# FPGA Implementation of Set-based Model Predictive Control

Renato Babojelić*, Bruno Vilić Belina*, Šandor Ileš*, Jadranko Matuško*.
* University of Zagreb, Faculty of Electrical Engineering and Computing, Zagreb, Croatia
renato.babojelic@fer.hr, bruno.vilic-belina@fer.hr, sandor.iles@fer.hr, jadranko.matusko@fer.hr

***Abstract*—This paper presents an field-programmable gate array (FPGA) implementation of a model predictive controller (MPC) based on one-step controllable ellipsoidal sets. Ellipsoidal sets are precomputed off-line, while the MPC optimization problem is solved in real time in FPGA hardware. For solving the optimization problem, a fast gradient projection method is used. The implementation of a controller is verified in a FPGA-in-the-loop simulation, showing the computational times in microsecond range.**

***Keywords*—model predictive control, fast gradient projection method, field programmable gate array (FPGA)**

## I. INTRODUCTION

The introduction of powerful microcontrollers and ever rising increase in computational power has enabled model predictive control (MPC) to be used in controlling systems with very fast dynamics such as inverters, rectifiers and drives in the last decades [1]–[3].

Initially used as hardware interface logic circuits, owing to inherent parallelism combined with the significant reduction in price and increase in count of logical elements contained on a chip, in the last decade field-programmable gate arrays (FPGA) have seen increased use in control applications [4]. To satisfy strict execution times of fast dynamics systems, FPGAs have seen increased use in implementing model predictive control [5], [6]. FPGAs also proved advantageous in comparison to microcontrollers and DSPs in power electronics applications shortening the computational time, which results in a lower control delay and better dynamic performance [7].

Recently significant advances have been made in the field of numerically efficient predictive control algorithms and it is still the area of intense research activities, both in academia and industry. One of the simplest and widely used approach for efficient solution of model predictive control problem is based on well-known fast (accelerated) gradient projection method (FGM), first introduced by Nesterov in 1983. However, the FGM algorithm can be used for the MPC problems with input constraints only, where the set of input constraints is relatively simple allowing for efficient projection of the candidate solution onto set of input constraints. Various solutions have been suggested to enable FGM to handle state constraints as well. Among them a dual FGM attracted the most of interest.

Usage of control invariant sets is well known technique in control [8], these sets may be approximated by either polyhedral or ellipsoidal sets. While polyhedral sets allow for large regions of attraction their representation as intersections of half-spaces is cumbersome even for the simplest of systems, leading to untractable computational cost. Ellipsoidal approximations of control invariant sets offer smaller regions of attraction, but they can be represented as a single symmetric matrix, leading to much lower computational costs.

In this paper we present an FPGA implementation of predictive controller based on one-step controllable ellipsoidal sets. Ellipsoidal sets are precomputed, while the MPC optimization problem is solved in real time by an FPGA using the FGM algorithm. This approach is proved viable with an design example; to control a grid-tied inverter, set-based predictive controller is synthesized and run on a modern mid-range FPGA device. Implementation is verified in a FPGA-in-the-loop simulation (FIL) in MATLAB/Simulink, and with computation time in microsecond range shown to be suitable for controlling the system in real time.

This paper is organized as follows: in section II we describe the model predictive control algorithm based on ellipsoidal sets; in section III fast gradient method for solving a MPC optimization problem is described; section IV discusses algorithm adaptation and data preparation for implementing in an FPGA; in section V, to prove the viability of the FPGA Set-based MPC algorithm, we show an design example for controlling an grid-tied inverter.

## II. SET-BASED MPC

Assume a discrete time linear dynamical system, with known parameters, in state space representation is given with

$$\begin{aligned} x^+ &= Ax + Bu \\ y &= Cx, \end{aligned} \tag{1}$$

where $x^+$ represents the system state in next discretization step. We say a set $\mathcal{I}$ is **control invariant** for the system (1) if for all $x \in \mathcal{I}$ there exists control action $u \in \mathcal{U}$ such that $Ax + Bu \in \mathcal{I}$ holds. Given such set $\mathcal{I}$ there exists a collection of sets $\{\mathcal{I}_i \mid i \in \mathbb{N}\}$ with

$$\begin{aligned} &\mathcal{I}_0 = \mathcal{I} \\ &\mathcal{I}_i = \{x \mid \exists u \in \mathcal{U}, Ax + Bu \in \mathcal{I}_{i-1}\}. \end{aligned}$$

Sets $\mathcal{I}_i$ contain every state $x$ that can be steered to $\mathcal{I}_{i-1}$ in one step using a single control action. To ease the computational burden for finding such sets we adopt the following ellipsoidal inner approximations of sets $\mathcal{I}_i$ proposed in [9]. As shown in [10], given a stabilizing feedback gain $K$ and an nonempty ellipsoidal control invariant set $\mathcal{E} \subset \mathbb{R}^n$, that is:

$$(A - BK)x \in \mathcal{E},\ \forall x \in \mathcal{E}, \tag{2}$$

there exists the family of ellipsoidal sets $\{\mathcal{E}_i \mid i \in \mathbb{N}\}$ satisfying the recursion

$$\begin{aligned} &\mathcal{E}_0 = \mathcal{E} \\ &\mathcal{E}_i = \textbf{In}(\{x \mid \exists u \in \mathcal{U}, Ax + Bu \in \mathcal{E}_{i-1}\}), \end{aligned} \tag{3}$$

where **In** is the operation finding inner ellipsoidal approximation of a given set, according to a predefined criterion (e.g. maximum volume, trace, etc.).

*1) Ellipsoidal sets computation (offline):* To compute the ellipsoidal sets we first design a stabilizing feedback gain $K$ for a system (1).

To represent an ellipsoidal set we use the usual quadratic constraints: let $\mathcal{P}$ be a square positive semidefinite matrix then $\mathcal{E}_{\mathcal{P}} = \{x \in \mathbb{R}^n \mid x^T \mathcal{P} x \leq 1\}$ is an ellipsoidal set with the center at the origin.

The first, so called terminal, ellipsoidal set $\mathcal{E}_0$ is obtained by solving the optimization problem

$$\begin{aligned} \min_{a \in \mathbb{R}} \quad & a \\ \text{s.t.} \quad & K^T K - a\mathcal{P} \leq 0, \\ & a \geq 0, \end{aligned} \tag{4}$$

where $\mathcal{P}$ is a positive definite matrix such that function $x \mapsto x^T \mathcal{P} x$ is a Lyapunov function for a closed loop system, and defining its positive semidefinite matrix $\mathcal{P}_0$ as

$$\mathcal{P}_0 = \frac{a}{u_{max}^2} \mathcal{P}, \tag{5}$$

where $u_{max}$ is the maximal magnitude of a control signal $u_{err}$, and $a \geq 0$ is the scaling factor.

To continue the recursive construction of ellipsoids $\mathcal{E}_i$, $i \in \{1, \dots n\}$ we use the following equation as proposed in [9]

$$\mathcal{E}_i = \textbf{Proj}_x \left( \textbf{In} \left( \tilde{\mathcal{E}}_{i-1} \cap (\mathbb{R}^n \times \mathcal{E}^{\mathcal{U}}) \right) \right) \tag{6}$$

where $\tilde{\mathcal{E}}_{i-1}$ denotes ellipsoidal sets defined in the extended space $(x\ u)^T \in \mathbb{R}^{n+m}$ as

$$\tilde{\mathcal{E}}_{i-1} = \{(x\ u)^T \mid Ax + Bu \in \mathcal{E}_{i-1}\}, \tag{7}$$

$\mathcal{E}^{\mathcal{U}}$ is ellipsoidal representation of the set of control inputs and $\textbf{Proj}_x$ is projection operator to the state space from the extended space.

To compute the inner ellipsoidal approximation of an intersection of ellipsoids in equation (6) we solve the following optimization problem that maximizes the volume of the ellipsoid $\mathcal{E}_i$

$$\begin{aligned} \max_{Q \in \mathbb{R}^{(n+m)\times(n+m)}} \quad & \log\det Q_{11} \\ \text{s.t.} \quad & 0 < Q \leq P_j^{-1}, \end{aligned} \tag{8}$$

where

$$Q = P_i^{-1} = \begin{pmatrix} Q_{11} & Q_{12} \\ Q_{12}^T & Q_{22} \end{pmatrix} \tag{9}$$

represents the searched inner ellipsoid in the extended space, the block $Q_{11} \in \mathbb{R}^{n\times n}$ represents the said ellipsoid in state space, and $P_j$ are positive semidefinite matrices representing the ellipsoids that need to be intersected in the extended space. Finally to project the found ellipsoid in the extended space to the state space we use the following equality

$$\begin{aligned} \mathcal{E}_i &= \textbf{Proj}_x \left\{ \begin{pmatrix} x \\ u \end{pmatrix}^T \begin{pmatrix} Q_{11} & Q_{12} \\ Q_{12}^T & Q_{22} \end{pmatrix}^{-1} \begin{pmatrix} x \\ u \end{pmatrix} \leq 1 \right\} \\ &= \left\{ x^T Q_{11}^{-1} x \leq 1 \right\}. \end{aligned} \tag{10}$$

*2) Control algorithm (online):* With the ellipsoidal sets computed offline, online MPC algorithm needs to find the smallest ellipsoid that contains the system state an find the optimal control signal to steer it to the next ellipsoid, we outline this procedure in Algorithm 1, where $J$ is some appropriate cost function.

**Algorithm 1:** Set-based Ellipsolidal MPC

**Data:** Elipsoidal sets $\mathcal{E}_0, \dots, \mathcal{E}_n$, state vector $x(k)$ at time instance $k$, system matrices $A$ and $B$.
$k = 0$
**loop**
  Find $i(k) = \min\{i \mid x(k) \in \mathcal{E}_i\}$
  **if** $i(k) == 0$ **then**

$$\begin{aligned} & u(k) = \min\ J(x(k), u(k)) \\ & \text{s.t.} \\ & Ax(k) + Bu(k) \in \mathcal{E}_0 \end{aligned} \tag{11}$$

  **end**
  **else**

$$\begin{aligned} & u(k) = \min\ J(x(k), u(k)) \\ & \text{s.t.} \\ & Ax(k) + Bu(k) \in \mathcal{E}_{i(k)-1} \end{aligned} \tag{12}$$

  **end**
  $k = k + 1$
**end**

## III. Fast gradient projection method for solving the Set-based MPC optimization problem

The Set-based MPC algorithm involves solving a quadratic problem (12) with constraints on the state space variable $x$. As the fast gradient method for solving an MPC problem needs to include only the constraints on the input vector $u$, we have to reformulate the problem (12) to contain only the constraints on the inputs. The reformulated problem is given with

$$u(k) = \min J(x(k), u(k)) \quad \text{s.t. } u(k) \in \mathcal{U}_k(x(k)) \tag{13}$$

where $\mathcal{U}_k(x(k))$ is the set of permissible control actions i.e.:

$$\mathcal{U}_k(x(k)) = \{u(k) \mid Ax(k) + Bu(k) \in \mathcal{E}_{i(k)-1}\}. \tag{14}$$

In other words, given state $x(k)$ the set $\mathcal{U}_k(x(k))$ represents a set of all inputs that will steer the next state $x(k+1)$ to set $\mathcal{E}_{i(k)-1}$. The set $\mathcal{U}_k(x(k))$ can be easily calculated from the set $\bar{\mathcal{E}}_i = \textbf{In}\left(\tilde{\mathcal{E}}_i \cap (\mathbb{R}^n \times \mathcal{E}^{\mathcal{U}})\right)$ (see equation 6) defined in an extended space by fixing $x(k)$. The obtained set is also ellipsoidal and as such represents a set that is simple enough to allow an efficient calculation of the projection operation onto it. In the extended space the ellipsoid $\bar{\mathcal{E}}_i$ is represented with a symmetric positive semidefinite matrix $P_{\text{extended}} \in \mathbb{R}^{8\times 8}$ as:

$$\begin{pmatrix} x \\ u \end{pmatrix}^T P_{\text{extended}} \begin{pmatrix} x \\ u \end{pmatrix} \le 1, \tag{15}$$

which can be written in a block matrix form, with blocks $P_1 \in \mathbb{R}^{6\times 6}$, $P_2 \in \mathbb{R}^{2\times 2}$ and $P_{12} \in \mathbb{R}^{6\times 2}$,

$$\begin{pmatrix} x \\ u \end{pmatrix}^T \begin{pmatrix} P_1 & P_{12} \\ P_{12}^T & P_2 \end{pmatrix} \begin{pmatrix} x \\ u \end{pmatrix} \le 1. \tag{16}$$

This can be further rewritten, for $x = x(k)$, as:

$$u^T P_2 u + 2u^T P_{12}^T x(k) + x(k)^T P_1 x(k) \le 1 \tag{17}$$

Equation (17) represents a scaled and translated ellipsoid $u^T P_2 u \le 1$ which can be written as:

$$(u-a)^T P_2 (u-a) \le \gamma, \tag{18}$$

where

$$a = -P_2^{-1} P_{12} x(k), \tag{19}$$

$$\gamma = 1 + x(k)^T (P_{12}^T (P_2^{-1})^T P_{12} - P_1) x(k). \tag{20}$$

Now the set $\mathcal{U}_k$ defined in (14) can equivalently be written as

$$\mathcal{U}_k(x(k)) = \{u(k) \mid (u(k)-a)^T P_2 (u(k)-a) \le \gamma\}. \tag{21}$$

Problem (13) is now readily solved with modified fast gradient algorithm stated in Algorithm 2. In each iteration of the algorithm, first a candidate solution $v_{i+1}$ is calculated by making one gradient step using (22), in the next step (23) the candidate solution is, if needed, projected to the input constraints set $\mathcal{U}_k$ using the following equation

$$u_{i+1} = \textbf{Proj}_{\mathcal{U}_k(x(k))}(v_{i+1}) = \begin{cases} v_{i+1}, & \|v_{i+1} - a\| \le \gamma, \\ \frac{v_{i+1}-a}{\|v_{i+1}-a\|} \cdot \gamma + a, & \|v_{i+1} - a\| > \gamma. \end{cases} \tag{25}$$

**Algorithm 2:** Set-based fast gradient method for constrained minimization

**Data:** Initial point $u_0 \in \mathcal{U}_{\mathcal{N}}, y_0 = u_0$, number of iterations $i_{\max}$, Lipschitz constant $L$, scaling factors $\beta_0, \ldots, \beta_{i_{\max}-1}$

Calculate set $\mathcal{U}_k(x(k))$ from the set $\bar{\mathcal{E}}_i$ by fixing $x(k)$ and parameters $a$ and $\gamma$;

**for** $i = 0 \to i_{\max} - 1$ **do**

$$v_{i+1} = y_i - \frac{1}{L}\nabla J_N(y_i) = My_i + g; \tag{22}$$

$$u_{i+1} = \textbf{Proj}_{\mathcal{U}_k(x(k))}(v_{i+1}); \tag{23}$$

$$y_{i+1} = u_{i+1} + \beta_i (u_{i+1} - u_i); \tag{24}$$

**end**

## IV. FPGA IMPLEMENTATION OF SET-BASED MPC

In this section we prepare Algorithm 1 for automatic code generation and synthesis on an FPGA by precomputing all the necessary data required by the algorithm, and by exploiting its structure to further ease computational burden. Main steps of the FPGA implemented algorithm are shown in figure 1.

Based on the procedure outlined in Section II we compute the matrices $\mathcal{P}_i$, for $i = 0, \ldots, n_{\max}$ representing the ellipsoidal sets $\mathcal{E}_i$. To check whether the state vector $x$ is contained in an ellipsoid $\mathcal{E}_i$ quadratic form $x^T \mathcal{P}_i x$ needs to be evaluated; this operation, in $n$ dimensional state space consists of $(n+1)n$ multiplications and $(n+1)(n-1)$ adding operations. Because matrices $\mathcal{P}_i$ are positive definite they admit the Cholesky factorization i.e.

$$\mathcal{P}_i = R_i^T R_i \tag{26}$$

where $R_i$ are right triangle matrices. Now the quadratic form

$$x^T \mathcal{P}_i x = x^T R_i^T R_i x = (R_i x)^T (R_i x) \tag{27}$$

can be evaluated with $\frac{1}{2}n(n+3)$ multiplications and $\frac{1}{2}(n+2)(n-1)$ additions.

To find the minimal ellipsoid simple parallel reduction procedure is done for all the indices satisfying

$$x^T \mathcal{P}_i x \le 1 \tag{28}$$

that finds the minimum with $n-1$ comparison operations in $\log_2 n$ "steps".

Based on the found ellipsoid index $i$ fast gradient method is used to solve MPC optimization problem with appropriate cost function

$$J_i(u) = (Ax + Bu)^T \mathcal{P}_{i-1} (Ax + Bu), \tag{29}$$

gradient of which is

$$\nabla J_i(u) = 2B^T \mathcal{P}_{i-1} Bu + 2B^T \mathcal{P}_{i-1} Ax, \tag{30}$$

now the gradient step in Algorithm 2 can be computed as

$$v_{i+1} = My_i + g, \tag{31}$$

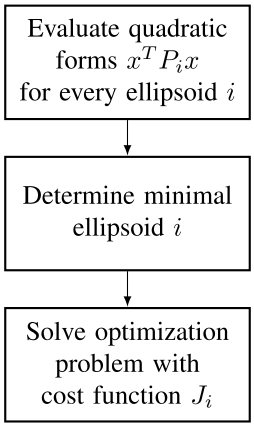


Fig. 1: Block diagram of FPGA computation flow

with

$$M = I - \frac{2}{L} B^T \mathcal{P}_{i-1} B \tag{32}$$

and

$$g = -\frac{2}{L} B^T \mathcal{P}_{i-1} A x =: Hx, \tag{33}$$

where $M$ and $H$ are computed offline. Similarly we precompute the matrices used in projection step of the algorithm using formulas (19) and (20).

## V. Example: controlling the grid-tied inverter

To obtain mathematical model of a grid-tied inverter (Fig. 2) we define three phase quantities

$$\begin{aligned} i_1 = &\ (i_{1a}\ i_{1b}\ i_{1c})^T, \\ v_c = &\ (v_{ca}\ v_{cb}\ v_{cc})^T, \\ i_2 = &\ (i_{2a}\ i_{2b}\ i_{2c})^T \end{aligned}$$

as inverter side current, filter capacitor voltage and grid side current. Three phase quantities are transformed to synchronous reference frame using the Parks transformation matrix

$$P_{dq} = \frac{2}{3} \begin{pmatrix} \sin\theta & \sin\left(\theta - \frac{2}{3}\pi\right) & \sin\left(\theta + \frac{2}{3}\pi\right) \\ \cos\theta & \cos\left(\theta - \frac{2}{3}\pi\right) & \cos\left(\theta + \frac{2}{3}\pi\right) \end{pmatrix}, \tag{34}$$

to obtain

$$\begin{pmatrix} i_{1d} \\ i_{1q} \end{pmatrix} = P_{dq} \begin{pmatrix} i_{1a} \\ i_{1b} \\ i_{1c} \end{pmatrix}$$

$$\begin{pmatrix} v_{cd} \\ v_{cq} \end{pmatrix} = P_{dq} \begin{pmatrix} v_{ca} \\ v_{cb} \\ v_{cc} \end{pmatrix}$$

$$\begin{pmatrix} i_{2d} \\ i_{2q} \end{pmatrix} = P_{dq} \begin{pmatrix} i_{2a} \\ i_{2b} \\ i_{2c} \end{pmatrix}$$

Defining the

$$x(t) = (i_{1d}(t)\ i_{1q}(t)\ v_d(t)\ v_q(t)\ i_{2d}(t)\ i_{2q}(t))^T$$

as the system state with quantities in stationary frame, and vector $u(t) = (u_d(t)\ u_q(t))^T$ as the controlled input that represents the voltages applied by the inverter and vector $v(t) = (v_d\ v_q)^T$ as the grid voltages. The output vector $y(t)$ represents the currents $(i_{2d}(t)\ i_{2q}(t))^T$ injected to the grid. Now the state-space linear model in continuous time is

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + Dv(t) \\ y(t) &= Cx(t) \end{aligned} \tag{35}$$

where

$$A_c = \begin{pmatrix} -\frac{r_1}{L_1} & \omega & -\frac{1}{L_1} & 0 & 0 & 0 \\ -\omega & -\frac{r_1}{L_1} & 0 & -\frac{1}{L_1} & 0 & 0 \\ \frac{1}{C} & 0 & 0 & \omega & -\frac{1}{C} & 0 \\ 0 & \frac{1}{C} & -\omega & 0 & 0 & -\frac{1}{C} \\ 0 & 0 & \frac{1}{L_2} & 0 & -\frac{r_2}{L_2} & \omega \\ 0 & 0 & 0 & \frac{1}{L_2} & -\omega & -\frac{r_2}{L_2} \end{pmatrix},$$

$$B_c = \begin{pmatrix} 1/L_1 & 0 \\ 0 & 1/L_1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}, D_c = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -1/L_2 & 0 \\ 0 & -1/L_2 \end{pmatrix}$$

and

$$C_c = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

where $\omega = 2\pi f$, with grid frequency $f$.

Discrete time model with sampling rate $T$ using the forward Euler discretization is given by

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + Dv(k) \\ y(k) &= Cx(k) \end{aligned} \tag{36}$$

with

$$\begin{aligned} A &= I + TA_c, \\ B &= TB_c, \\ C &= C_c, \\ D &= TD_c. \end{aligned}$$

An equilibrium point $(\bar{x}\ \bar{u})^T$ for system (35), considering a desired reference $r$ of the output $y = (i_{2d}\ i_{2q})^T$ under constant grid voltage $v$, is computed as

$$\begin{pmatrix} \bar{x} \\ \bar{u} \end{pmatrix} = \begin{pmatrix} A_c & B_c \\ C_c & 0 \end{pmatrix}^{-1} \begin{pmatrix} -D_c v \\ r \end{pmatrix}. \tag{37}$$

Defining the $e(k) := x(k) - \bar{x}$ as an error around equilibrium point $(\bar{x}\ \bar{u})^T$ of a system (36) at time $k$, for a given grid current reference $r$ under constant grid voltage $v$, we obtain the error dynamics

$$e(k+1) = Ae(k) + Bu_{\text{err}}(k), \tag{38}$$

where $u_{\text{err}}(k) = u(k) - \bar{u}$.

For the system (38), with parameters given in table I all the data needed for the Set-based MPC algorithm is first computed using the procedures outlined in section

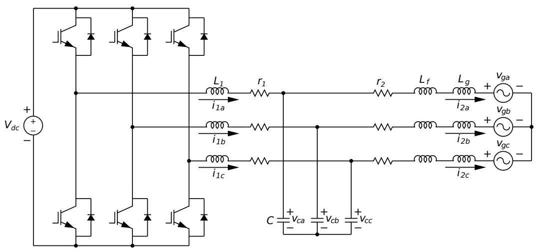

Fig. 2: Two-level grid connected inverter with an LCL filter

TABLE I: Inverter and grid parameters

| Symbol | Description | Value | Unit |
|---|---|---|---|
| $r_1$ | Inverter resistance | 0.5 | $\Omega$ |
| $L_1$ | Filter inductance | 3.4 | mH |
| $C$ | Filter capacitance | 20 | $\mu$F |
| $r_2$ | Grid resistance | 0.5 | $\Omega$ |
| $L_2$ | Filter inductance | 1.8 | mH |
| $V_{dc}$ | DC link voltage | 650 | V |
| $v_g$ | Grid phase voltage(rms) | 230 | V |
| $f$ | Grid frequency | 50 | Hz |
| $f_{PWM}$ | PWM frequency | 10 | kHz |
| $f_s$ | Sampling frequency | 20 | kHz |

IV. Algorithm 1 is then implemented in MATLAB using fixed-point arithmetic, from which the VHDL code is generated using the MATLAB's HDL coder. The generated code is then synthesized for the Digilent Nexys Video board containing the Xilinx Artix-7 xc7a200tsbg484-3 FPGA device using Xilinx Vivado Design Suite. Table II shows modest resource use, while the timing analysis reports a critical path setup time of 332ns, proving this implementation viable for FPGA clock frequencies of up to 3MHz.

The design was tested in a MATLAB/Simulink simulation. The grid-tied inverter was modelled in Simulink and simulation run on a host PC, while the synthesized controller was run on the Xilinx Artix-7 FPGA device. The communication between host PC and FPGA board was realized using a JTAG connection; at every time step of the simulation the measurements (i.e. currents and voltages) were sent to the FPGA board, based on the measurements the control signal is computed by the FPGA device and then sent back to the host PC to be applied to the inverter. The control signal computed by FPGA implementation is then compared to the control signal computed by reference MATLAB implementation of a Set-based MPC algorithm. As can be seen in Figure 3 control actions are bit for bit equal at every time instance.

TABLE II: FPGA resource utilization

| Resource | Utilization | Available | Utilization(%) |
|---|---|---|---|
| LUT | 20191 | 133800 | 15.1 |
| LUTRAM | 135 | 46200 | 0.3 |
| FF | 1411 | 267600 | 0.5 |
| BRAM | 2.5 | 365 | 0.7 |
| DSP | 502 | 740 | 67.8 |
| IO | 2 | 285 | 0.7 |
| BUFG | 3 | 32 | 9.4 |
| MMCM | 1 | 10 | 10.0 |

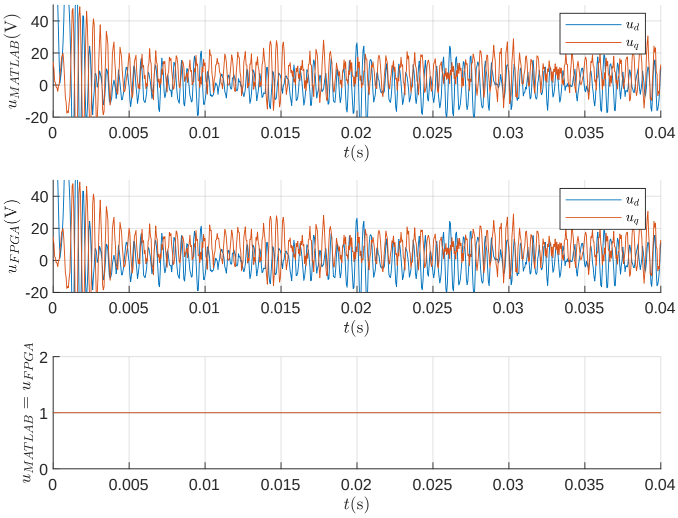


Fig. 3: Control actions computed with MATLAB (top) and FPGA (mid.) implementations of Set-based MPC. (bot.) Bit equality of two signals (1 = true,0 = false).

## VI. Conclusion

This paper presented an FPGA implementation procedure of a model predictive controller based on ellipsoidal sets. To speed up execution time of algorithm ellipsoidal set computation is done offline, online an optimization problem is solved using fast gradient method. Proposed MPC algorithm is shown to be synthesizable on an Xilinx Artix7 device, with computation times in microsecond range, making it well suited for controlling the systems with very fast dynamics, such as power converters. Programmed FPGA device was tested in FIL simulation and shown to produce bit for bit equal control signals as the reference MATLAB implementation.

## References


[1] S. Kouro, M. A. Perez, J. Rodriguez, A. M. Llor, and H. A. Young, "Model predictive control: Mpc's role in the evolution of power electronics," *IEEE Industrial Electronics Magazine*, vol. 9, no. 4, pp. 8–21, 2015.

[2] S. Vazquez, J. Rodriguez, M. Rivera, L. G. Franquelo, and M. Norambuena, "Model predictive control for power converters and drives: Advances and trends," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 2, pp. 935–947, 2016.

[3] P. Karamanakos, E. Liegmann, T. Geyer, and R. Kennel, "Model predictive control of power electronic systems: Methods, results, and challenges," *IEEE Open Journal of Industry Applications*, vol. 1, pp. 95–114, 2020.

[4] S. Lucia, D. Navarro, O. Lucia, P. Zometa, and R. Findeisen, "Optimized fpga implementation of model predictive control for embedded systems using high-level synthesis tool," *IEEE transactions on industrial informatics*, vol. 14, no. 1, pp. 137–145, 2017.

[5] O. Gulbudak and E. Santi, "Fpga-based model predictive controller for direct matrix converter," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 7, pp. 4560–4570, 2016.

[6] S. Richter, T. Geyer, and M. Morari, "Resource-efficient gradient methods for model predictive pulse pattern control on an fpga," *IEEE Transactions on control systems technology*, vol. 25, no. 3, pp. 828–841, 2016.

[7] Z. Ma, S. Saeidi, and R. Kennel, “Fpga implementation of model predictive control with constant switching frequency for pmsm drives,” *IEEE transactions on industrial informatics*, vol. 10, no. 4, pp. 2055–2063, 2014.
[8] F. Blanchini, “Set invariance in control,” *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999.
[9] D. Angeli, A. Casavola, G. Franzè, and E. Mosca, “An ellipsoidal off-line mpc scheme for uncertain polytopic discrete-time systems,” *Automatica*, vol. 44, no. 12, pp. 3113–3119, 2008.
[10] D. Angeli, A. Casavola, and E. Mosca, “Predictive control with partial state information,” in *2001 European Control Conference (ECC)*. IEEE, 2001, pp. 962–967.