty-ratio is $d = 0.8$. From the results from Fig. 13 can be seen that converter operates in DCVM.

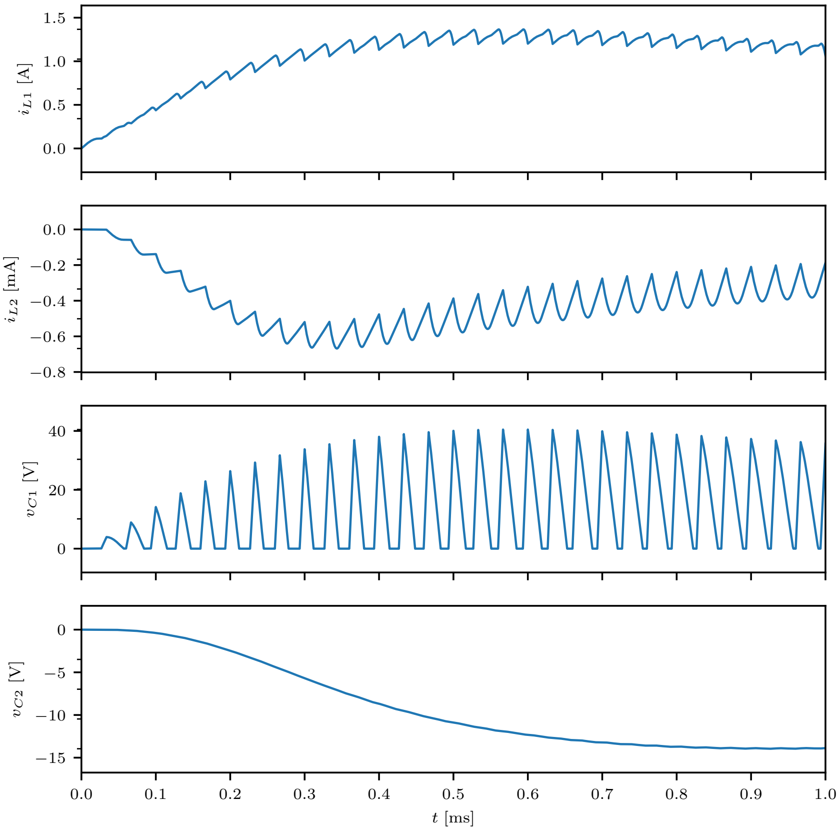


Figure 13: Transient diagrams of state variables for Ćuk converter depicted in Fig. 3 in the open loop with PWM signal, whose frequency is $f_S = 30$ kHz and duty-ratio is $d = 0.8$, for the first 1 ms.

4. **Boost converter in open loop configuration and controlled using NMPC:** `examples\boost`

   Simulated boost converter is depicted in Fig. 14. Chosen circuit parameters are $v_{IN} = 10$ V, $L = 100$ μH, $C = 20$ μF, $R = 4$ Ω, and PWM signal has switching frequency $f_S = 50$ kHz.

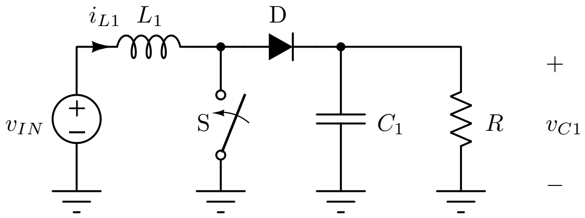


Figure 14: Boost converter.

Boost converter is simulated in open loop and using NMPC method both for desired duty-ratio $d = 0.3$. In the open loop obtained transient diagrams are depicted in Fig. 15. The same converter is simulated when the NMPC control method was applied for the same desired duty-ratio. NMPC simulation diagrams are depicted in Fig. 16.

For the simulation, horizon and matrices $\mathbf{P}$, $\mathbf{Q}$ and $\mathbf{R}$ can be changed inside the file `Controllers\simulate_nmpc.py`. For the boost example, matrix $\mathbf{P}$ is determined using MATLAB code in the folder `examples\boost\nmpc_d_0.3\symbolic.m`.

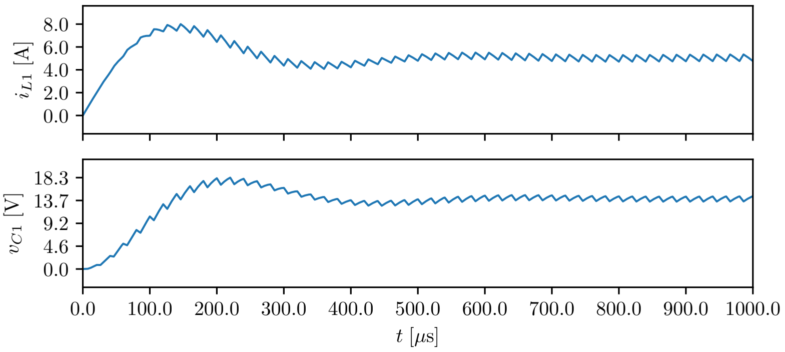


Figure 15: Transient diagrams of state variables for boost converter depicted in Fig. 14 in the open loop with PWM signal, whose frequency is $f_S = 50$ kHz and duty-ratio is $d = 0.3$, for the first 1 ms.

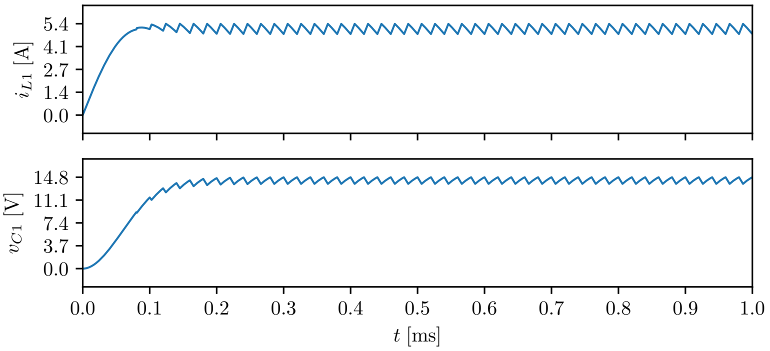


Figure 16: Transient diagrams of state variables for boost converter depicted in Fig. 14 controlled using NMPC for desired duty-ratio is $d = 0.3$, for the first 1 ms.

References

[1] D. Maksimović, "Synthesis of pwm and quasi-resonant dc-to-dc power converters," Ph.D. dissertation, California Institute of Technology, 1989.

[2] C.-W. Ho, A. Ruehli, and P. Brennan, "The modified nodal approach to network analysis," *IEEE Transactions on circuits and systems*, vol. 22, no. 6, pp. 504–509, 1975.

[3] S. Ćuk, "Modelling, analysis, and design of switching converters," Ph.D. dissertation, California Institute of Technology, 1977.

[4] L. Stella, A. Themelis, P. Sopasakis, and P. Patrinos, "A simple and efficient algorithm for nonlinear model predictive control," in *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on.* IEEE, 2017, pp. 1939–1944.

[5] A. Prodić, D. Maksimović, and R. W. Erickson, "Design and implementation of a digital pwm controller for a high-frequency switching dc-dc power converter," in *Industrial Electronics Society, 2001. IECON'01. The 27th Annual Conference of the IEEE*, vol. 2. IEEE, 2001, pp. 893–898.

[6] R. W. Erickson and D. Maksimović, *Fundamentals of power electronics.* Springer Science & Business Media, 2001.

[7] K. Schenk and S. Cuk, "Small signal analysis of converters with multiple discontinuous conduction modes," in *Power Electronics Specialists Conference, 1998. PESC 98 Record. 29th Annual IEEE*, vol. 1. IEEE, 1998, pp. 623–629.

[8] D. Maksimović and S. Ćuk, "A unified analysis of pwm converters in discontinuous modes," *IEEE Transactions on Power Electronics*, vol. 6, no. 3, pp. 476–490, 1991.

[9] M. Zhu, F. L. Luo, and Y. He, "Remaining inductor current phenomena of complex dc–dc converters in discontinuous conduction mode: General concepts and case study," *IEEE Transactions on Power Electronics*, vol. 23, no. 2, pp. 1014–1019, 2008.

[10] R. Tymerski and V. Vorperian, "Generation, classification and analysis of switched-mode dc-to-dc converters by the use of converter cells," in *Telecommunications Energy Conference, 1986. INTELEC'86. International.* IEEE, 1986, pp. 181–195.

[11] A. Lekić, "Simulation of pwm dc-dc converters using eigenvalues and eigenvectors," *Journal of Electrical Engineering*, vol. 68, no. 1, pp. 13–22, 2017.

[12] "Nmpc-codegen simulator," https://github.com/kul-forbes/nmpc-codegen, accessed: 2018-08-24.

[13] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "Casadi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, pp. 1–36, 2018.

[14] D. Maksimović and S. Ćuk, "A general approach to synthesis and analysis of quasi-resonant converters," *IEEE Transactions on Power Electronics*, vol. 6, no. 1, pp. 127–140, 1991.

[15] D. Li, R. Tymerski, and T. Ninomiya, "Pecs-an efficient solution for simulating switched networks with nonlinear elements," *IEEE Transactions on industrial Electronics*, vol. 48, no. 2, pp. 367–376, 2001.