Lyapunov-based Proportional-Integral Controller Design with Guaranteed Region of Convergence for dc-dc Power Converters

Mohsen Vatani and Morten Hovd
Engineering Cybernetics Department
Norwegian University of Science and Technology
N-7491 Trondheim, Norway
Email: {mohsen.vatani,morten.hovd}@itk.ntnu.no

***Abstract*—In this paper, Lyapunov stability of dc-dc power converters controlled with proportional-integral (PI) controllers is investigated. The class of dc-dc converters with discrete-time bilinear averaged dynamic model is considered. An integral action on the output voltage is introduced for robust output regulation in case of model parameter uncertainties and external disturbances. An algorithmic method, using sum of squares programming, is introduced to calculate a candidate Lyapunov function, and the coefficients of P and PI controllers are calculated to guarantee a decrease in the Lyapunov function in each time step while input constraints are fulfilled. The controller design method is iterated in order to improve the region of convergence of the state trajectories. The resulting controller designs are evaluated through simulations of detailed switching models for a boost and a cuk converter.**

I. Introduction

Dc-dc power converters, such as buck, boost, and cuk converters, have been widely used in renewable energy applications. Variations in dc voltages generated by renewable cells and unpredictable load transients necessitate the need for a reliable dc-dc conversion stage in these applications in order to provide a stable and regulated output dc voltage. It is a well known fact that the state space averaged model of these class of dc-dc power converters are described by bilinear models which include the product of the duty cycle and states. In addition, in the case of boost converter as an example, the converter is modeled as a non-minimum phase system with a right-half-plane zero in the output to be controlled. Therefore, the problem of output voltage regulation of these type of converters provides challenging theoretical case studies in addition to their practical advantages; bringing this problem to the attention of many control engineerings as well as power electronic engineers.

The proportional-integral (PI) controllers are the most popular control strategies in industry due to their simplicity, ease of design, and low cost. The basic idea of using integral action in the controller is to induce robust output regulation in the case of model parameter uncertainties and external disturbances. The design principle of PI controllers is based on the linearized small-signal model of the system around the operational point. In the case of controller design for dc-dc converters, this linearization results in neglecting the bilinear terms which are the main source of system's nonlinearity. Therefore, these designs are not suitable for larger disturbances, such as large load variations, as the stability and performance of the system are not analyzed for the initial points distant from the steady-state operating point.

Sum Of Squares (SOS) programming is a technique for proving non-negativity of polynomial functions, and can be used in implementation of different system analysis methods. The main idea of using SOS in the stability analysis of the discrete-time systems is to prove that the Lyapunov function decrement in each sampling time is a SOS, i.e., positive. In this case, by replacing the non-negativity conditions with SOS conditions, not only testing the derivative of the Lyapunov function but also constructing the Lyapunov function can be performed by SOS techniques. This is an important improvement in system analysis tools as it provides an algorithmic procedure to build a candidate Lyapunov function.

The SOS technique can be classified as a generalization of linear matrix inequality (LMI) methods. To solve LMI-based analysis and synthesis problems, semi-definite programming (SDP) techniques are exploited where the complexity is worst-case polynomial in time. The SOS technique uses the same algorithm by transforming the polynomial representation into a quadratic one. Therefore, the non-negativity test can be formulated as an LMI. Considering the fact that checking non-negativity of a polynomial, when the polynomial degree is at least 4, is an NP-hard problem, the SOS provides an alternative solution for proving the non-negativity of a polynomial with polynomial complexity in time.

The transformation from the polynomial representation to the corresponding SDP representation can be manually performed for specific problems. However, such a conversion can be cumbersome in general. In order to facilitate this transformation, a few software products have been developed, e.g., SOSTOOLS and YALMIP. They automate the transformation of SOS problems to SDP form, call the SDP solver, and transform the SDP solution back to the solution of the original SOS program.

The applications of SOS programming methods in power engineering have been recently reported in the literature [1]–[3] where SOS-based techniques for the algorithmic construction of Lyapunov functions for the transient stability analysis of power systems are introduced. Reference [4] investigates the stabilization of the discrete-time bilinear systems by using SOS decomposition method. The input is defined as the ratio of polynomials and a quadratic Lyapunov function is used to prove the stability. This method is used in [5] to investigate the stabilization of dc-dc converters, however, the rational polynomial structure of the controller makes the implementation and structure of the system complicated.

In this paper, the Lyapunov-based design of PI controllers for the class of dc-dc power converters described by discrete-time bilinear models is considered. By using the proposed method, not only an algorithmic design process for Lyapunov stable PI controllers is proposed but also a guaranteed region of convergence for the state trajectories is calculated. The proposed controller in the PI form not only simplifies the controller structure but also provides robustness in the output voltage to parameter uncertainties and disturbances.

In the following, in Section II, the general averaged discrete-time bilinear model of dc-dc converters with PI controller is developed and a coordinate transformation is performed to transform the equilibrium point of the model to the origin. Section III presents the controller design process using SOS programming. In Section IV, performance of the proposed control strategy for dc-dc converters is evaluated. Finally, Section V concludes this paper.

II. Bilinear Averaged Model of the Converter

A dc-dc power converter with a single semiconductor switch can be modeled as a switched system with a specific model for each switching state as

$$\begin{aligned} \text{switch status } on\text{: } \dot{\mathbf{x}}(t) &= A_1\mathbf{x}(t) + B_1\mathbf{v}, \\ \text{switch status } off\text{: } \dot{\mathbf{x}}(t) &= A_2\mathbf{x}(t) + B_2\mathbf{v}, \end{aligned} \tag{1}$$

where $\mathbf{x}$ represents states including capacitor voltages and inductor currents and vector $\mathbf{v}$ represents source voltages and diode voltages. A Pulse Width Modulation (PWM) signal with switching frequency $f_s = 1/T_s$ controls the on/off status of the converter switches. The duty cycle d is defined as the ratio of t_{on}/T_s where t_{on} is the time period in which the switching state remains in *on* position. Consequently, t_{off} is equal to $(1-d)T_s$. Assuming that the inductor current is not saturated and by considering the duty cycle definition, the continuous-time average model of the converter is formulated as

$$\dot{\mathbf{x}} = (d(t)A_1 + (1-d(t))A_2)\mathbf{x} + (d(t)B_1 + (1-d(t))B_2)\mathbf{v},$$

which can be simplified and reformulated as

$$\dot{\mathbf{x}} = A_2\mathbf{x} + (A_1 - A_2)\mathbf{x}d(t) + (B_1\mathbf{v} - B_2\mathbf{v})d(t) + B_2\mathbf{v}.$$

Assuming a sampling period of T_s, the discrete-time average model of the converter, based on a forward Euler approximation, becomes

$$\begin{aligned} \mathbf{x}_{k+1} = &\underbrace{(T_sA_2 + I)}_{A_d}\mathbf{x}_k + \underbrace{T_s(A_1 - A_2)}_{B_{db}}\mathbf{x}_k d_k \\ &+ \underbrace{T_s(B_1\mathbf{v} - B_2\mathbf{v})}_{B_d} d_k + \underbrace{T_sB_2\mathbf{v}}_{d_d}. \end{aligned} \tag{2}$$

Equation (2) is in the form of a standard bilinear discrete-time system as

$$\mathbf{x}_{k+1} = A_d\mathbf{x}(t) + B_{db}\mathbf{x}_k u_k + B_d u_k + d_d, \tag{3}$$

where $u_k = d_k$ is considered as the input of the system. In general, for converters with more than one switch, the averaged discrete-time bilinear model of the converter is represented by

$$\mathbf{x}_{k+1} = A_d\mathbf{x}_k + \sum_{i=1}^{m}(B_{db,i}\mathbf{x}_k + B_{d,i})u_{i,k} + d_d, \tag{4}$$

where $u_{i,k} = d_{i,k}$ is the duty cycle of the ith switch and m is the number of switches.

The desired equilibrium operating point of the bilinear model of the converter is nonzero. By defining the desired equilibrium state vector and input as $\mathbf{x}^{ss}$ and $\mathbf{u}^{ss}$, respectively, the coordinate transformation to bring the equilibrium point to the origin is defined as

$$\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}^{ss} \quad , \quad \tilde{\mathbf{u}} = \mathbf{u} - \mathbf{u}^{ss}. \tag{5}$$

Substituting for the state variables and input from (5) in (4) yields

$$\begin{aligned} \tilde{\mathbf{x}}_{k+1} + \mathbf{x}^{ss} = &A_d(\tilde{\mathbf{x}}_k + \mathbf{x}^{ss}) \\ &+ \sum_{i=1}^{m}(B_{db,i}(\tilde{\mathbf{x}}_k + \mathbf{x}^{ss}) + B_{d,i})(\tilde{u}_{i,k} + u_i^{ss}) + d_d. \end{aligned} \tag{6}$$

Equation (6) can be decomposed into two equations. The first equation represents the relation between the desired equilibrium operating point and equilibrium input as

$$\mathbf{x}^{ss} = A_d\mathbf{x}^{ss} + \sum_{i=1}^{m}(B_{db,i}\mathbf{x}^{ss} + B_{d,i})u_i^{ss} + d_d, \tag{7}$$

while the second equation represents the dynamic of the converter with its equilibrium point at the origin as

$$\tilde{\mathbf{x}}_{k+1} = \underbrace{(A_d + \sum_{i=1}^{m}B_{db,i}u_i^{ss})}_{A}\tilde{\mathbf{x}}_k + \sum_{i=1}^{m}(\underbrace{B_{db,i}}_{B_{b,i}}\tilde{\mathbf{x}}_k + \underbrace{B_{db,i}\mathbf{x}^{ss} + B_{d,i}}_{B_i})\tilde{u}_{i,k},$$

which is in the form of a standard discrete-time bilinear system as

$$\tilde{\mathbf{x}}_{k+1} = A\tilde{\mathbf{x}}_k + \sum_{i=1}^{m}(B_{b,i}\tilde{\mathbf{x}}_k + B_i)\tilde{u}_{i,k}. \tag{8}$$

with m showing the number of inputs. For the sake of simplicity, the bilinear system dynamics in (8) is expressed as

$$\tilde{\mathbf{x}}_{k+1} = A\tilde{\mathbf{x}}_k + (B_{\tilde{\mathbf{x}}} + B)\tilde{\mathbf{u}}_k, \tag{9}$$

where $B_{\tilde{\mathbf{x}}} = \left[\, B_{b,1}\tilde{\mathbf{x}}_k \; B_{b,2}\tilde{\mathbf{x}}_k \; \cdots \; B_{b,m}\tilde{\mathbf{x}}_k \,\right]$, and $B = \left[\, B_1 \; B_2 \; \cdots \; B_m \,\right]$.

The control block diagram of a dc-dc power converter is shown in Fig.1. A necessity for the control systems of dc-dc converters is their ability to track their reference output voltage v_o^{ref}, without any steady state error, even in the presence of disturbances. Therefore, to robustly bring the tracking error of output voltage to zero, an integral action should be placed in the control loop of output voltage. This means that output voltage is controlled through a PI controller while only proportional controllers are used in the feedback loop of other states.

The set of state-space equations for integrating controller with respect to Fig.1 is as follows

$$\tilde{\mathbf{x}}_{j,k} = \mathbf{c}_j \tilde{\mathbf{x}}_k, \tag{10a}$$
$$\tilde{\mathbf{x}}_k^{int} = \tilde{\mathbf{x}}_{k-1}^{int} + \tilde{\mathbf{x}}_{j,k}, \tag{10b}$$
$$\tilde{\mathbf{u}}_{j,k} = \mathbf{k}_{p,i}\tilde{\mathbf{x}}_{j,k} + \mathbf{k}_i \tilde{\mathbf{x}}_k^{int}. \tag{10c}$$

where vector $\mathbf{c}_j$ determines the linear combination of states that should be integrated, $\mathbf{k}_{p,i}$ is the vector of proportional control gains for integrated states, and $\mathbf{k}_i$ is the integrator control gain. Calculating (10c) at instant $(k-1)$ and subtracting it from (10c) at instant k results in

$$\tilde{\mathbf{u}}_{j,k} - \tilde{\mathbf{u}}_{j,k-1} = \mathbf{k}_{p,i}(\tilde{\mathbf{x}}_{j,k} - \tilde{\mathbf{x}}_{j,k-1}) + \mathbf{k}_i(\tilde{\mathbf{x}}_k^{int} - \tilde{\mathbf{x}}_{k-1}^{int}) \tag{11}$$

Substituting from (10b) in (11), the input dynamic is calculated as follows

$$\tilde{\mathbf{u}}_{j,k} = \tilde{\mathbf{u}}_{j,k-1} + \mathbf{k}_{p,i}(\tilde{\mathbf{x}}_{j,k} - \tilde{\mathbf{x}}_{j,k-1}) + \mathbf{k}_i \tilde{\mathbf{x}}_{j,k} \tag{12}$$

which is in the form of the time-response of a discrete-time PI controller, and therefore, it is confirmed that the set of state-space equations in (10) describes a PI controller. Converting (12) to frequency domain results in

$$\frac{U_j}{X_j} = \mathbf{k}_{p,i} + \frac{\mathbf{k}_i}{1 - z^{-1}},$$

which is the form of a discrete-time PI controller with a backward Euler approximation.

Considering the feedback from non-integrating states $\mathbf{x}_n$ with the proportional controller in the control loop, the control input $\tilde{u}$ is calculated as

$$\tilde{\mathbf{u}}_k = \mathbf{k}_{p,n}\tilde{\mathbf{x}}_{n,k} + \mathbf{k}_{p,i}\tilde{\mathbf{x}}_{j,k} + \mathbf{k}_i \tilde{\mathbf{x}}_k^{int} \tag{13}$$

where $\mathbf{k}_{p,n}$ is the vector of proportional control gains for non-integrated states.

Based on (9), (10b), and (13), the dynamic of the converter combined with P&PI controllers can be described as

$$\begin{bmatrix} \tilde{\mathbf{x}} \\ \tilde{\mathbf{x}}^{int} \end{bmatrix}_{k+1} = \underbrace{\begin{bmatrix} A & 0 \\ 0 & I \end{bmatrix}}_{A_t} \begin{bmatrix} \tilde{\mathbf{x}} \\ \tilde{\mathbf{x}}^{int} \end{bmatrix}_{k} + \underbrace{\begin{bmatrix} 0 & 0 \\ \mathbf{c}_j & 0 \end{bmatrix}}_{C_t} \begin{bmatrix} \tilde{\mathbf{x}} \\ \tilde{\mathbf{x}}^{int} \end{bmatrix}_{k+1} + \left(\underbrace{\begin{bmatrix} B_{\tilde{\mathbf{x}}} \\ 0 \end{bmatrix}}_{B_{\tilde{\mathbf{x}},t}} + \underbrace{\begin{bmatrix} B \\ 0 \end{bmatrix}}_{B_t} \right) \underbrace{\underbrace{\begin{bmatrix} \mathbf{k}_{p,n} & \mathbf{k}_{p,i} & \mathbf{k}_i \end{bmatrix}}_{\mathbf{k}} \begin{bmatrix} \tilde{\mathbf{x}} \\ \tilde{\mathbf{x}}^{int} \end{bmatrix}_{k}}_{\tilde{\mathbf{u}}_k} \tag{14}$$

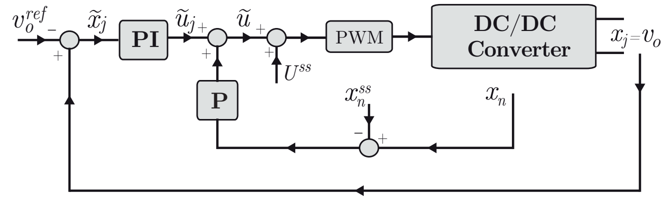

Figure 1. Control block diagram.

assuming $\tilde{\mathbf{x}}_k = \begin{bmatrix} \tilde{\mathbf{x}}_{n,k} & \tilde{\mathbf{x}}_{j,k} \end{bmatrix}^T$. By defining

$$C = I - \begin{bmatrix} 0 & 0 \\ \mathbf{c}_j & 0 \end{bmatrix},$$

equation (14) can be simplified as

$$\begin{bmatrix} \tilde{\mathbf{x}} \\ \tilde{\mathbf{x}}^{int} \end{bmatrix}_{k+1} = C^{-1}A_t \begin{bmatrix} \tilde{\mathbf{x}} \\ \tilde{\mathbf{x}}^{int} \end{bmatrix}_{k} + C^{-1}(B_{\tilde{\mathbf{x}},t} + B_t)\tilde{\mathbf{u}}_k$$

which is again in the form of a standard discrete-time bilinear system as

$$\hat{\mathbf{x}}_{k+1} = \hat{A}\hat{\mathbf{x}}_k + (\hat{B}_{\hat{\mathbf{x}}} + \hat{B})\hat{\mathbf{u}}_k \quad , \quad \hat{\mathbf{u}}_k = \mathbf{k}\hat{\mathbf{x}}_k \tag{15}$$

III. Controller design based on the SOS

This section presents a modified version of the SOS-based controller design procedure proposed in [4], which is adjusted in order to calculate the coefficients of the P&PI controller that maximizes the region of convergence for a discrete-time bilinear system (15) with equilibrium point in the origin.

The main idea here, to design a Lyapunov-based stable P&PI controller, is to guarantee that a candidate Lyapunov function is decreased in each time step when the system is controlled by the proposed controller. A quadratic Lyapunov function $V_k = \hat{\mathbf{x}}_k^T P \hat{\mathbf{x}}_k$ is used in this paper for some given weighting matrix $P > 0$. Consequently, the closed loop stability is guaranteed by ensuring that the candidate quadratic Lyapunov function is decreasing in each time step:

$$V(\hat{\mathbf{x}}_k) - V(\hat{\mathbf{x}}_{k+1}) = \hat{\mathbf{x}}_k^T P \hat{\mathbf{x}}_k - \hat{\mathbf{x}}_{k+1}^T P \hat{\mathbf{x}}_{k+1} > 0. \tag{16}$$

A. Introduction on sum of squares programming

To evaluate the positiveness in inequality (16), SOS programming is exploited in the present work. The basic idea behind the SOS approach for checking the non-negativity of a polynomial $p(x)$, is to replace the non-negativity with the condition that the polynomial can be expressed as an SOS [6]:

Definition 1. *For $x \in \mathbb{R}^n$, a polynomial $p(x)$ is an SOS if there exist some polynomials $f_j(x)$, $j = \{1, 2, \cdots m\}$, such that*

$$p(x) = \sum_{j=1}^{m} f_j^2(x).$$

It is shown in [6] that the problem of determining whether a polynomial is SOS can be reduced to the problem of feasibility of a Semi-definite Program (SDP) which can be efficiently solved in polynomial time. The main proposition to relate the SOS with SDP is as follows [6]:

Proposition 2. *A polynomial $p(x)$ of degree $2d$ is an SOS if and only if there exist a positive semi-definite matrix Q (Gram matrix) such that $p(x) = z^T Qz$, where z is the vector of monomials of degree up to d, i.e.,*

$$z = [1, x_1, x_2, \cdots, x_n, x_1^2, x_1x_2, \cdots, x_n^d]^T$$

Given a polynomial $p(x)$, by expanding $z^T Qz$ and matching the coefficients of the resulting monomials to the ones in $p(x)$, linear constraints on the entries of Q is obtained. The $p(x)$ being SOS is equivalent to Q be a positive semi-definite matrix. Therefore, the problem of finding whether $p(x)$ is an SOS can be set as an SDP to find positive definite matrix Q that satisfies the mentioned linear constraints. The conversion step of going from an SOS decomposition problem to an SDP problem is fully algorithmic and can be computed with the help of available software, e.g., YALMIP [7].

An interesting consequence of Proposition 2 is the case in which the monomials in the polynomial $p(x)$ have unknown coefficients. The problem here is to search for feasible coefficients that make $p(x)$ nonnegative. Affine constraints relate the unknown coefficients of $p(x)$ to the elements of Q, therefore, it is obvious that the search for the coefficients that make $p(x)$ an SOS can also be formulated as an SDP with the coefficients as decision variables. This observation is very useful in the problem of constructing Lyapunov functions.

If the test of the SOS decomposition for $p(x)$ is desired to be performed only on a restricted domain, rather than globally, then the Positivestellensatz can be exploited. It is shown that $p(x) \geq 0$ is satisfied $\forall x$ with $\theta(x) \leq 0$ if SOS polynomial $\beta(x)$ can be found such that

$$p(x) + \beta(x)\theta(x) \quad \text{is an SOS}$$

See [6] for a more general version of the Positivestellensatz for SOS.

The definitions of positive semidefiniteness and SOS of scalar polynomials can be extended to polynomial matrices, i.e., matrices with polynomials as elements. A symmetric polynomial matrix $P(x)$ with the dimension $m \times m$ is a positive semidefinite polynomial matrix if $P(x)$ is a positive semidefinite matrix $\forall x \in \mathbb{R}^n$, i.e., point-wise positive semidefinite [8].

The following lemma allows to easily determine whether a given polynomial matrix is an SOS matrix using the SDP techniques mentioned above. See [9] for the proof.

Lemma 3. *A polynomial matrix $P(x)$ with dimension $m \times m$ and $x \in \mathbb{R}^n$ is an SOS polynomial matrix if and only if the scalar polynomial $y^T P(x)y$ is an SOS.*

B. Controller Design

This section provides the controller design procedures, using controllers on the form (15), to stabilize the system to the origin. It is assumed that the controller is obliged to satisfy the control constraints of the form

$$|\hat{u}_i(x)| \leq \hat{u}_{i,max}. \tag{17}$$

where the region of convergence is determined as $(\gamma - \hat{\mathbf{x}}_k^T P \hat{\mathbf{x}}_k) > 0$.

To fulfill (16) within all time steps, and for all the initial points in the region of convergence determined as $(\gamma - \hat{\mathbf{x}}_k^T P \hat{\mathbf{x}}_k) > 0$, the coefficients of P&PI controller in vector $\mathbf{k}$ in (15) is calculated based on the following theorem. All proofs of following theorems can be found in [4].

Theorem 4. *Region of convergence: Given a quadratic function $V(\hat{\mathbf{x}}) = \hat{\mathbf{x}}_k^T P \hat{\mathbf{x}}_k$, the vector $\boldsymbol{k}$, and the SOS polynomial $s_1(\tilde{\mathbf{x}}_k, z)$, a bilinear discrete-time system with the control law (15) is closed-loop stable $\forall \hat{\mathbf{x}}_k | \hat{\mathbf{x}}_k^T P \hat{\mathbf{x}}_k < \gamma$, provided that*

$$\begin{bmatrix} \hat{\mathbf{x}}_k \\ z \end{bmatrix}^T M(\hat{\mathbf{x}}) \begin{bmatrix} \hat{\mathbf{x}}_k \\ z \end{bmatrix} - s_1(\hat{\mathbf{x}}_k, z)(\gamma - \hat{\mathbf{x}}_k^T P \hat{\mathbf{x}}_k) > 0, \tag{18}$$

where

$$M(\hat{\mathbf{x}}_k) = \begin{bmatrix} P & (\hat{A} + (\hat{B}_{\hat{\mathbf{x}}} + \hat{B})\boldsymbol{k})^T P \\ P(\hat{A} + (\hat{B}_{\hat{\mathbf{x}}} + \hat{B})\boldsymbol{k}) & P \end{bmatrix}. \tag{19}$$

Theorem 5. *Given the $\boldsymbol{k}$, SOS polynomial $q_i(\hat{\mathbf{x}}_k)$, the input constraint in (17) is satisfied $\forall\, \hat{\mathbf{x}}_k | \hat{\mathbf{x}}_k^T P \hat{\mathbf{x}}_k < \gamma$ provided*

$$\begin{bmatrix} \hat{u}_{max,i}^2 - q_i(\hat{\mathbf{x}}_k)(\gamma - \hat{\mathbf{x}}_k^T P \hat{\mathbf{x}}_k) & \boldsymbol{k}\hat{\mathbf{x}}_k \\ \boldsymbol{k}\hat{\mathbf{x}}_k & 1 \end{bmatrix} > 0. \tag{20}$$

C. Improving region of convergence

For a given Lyapunov function weighting matrix P, Theorems 4 and 5 allow for the controller design according to

$$\begin{aligned} &\max_{\mathbf{k}, s_1(\hat{\mathbf{x}}_k, z), q_i(\hat{\mathbf{x}}_k)} \gamma \\ &\quad \text{such that (18) and (20) hold} \\ &\qquad s_1(\hat{\mathbf{x}}_k, \mathbf{z}), q_i(\hat{\mathbf{x}}_k) \text{ are SOS} \\ &\qquad P > 0, \text{ trace}(P) = \text{constant}. \end{aligned} \tag{21}$$

The coefficients in $\mathbf{k}$ and $s_1(\hat{\mathbf{x}}_k, \mathbf{z})$ enter linearly in (18). Thus, for given γ and P, (18) can be verified with the polynomial coefficients as free variables. This corresponds to finding a feasible point for a constrained semi-definite programming problem, and can be easily formulated and solved using readily available software such as YALMIP (with an appropriate semi-definite programming solver). As a result of Theorem 4, the calculated control input (15) stabilizes the system within the region defined by $\hat{\mathbf{x}}^T P \hat{\mathbf{x}} < \gamma$.

Consequently, to solve this bilinear optimization problem, it is advantageous to iteratively fix some variables and solve for the other variables, which is a common approach to solving bilinear SOS. Therefore, the following optimization problems are solved iteratively until a sufficiently large region of convergence is obtained :

1) For a given P, verify the constraints of (21) with the coefficients of the polynomials $\mathbf{k}$, $s_1(x, z)$, and $q_i(x)$ as free variables, for iteratively increasing values of γ.
2) For given polynomials $\mathbf{k}$, $s_1(x, z)$, and $q_i(x)$, solve (21) with P as the free variable.

For the initialization of the optimization procedure, considering the fact that close to the origin the system dynamics are essentially linear, it is possible to initialize P as the solution of the Riccati equation for LQ optimal control of the linear sub-system (with weights on states and inputs reflecting the designer's performance criteria).

IV. Case studies

In this section, to validate the effectiveness of the proposed control method, the detailed switching model of two different dc-dc converters, a boost and a cuk converter, is simulated in PLECS/Simulink. The controller is designed using the MATLAB/YALMIP software [10]. The parameters of the study systems are listed in Table I.

A. Case study 1: boost converter

The circuit diagram of a dc-dc boost converter is shown in Fig. 2. The states of the system are considered as the inductor current $x_1 = i_L$ and the capacitor voltage $x_2 = v_C$. The load voltage v_o is considered as the output which should be kept at the desired voltage v_o^{ref}. During the on state, the status of the switch is $S = 1$ for the period of t_{on} and the system matrices with respect to (1) is as follows

$$A_1 = \begin{bmatrix} -\frac{R_1}{L_1} & 0 \\ 0 & -\frac{1}{R_o C_1} \end{bmatrix}, \quad B_1 = \begin{bmatrix} \frac{1}{L_1} \\ 0 \end{bmatrix}, \quad v = E,$$

while during the off state, the status of the switch is $S = 0$ for the period of $t_{off} = T_s - t_{on}$ and the system matrices with respect to (1) is as follows

$$A_2 = \begin{bmatrix} -\frac{(R_1+R_s)}{L_1} & -\frac{1}{L_1} \\ \frac{1}{C_1} & -\frac{1}{R_o C_1} \end{bmatrix}, \quad B_2 = \begin{bmatrix} \frac{1}{L_1} \\ 0 \end{bmatrix}, \quad v = E.$$

The desired output voltage is set to $v_o^{ref} = 24\,\text{V}$, and therefore, based on (7), $i_L^{ss} = 5.4288\,\text{A}$, and $d^{ss} = 0.5579$. The control effort constraint is set to $\hat{u}_{max} = 0.4$. The problem to be solved is the determination of the P&PI controller coefficients which stabilizes the system in the region determined by $\hat{\mathbf{x}}_k^T P \hat{\mathbf{x}}_k < \gamma$. Based on the controller design method presented in Section III-B, the SOS problem is solved iteratively for 28 times and the following controller is obtained:

$$\hat{\mathbf{u}}_k = \mathbf{k}\hat{\mathbf{x}}_k \quad , \quad \mathbf{k} = \begin{bmatrix} -0.1804 & -0.7799 & -0.0194 \end{bmatrix}$$

The simulation results of the designed controllers are shown in Fig. 3. The simulation starts from the initial state at $\mathbf{x}_0 = [1.63, 24.7]^T$. Afterwards, at $t = 1$ ms, the source voltage is decreased to $E = 10\,\text{V}$, and then back to $E = 12\,\text{V}$ at $t = 2$ ms. Thereafter, the load resistor is changed to $R_o = 12\ \Omega$ at $t = 3$ ms and to $R_o = 9\ \Omega$ at $t = 4$ ms. The output voltage of the boost converter, $x_2 = v_o$, is shown in Fig. 3 (a) and the inductor current, $x_1 = i_L$, is shown in Fig. 3 (b). It is shown that the output voltage robustly follows its reference even after parameter mismatch and load changes. Fig. 3 (c) presents the duty cycle of switch S_1 which satisfies the input constraints for designed controllers. Fig. 3 (d) shows the simulated disturbances in source voltage and load resistor. Finally, the region of convergence $\hat{\mathbf{x}}_k^T P \hat{\mathbf{x}}_k \leq \gamma$ for each iteration of controller design process is shown in Fig. 3 (e) which shows how the region of convergence is gradually increased after each iteration.

Table I
Parameters of the case study converters

Parameter	Case 1: boost	Case 2: cuk
R_1	35 $m\Omega$	1 Ω
L_1	100 μH	100 μH
R_2	-	0.5 Ω
L_2	-	100 μH
R_s	0.5 Ω	-
C_1	200 μF	100 μF
C_2	-	10 μF
R_o	10 Ω	15 Ω
E	12 V	30 V
v_o^{ref}	24 V	30 V
T_s	5 μs	10 μs

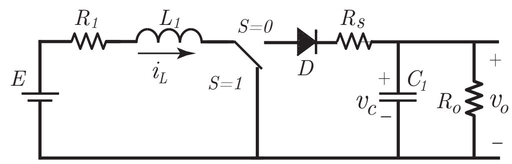

Figure 2. Circuit diagram of the boost converter.

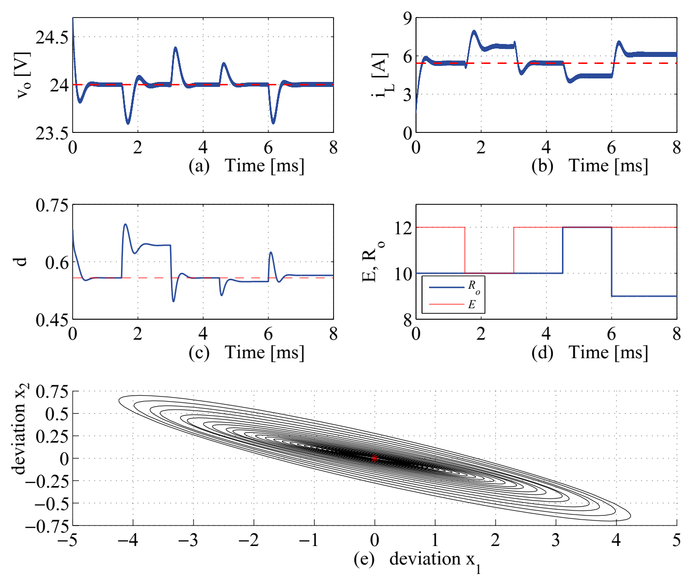

Figure 3. Simulation results for the boost converter: (a) output voltage, (b) inductor current, (c) duty cycle, (d) source voltage and load resistor, and (e) region of convergence.

B. Case study 2: cuk converter

The circuit diagram of a cuk converter is shown in Fig. 4. The state vector is selected as $\mathbf{x} = [i_1\ v_1\ i_2\ v_o]^T$. The duty cycle of the switche S is considered as the input. The system matrices for the *on* and *off* states of the converter are calculated as follows:

$$A_1 = \begin{bmatrix} -\frac{R_1}{L_1} & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{C_1} & 0 \\ 0 & \frac{1}{L_2} & -\frac{R_2}{L_2} & -\frac{1}{L_2} \\ 0 & 0 & \frac{1}{C_2} & -\frac{1}{C_2 R_L} \end{bmatrix}, \quad B_1 = \begin{bmatrix} \frac{1}{L_1} \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} -\frac{R_1}{L_1} & -\frac{1}{L_1} & 0 & 0 \\ \frac{1}{C_1} & 0 & 0 & 0 \\ 0 & 0 & -\frac{R_2}{L_2} & -\frac{1}{L_2} \\ 0 & 0 & \frac{1}{C_2} & -\frac{1}{C_2 R_L} \end{bmatrix}, \; B_2 = \begin{bmatrix} \frac{1}{L_1} \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

The desired output voltage is selected as $v_o^{ref} = 30\,\text{V}$, which, by solving (7), results in the equilibrium input $u_{1,ss} = 0.5275$ and equilibrium state $\mathbf{x}^{ss} = [2.23 \;\; 58.77 \;\; 2 \;\; 30]^T$. The control effort constraint is set to $\hat{u}_{max} = 0.5$. The designed controller after 8 iterations is as follows:

$$\mathbf{k} = \begin{bmatrix} 0.0041 & -0.0736 & -2.2840 & -1.2210 & -0.3221 \end{bmatrix}$$

The simulation results of the designed P&PI controller are shown in Fig. 5. The initial output voltage is set to $v_o = 30.5\,\text{V}$. The load resistor is changed to $R = 13.5\ \Omega$ at $t = 1\,\text{ms}$ and back to its nominal value at $t = 2\,\text{ms}$. In addition, the source voltage is changed to $E = 27\,\text{V}$ at $t = 3\,\text{ms}$ and to $E = 32\,\text{V}$ at $t = 6\,\text{ms}$. The output voltage of the cuk converter, $x_4 = v_o$, is shown in Fig. 5 (a) where the output voltage follows the reference after load changes and remains almost unchanged after source voltage changes. Fig. 5 (b) shows the first inductor current $x_1 = i_1$ and Fig. 5 (c) represents the first capacitor voltage $x_2 = v_1$. It can be concluded from these figures that the first capacitor acts as a buffer between input and output voltages, therefore, the oscillation in input voltage results in high oscillation of first capacitor while no oscillation is observed in the output. While it is not shown in the figures for lack of space, the oscillations in these two parameters will settle down after a while. Fig. 5 (d) shows the second inductor current $x_3 = i_2$ and Fig. 5 (e) shows the calculated duty cycle d which remains in the defined bound during the simulation. Finally, the changes in load resistor and source voltage is shown in Fig. 5 (f).

V. Conclusion

In this paper, the SOS programming based controller synthesis method for discrete-time bilinear systems developed in [4] is used for the P and PI controllers design for a class of power converters described with discrete-time bilinear averaged dynamic model. First, the general averaged discrete-time model of the converter is developed and a coordinate transformation is performed to transform the equilibrium point of the model to the origin. Then, an integral state is added for robust output voltage regulation of the converter regardless of parameter mismatches and external disturbances. In the next step, after an introduction on SOS programming, an algorithmic method to construct candidate Lyapunov functions is introduced. Using an optimization formulation, the coefficients of P&PI controllers are calculated in a way that a decrease in Lyapunov function in each time step is guaranteed, i.e. the system stability is guaranteed. An iteration procedure is used to improve the region of convergence for the controller. The applicability of the resulting controller designs is verified through simulation of a boost and a cuk converter.

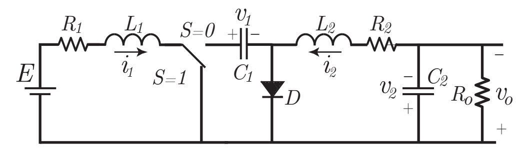

Figure 4. Circuit diagram of the cuk converter.

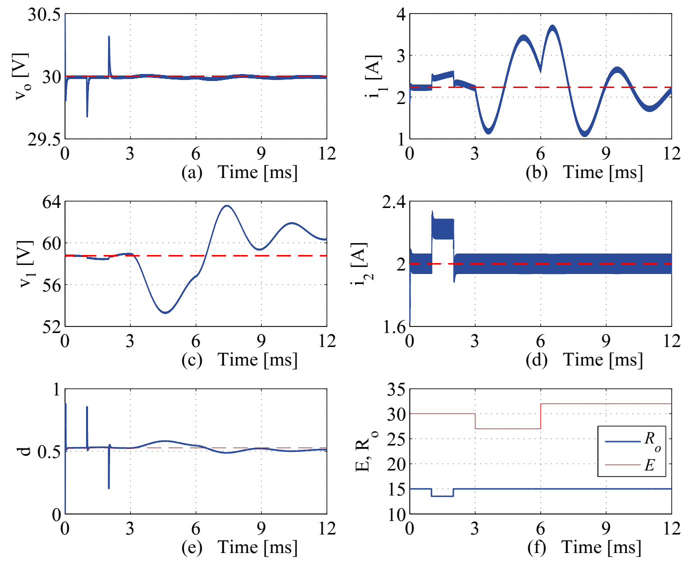

Figure 5. Simulation results for the cuk converter: (a) output voltage, (b) first inductor current, (c) first capacitor voltage, (d) second inductor current, (e) duty cycle, and (f) source voltage and load resistor.

References

[1] M. Anghel, F. Milano, and A. Papachristodoulou, "Algorithmic Construction of Lyapunov Functions for Power System Stability Analysis," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 9, pp. 2533–2546, sep 2013.

[2] S. K. Mazumder and E. P. de la Fuente, "Transient stability analysis of power system using polynomial Lyapunov function based approach," in *IEEE PES General Meeting — Conference & Exposition*. IEEE, jul 2014, pp. 1–5.

[3] ——, "Dynamic stability analysis of power network," in *IEEE Energy Conversion Congress and Exposition (ECCE)*. IEEE, sep 2015, pp. 5808–5815.

[4] M. Vatani, M. Hovd, and S. Olaru, "Control design and analysis for discrete time bilinear systems using Sum of Squares methods," in *53rd IEEE Conference on Decision and Control (CDC)*, dec 2014, pp. 3143–3148.

[5] M. Vatani and M. Hovd, "Control of bilinear power converters using sum of squares programming," in *14th European Control Conference (ECC)*, jul 2015, pp. 654–659.

[6] P. A. Parrilo, "Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization," Ph.D. dissertation, California Institute of Technology, 2000.

[7] J. Löfberg, "Pre- and Post-Processing Sum-of-Squares Programs in Practice," *IEEE Transactions on Automatic Control*, vol. 54, no. 5, pp. 1007–1011, may 2009.

[8] G. Blekherman, P. A. Parrilo, and R. R. Thomas, *Semidefinite Optimization and Convex Algebraic Geometry*. SIAM, 2013.

[9] A. A. Ahmadi and P. A. Parrilo, "A positive definite polynomial Hessian that does not factor," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*. IEEE, dec 2009, pp. 1195–1200.

[10] J. Löfberg, "YALMIP : A Toolbox for Modeling and Optimization in MATLAB," in *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.