Hardware Realization of Robust Controller Designed Using Reflection Vectors

Ján Cigánek, Michal Kocúr, Štefan Kozák
Faculty of Electrical Engineering and Information Technology
Slovak University of Technology in Bratislava
Bratislava, Slovakia
E-mail: jan.ciganek@stuba.sk, michal.kocur@stuba.sk, stefan.kozak@stuba.sk

***Abstract*— The presented paper deals with the new approach of robust controller design using the reflection vectors techniques. The control structure consists of feed-forward and feedback part. Proposed algorithms were tested using Field-Programmable Gate Array (FPGA) technology for DC motor. Simulations and co-summations were realized in MATLAB-Simulink. The co-simulation allows as to validate working hardware and to accelerate simulations in Simulink and MATLAB. The obtained results demonstrate very effective applicability of the theoretical principles for control of processes subject to parametrical model uncertainty.**

Keywords - robust control; robust stability; parametrical uncertainty; quadratic programming; reflection vectors; FPGA; co-simulation.

I. INTRODUCTION

During the last ten years, development of robust control elementary principles and evolution of new robust control methods for different model uncertainty types are visible. Based on theoretical assumptions, modeling and simulation methods, an effective approach to the control of processes with strong and undefined uncertainties is designed. Such uncertainties are typical for biotechnology processes, chemical plants, automobile industry, aviation, etc. For such processes, it is necessary to design robust and practical algorithms which ensure the high performance and robust stability using proposed mathematical techniques with respect the parametric and unmodelled uncertainties [1][2]. Solution to such problems is possible using robust predictive methods and „soft-techniques“ which include fuzzy sets [3], neuron networks and genetic algorithms.

Robust control is used to guarantee stability of plants with parameter changes. The robust controller design consists of two steps:

- analysis of parameter changes and their influence for closed-loop stability,
- robust control synthesis.

There are two approaches for implementing control systems using digital technology. The first approach is based on software which implies a memory-processor interaction. The memory holds the application program while the processor fetches, decodes, and executes the program instructions. Programmable Logic Controllers (PLCs), microcontrollers, microprocessors, Digital Signal Processors (DSPs) and general purpose computers are tools for software implementation. On the other hand, the second approach is based on hardware. Early hardware implementation is achieved by magnetic relays extensively used in old industry automation systems. Then, it became achievable by means of digital logic gates and Medium Scale Integration (MSI) components. When the system size and complexity increases, Application Specific Integrated Circuits (ASICs) are utilized. The ASIC must be fabricated on a manufacturing line, a process that takes several months, before it can be used or even tested [4][5]. FPGAs are configurable ICs and used to implement logic functions.

Today's high-end FPGAs can hold several millions gates and have some significant advantages over ASICs. They ensure ease of design, lower development costs, more product revenue and the opportunity to speed products to market [6]. At the same time, they are superior to software-based controllers as they are more compact, power-efficient, while adding high speed capabilities.

The presented paper is organized as follows. In Section II, the complete procedure of robust controller design using reflection vectors is presented. Section III offers a short overview of hardware implementation of the proposed control algorithm using FPGA. The applicability of the control algorithm is shown on the case study for a real DC system with parametrical model uncertainty in Section IV. In Section V, the summary of the paper is discussed.

II. PROBLEM STATEMENT

Let us consider the robust control synthesis of a scalar discrete-time control loop. The transfer function of the original continuous-time system is described by the transfer function

$$G_P(s) = \frac{\bar{B}(s)}{\bar{A}(s)} e^{-Ds} = \frac{\bar{b}_m s^m + \bar{b}_{m-1} s^{m-1} + \ldots + \bar{b}_0}{\bar{a}_n s^n + \bar{a}_{n-1} s^{n-1} + \ldots + \bar{a}_0} e^{-Ds} \quad (1)$$

The transfer function of (1) can be converted to its discrete-time counterpart

$$G_P(z^{-1}) = \frac{b_0 + b_1 z^{-1} + \ldots + b_n z^{-n}}{1 + a_1 z^{-1} + \ldots + a_n z^{-n}} z^{-d} \quad (2)$$

 ISBN: 978-1-61208-394-0

For (2), a discrete-time controller is to be designed in form

$$G_R(z)=\frac{q_0+q_1z^{-1}+\ldots+q_\upsilon z^{-\upsilon}}{1+p_1z^{-1}+\ldots+p_\mu z^{-\mu}}=\frac{Q(z)}{P(z)}=\frac{U(z)}{E(z)} \quad (3)$$

The corresponding closed-loop characteristic equation is

$$1+G_P(z^{-1})G_R(z^{-1})=0 \quad (4)$$

Substituting (3) and (2) in (4), after a simple manipulation yield the characteristic equation

$$1+G_PG_R=(1+p_1z^{-1}+\ldots+p_\mu z^{-\mu})(1+a_1z^{-1}+\ldots+a_nz^{-n})+ \quad (5)$$
$$+(q_0+q_1z^{-1}+\ldots+q_\upsilon z^{-\upsilon})(b_1z^{-1}+\ldots+b_nz^{-n})z^{-d}=0$$

Unknown coefficients of the discrete controller can be designed using various methods. In this paper, a robust controller design method based on reflection vectors is used.

The pole assignment problem is as follows: find a controller $G_R(z)$ such that $C(z)=e(z)$ where $e(z)$ is a given (target) polynomial of degree k. It is known [7] that, when $\mu=n-1$, the above problem has a solution for arbitrary $e(z)$ whenever the plant has no common pole-zero pairs. In general, for $\mu < n-1$ exact attainment of a desired target polynomial $e(z)$ is impossible.

Let us relax the requirement of attaining the target polynomial $e(z)$ exactly and enlarge the target region to a polytope V in the polynomial space containing the point e representing the desired closed-loop characteristic polynomial. Without any restriction, we can assume that $a_n=p_0=1$ and deal with monic polynomials $C(z)$, i.e., $\alpha_0=1$.

Let us introduce the stability measure as $\rho = c^T c$, where

$$c=S^{-1}C \quad (6)$$

and S is a matrix of dimensions $(n+\mu+1)\times(n+\mu+1)$ representing vertices of the target polytope V. For monic polynomials holds

$$\sum_{i=1}^{k+1}c_i=1 \quad (7)$$

where $k=n+\mu$. If all coefficients are positive, i.e., $c_i>0$, $i=1,\ldots,k+1$, then the point C is placed inside the polytope V.

The minimum ρ is attained if

$$c_1=c_2=\ldots=c_{k+1}=\frac{1}{k+1} \quad (8)$$

then the point C is placed in centre of the polytope V.

In the matrix form, we have

$$C=Gx \quad (9)$$

where G is the Sylvester matrix of the plant with dimensions $(n+\mu+d+1)\times(\mu+\upsilon+2)$ and x is the $(\mu+\upsilon+2)$-vector of controller parameters: $x=[p_\mu,\ldots,p_1,1,q_\upsilon,\ldots,q_0]^T$.

Now, we can formulate the following control design problem: find a discrete controller, where the closed-loop characteristic polynomial $C(z)$ is placed:

a) In a stable target polytope V, $C(z)\in V$ (to guarantee stability),
b) As close as possible to a target polynomial $e(z)$, $e(z)\in V$ (to guarantee performance).

Let the polytope V denote the $(k+1)\times N$ matrix composed of coefficient vectors v_j, $j=1,\ldots,N$ corresponding to vertices of the polytope V.

Then, we can formulate the above controller design problem as an optimization task: Find x that minimizes the cost function

$$J_1=\min_x x^TG^TGx-2e^TGx=\min_x\|Gx-e\|^2 \quad (10)$$

subject to the linear constraints

$$Gx=Vw(x), \quad (11)$$
$$w_j(x)>0,\ j=1,\ldots,N, \quad (12)$$
$$\sum_j w_j(x)=1. \quad (13)$$

Here, $w(x)$ is the vector of weights of the polytope V vertices to obtain the point $C = G x$. Fulfillment of the latter two constraints (12), (13) guarantees that the point C is indeed located inside the polytope V. Then, finding the robust pole-placement controller coefficients represents an optimization problem that can be solved using the Matlab Toolbox OPTIM (quadprog) with constraints.

Generally J_1 is a kind of distance to the centre of the target polytope V. It is better to use another criterion J_2, which measures the distance to the Schur polynomial $E(z)$

$$J_2=(C-E)^T(C-E)=(Gx-E)^T(Gx-E). \quad (14)$$

It is possible to use the weighted combination of J_1 and J_2

$$J=(1-\alpha)J_1+\alpha J_2,\quad 0\le\alpha\le1 \quad (15)$$

and to solve the following quadratic programming task

$$J=\min_x\{x^TG^T[(1-\alpha)(SS^T)^{-1}+\alpha I_{k+1}]Gx-2\alpha E^TGx\}, \quad (16)$$
$$S^{-1}Gx<0.$$

Assume the discrete robust controller design task with parametrical uncertainties in system description. Let us also

assume that coefficients of the discrete-time system transfer functions a_n, ..., a_1 and b_n, ..., b_1 are placed in polytope W with the vertices $d^j = [a_n^j, \ldots a_1^j, b_n^j, \ldots, b_1^j]$:

$$W = conv\{d_j, j = 1,..., M\} \quad (17)$$

As (9) is linear in system parameters, it is possible to claim that for arbitrary vector of the controller coefficients x is the vector of the characteristic polynomial coefficients $C(z)$ placed in the polytope A with vertices $a^1, \ldots, a^M$:

$$A = conv\{a^j, j = 1,..., M\} \quad (18)$$

where $a^j = D^j x$ and D^j is the Sylvester matrix of dimensions $(n + \mu + d + 1) \times (\mu + \upsilon + 2)$, composed of vertices set d^j, as in case of the exact model (9).

A. Stable Region Computation via Reflection Coefficients

Polynomials are usually specified by their coefficients or roots. They can be characterized also by their reflection coefficients using Schur-Cohn recursion.

Let $C_k(z^{-1})$ be a monic polynomial of degree k with real coefficients $c_i \in R$, $i = 0, ..., k$

$$C(z^{-1}) = 1 + c_1 z^{-1} + ... + c_k z^{-k} \quad (19)$$

Reciprocal polynomial $C_k^*(z^{-1})$ of the polynomial $C_k(z^{-1})$ is defined in [8] as follows

$$C_k^*(z^{-1}) = c_k + c_{k-1} z^{-1} + \ldots + c_1 z^{-k+1} + z^{-k} \quad (20)$$

Reflection coefficients r_i, $i = 1, ..., k$, can be obtained from the polynomial $C_k(z^{-1})$ using backward Levinson recursion [9]

$$z^{-1} C_{i-1}(z^{-1}) = \frac{1}{1 - |r_i^2|} \left[C_i(z^{-1}) - r_i C_i^*(z^{-1}) \right] \quad (21)$$

where $r_i = -c_i$ and c_i is the last coefficient of $C_i(z^{-1})$ of degree i. From (21) we obtain in a straightforward way:

$$C_i(z^{-1}) = z^{-1} C_{i-1}(z^{-1}) + r_i C_{i-1}^*(z^{-1}). \quad (22)$$

Expressions for polynomial coefficients $C_{i-1}(z^{-1})$ and $C_i(z^{-1})$ result from equations (22,23):

$$C_{i-1}(z^{-1}) = \frac{1}{1 - |r_i^2|} \left[\sum_{j=0}^{i-1} \left(c_{i,j+1} - r_i c_{i,i-j-1} \right) z^{-j} \right] \quad (23)$$

$$C_i(z^{-1}) = \sum_{j=0}^{i} \left(c_{i-1,j-1} + r_i c_{i-1,i-j-1} \right) z^{-j}. \quad (24)$$

The reflection coefficients r_i are also known as Schur-Szegö parameters [8], partial correlation coefficients [11] or k-parameters [10]. Presented forms and structures were effectively used in many applications of signal processing [10] and system identification [11]. A complete characterization and classification of polynomials using their reflection coefficients instead of roots (zeros) of polynomials is given in [8].

The main advantage of using reflection coefficients is that the transformation from reflection to polynomial coefficients is very simple. Indeed, according to (22) and (24), polynomial coefficients c_i depend multilinearly on the reflection coefficients r_i. If the coefficients $c_i \in R$ are real, then also the reflection coefficients $r_i \in R$ are real.

Transformation from reflection coefficients $r_i, i=1,...,k$, to polynomial coefficients $c_i, i=1,...,k$, is as follows

$$\begin{aligned} c_i &= c_i^{(k)}, \\ c_i^{(i)} &= -r_i, \\ c_j^{(i)} &= c_j^{(i-1)} - r_i c_{i-j}^{(i-1)} \\ i &= 1,\ldots,k;\ j = 1,\ldots,i-1 \end{aligned} \quad (25)$$

Lemma 1. A linear discrete-time dynamic system is stable if its characteristic polynomial is Schur stable, i.e., if all its poles lie inside the unit circle.

The stability criterion in terms of reflection coefficients is as follows [8].

Lemma 2. A polynomial $C(z^{-1})$ has all its roots inside the unit disk if and only if $|r_i| < 1$, $i = 1, ..., k$.

A polynomial $C(z^{-1})$ lies on the stability boundary if some $r_i = \pm 1$, $i = 1, ..., k$. For monic Schur polynomials, there is a one-to-one correspondence between $C = [c_k, \ldots, c_1]^T$ and $r = [r_1, \ldots, r_k]^T$.

Stability region in the reflection coefficient space is simply the k-dimensional unit hypercube $R = \{r_i \in (-1,1), i = 1,\ldots,k\}$. The stability region in the polynomial coefficient space can be found starting from the hypercube R.

B. Stable Polytope of Reflection Vectors

It will be shown that, for a family of polynomials, the linear cover of the so-called reflection vectors is Schur stable.

Definition 1. The reflection vectors of a Schur stable monic polynomial $C(z^{-1})$ are defined as the points on stability boundary in polynomial coefficient space generated by changing a single reflection coefficient r_i of the polynomial $C(z^{-1})$.

Let us denote the positive reflection vectors of $C(z^{-1})$ as $v_i^+(C)=(C|r_i=1), i=1,\ldots,k,$ and the negative reflection vectors of $C(z^{-1})$ as $v_i^-(C)=(C|r_i=-1), i=1,\ldots,k.$

The following assertions hold:

1) every Schur polynomial has *2k* reflection vectors $v_i^+(C)$ and $v_i^-(C), i=1,\ldots,k$;
2) all reflection vectors lie on the stability boundary $(r_i^v=\pm 1)$;
3) the line segments between reflection vectors $v_i^+(C)$ and $v_i^-(C)$ are Schur stable.

In the following theorem a family of stable polynomials is defined such that the polytope generated by reflection vectors of these polynomials is stable.

Theorem 1. Consider $r_1^C\in(-1,1)$, $r_k^C\in(-1,1)$ and $r_2^C=\ldots=r_{k-1}^C=0$. Then, the inner points of the polytope *V(C)* generated by the reflection vectors of the point *C*

$$V(C)=conv\{v_i^{\pm}(C), i=1,\ldots,k\} \qquad (26)$$

are Schur stable.

C. Robust Controller Design

A robust controller is to be designed such that the closed-loop characteristic polynomial is placed in the stable polytope (linear cover) of reflection vectors. It means that the following problems have to be solved:

1. choice of initial polynomial $C(z^{-1})$ for generating the polytope *V(C)*,
2. choice of *k + 1* most suitable vertices of *V(C)* to build a target simplex *S*,
3. choice of a target polynomial $E(z^{-1})$.

In the following section some "thumb rules" are given for choosing a stable target simplex *S*.

To choose *k + 1* vertices of the target simplex *S* we use the well known fact that poles with positive real parts are preferred to those with negative ones [1]. The positive reflection vectors $v_i^{+(C)}$ with *i* odd and negative reflection vectors $v_i^{-(C)}$ with *i* even are chosen yielding *k* vertices. The *(k+1)*th vertex of the target simplex *S* is chosen as the mean of the remaining reflection vectors.

The target polynomial $E(z^{-1})$ of order k is reasonable to be chosen inside the stable polytope of reflection vectors *V(C)*. A common choice is $E(z^{-1})=C(z^{-1})$.

For higher-order polynomials, the size of the target simplex *S* is considerably less than the volume of the polytope of reflection vectors *V*. That is why the above quadratic programming method with a preselected target simplex *S* works only if uncertainties are sufficiently small. Otherwise, it is reasonable to use some search procedure to find a robust controller such that the polytope of closed-loop characteristic polynomial is placed inside the stable polytope of reflection vectors *V(C)*.

In terms of the performance, the comparison of the proposed solution with other solutions would not be quite transparent due to different structure of the control loops or due to the different polynomial degrees of the controller.

III. IMPLEMENTATION OF CONTROL ALGORITHM

The digital form of the controller can be obtained from (3). Recursive form of control algorithm is expressed by the following equation:

$$u(k)=q_1e(k-1)+q_2e(k-2)-p_1u(k-1)-p_2u(k-2) \qquad (27)$$

For implementation of control algorithm (27) for FPGA it is necessary to decompose equation into simple arithmetic operations:

$$\begin{aligned} e(k)&=w(k)-y(k)\\ q_1e_{-1}&=q_1*e(k-1)\\ q_2e_{-2}&=q_2*e(k-2)\\ p_1u_{-1}&=p_1*u(k-1)\\ p_2u_{-2}&=p_2*u(k-2)\\ s_1&=q_1e_{-1}+q_2e_{-2}\\ s_2&=p_1u_{-1}+p_2u_{-2}\\ s_3&=s_1-s_2 \end{aligned} \qquad (28)$$

Control output *u* must be bounded in the range from u_{min} to u_{max}.

$$\begin{aligned} u_{\max}&\rightarrow if(s3>u_{\max})\\ u(k)=s3&\rightarrow if(u_{\min}\le s3\le u_{\max})\\ u_{\min}&\rightarrow if(s3<u_{\min}) \end{aligned} \qquad (29)$$

In this case, the parallel design of control algorithm is used, which means that each of the operation has its own arithmetic unit, either accumulator or multiplier. Parallel design is shown in Figure 1.

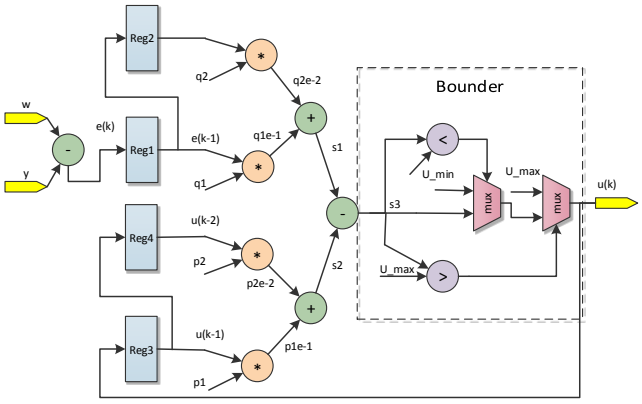

Figure 1. Parallel design of control algorithm

Each sampling period is loaded the motor system output *y(k)* from the input in. Control error *e(k)* is computed in sub block where the signal *y(k)* is subtracted from *w(k)*. Signal *e(k)* is held in the registry *REG1* for one sampling period. Register *REG1* output signal is thus *e(k-1)*. In the same

manner, *e(k−2), u(k−1)* and *u(k−2)* are recorded at *REG2*, *REG3* and *REG4* by latching *e(k−1), u(k)* and *u(k-1)* respectively. For multiplication, they are using digital signal processor (DSP) cores in FPGA chip. Results of multiplications are counted in to control output. This control output is then bounded in the range from −12V to 12V.

Inputs *w* and *y* are represented with rpm (Revolutions per minute). Input range is -2048 to 2047 rpm, because of the 12bit signed data type. Output of the controller is represented with volts. In signed binary representations the maximum control output is $12_{(10)}$V = $01100_{(2)}$ and the minimum is $-12_{(10)}$ V = $10100_{(2)}$. We can write this range into 5 bits. Real numbers are useful for better quantization of the control output. For implementation of the real numbers, it has been used fixed point arithmetic [12]. As we can see in the Figure 2, the first (MSB) bit of output vectors is reserved for a sign. Next four bits are reserved for the integer part and last seven bits are used for the fractional part.

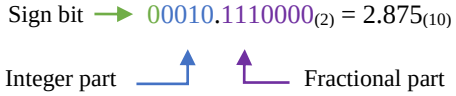

Figure 2. Fixed-point control output

Fixed point arithmetic is applied throughout the control algorithm. In designing this algorithm, the fixed-point arithmetic range rules must be respected. The data widths in the fixed-point arithmetic were designed that there is no possibility of an overflow. For example, the result of summation or subtraction of two 12-bit vectors has range 13-bit. Table 1 represents used range rules for fixed point arithmetic.

TABLE I. FIXED-POINT RANGE RULES

Operation	Result Range
A + B	Max(A'left, B'left)+1 Min(A'right, B'right)
A – B	Max(A'left, B'left)+1 Min(A'right, B'right)
A * B	A'left + B'left+1 A'right + B'right

In the case of parallel the design of control algorithm, the control output after last summation (resp. subtraction) has range 40-bit (16-bit for fractional part). It must be used a bounder block to ensure of range (-12 V to 12 V) for 12bit. Bounder is the value limitation logic that keeps the output in the defined range. Bounded signal is latched at register *REG3*, thus becomes *u(k−1)* of next cycle. In this way, the anti-windup protection is also ensured.

System Generator toolbox for Simulink ensures that between the blocks gateway in and gateway out algorithm performs as it was implemented on FPGA (Figure 3). The parallel design of control algorithm designed in VHDL is contained in the control algorithm block.

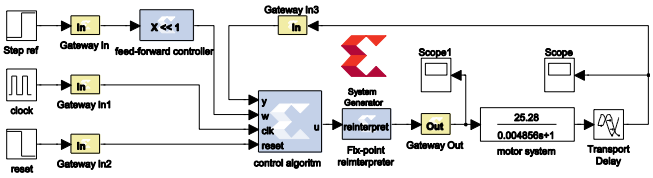

Figure 3. Schematic of control circuit using Xilinx blocks

In the co-simulation process, we used Xilinx Spartan-6 FPGA which is included in SP-601 demoboard. Spartan-6 FPGAs offer advanced power management technology, up to 150K logic cells, advanced memory support, 250MHz DSP slices, and 3.2Gbps low-power transceivers [13].

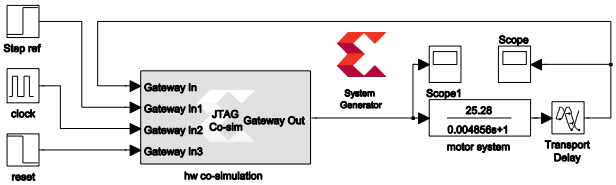

Figure 4. Schematic of control circuit using Co-sim block

Based on the successful verification of the Xilinx blocks algorithm, we generated the co-simulation block by Xilinx System Generator (Figure 4). If is co-sim block included in simulation scheme it must be FPGA connected during the simulation process into the computer. At the start of simulation System Generator records functionality of co-sim block into FPGA. The co-sim block behaves as an in-out black box. Control output is computed in FPGA. Other blocks like the transfer function or step generator are still simulated in Simulink. Communication between FPGA board and the computer can be provided by Point to Point Ethernet or USB JTAG.

IV. CASE STUDY

Let us consider a DC system described by the first order nominal transfer function

$$G_P(s) = \frac{B(s)}{A(s)} e^{-Ds} = \frac{K}{T_1 s + 1} e^{-Ds} = \frac{25.28}{0.004856 s + 1} e^{-0.001 s} \quad (30)$$

where the coefficients K, T_1 are varying in uncertainty intervals $K \in \langle 25; 25.5 \rangle$, $T_1 \in \langle 0.0045; 0.0052 \rangle$

Let us consider the nominal continuous-time transfer function which is converted to the discrete-time form with the sampling period *T=0.001s:*

$$G_p(z^{-1}) = \frac{0.4228 z^{-1} + 4.282 z^{-2}}{1 - 0.8139 z^{-1}} \quad (31)$$

The main task is to design a robust iscrete-time controller (3), with polynomial degrees $\upsilon=1$, $\mu=2$.

From the transfer function (31) and matrix form of (9), we can obtain:

 ISBN: 978-1-61208-394-0

$$C=\begin{bmatrix} 0 & 0 & 0 & 0.4228 & 0 \\ -0.8139 & 0 & 0 & 0 & 0.4228 \\ 1 & -0.8139 & 0 & 0 & 0 \\ 0 & 1 & -0.8139 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} p_2 \\ p_1 \\ 1 \\ q_1 \\ q_0 \end{bmatrix}$$

Let us choose the initial polynomial $C(z^{-1})$ for generating the polytope $V(C)$ as follows

$$C(z)=[1-0.2z^{-1}][1-0.1z^{-1}][1+0.2z^{-1}][1+0.3z^{-1}] \qquad (32)$$

with reflection coefficients $r_1=0.2,\ r_2=-0.07,\ r_3=-0.008,$ $r_4=0.0012.$

Now, we can find the reflection vectors $v_i(C)$ of the initial polynomial $C(z^{-1})$ leading to the matrix form of the target simplex S (vertex polynomial coefficients)

$$S=\begin{bmatrix} 0 & 0 & 0 & -0.3 & 0.5 \\ 0 & 0 & -0.3 & 0.5 & 0 \\ 0 & -0.3 & 0.5 & 0 & 0 \\ -0.3 & 0.5 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \qquad (33)$$

The discrete-time controller design task for the nominal transfer function (30) has been solved via quadratic programming taking $\alpha=0.3$ in the cost function J (16).

For the selected target simplex S, we have obtained the following discrete-time feedback controller

$$G_{FB}(z^{-1})=\frac{Q(z^{-1})}{P(z^{-1})}=\frac{0.025+0.0231z^{-1}}{1+0.0159z^{-1}+0.013z^{-2}} \qquad (34)$$

and the control law is expressed in recursive form

$$u_2(k)=-0.016u_2(k-1)-0.013u_2(k-2)+ \\ +0.025y(k)+0.0231y(k-1) \qquad (35)$$

For verification of the FGPA hardware co-simulation of the digital controller, we realized a practical experiment. In co-simulation, we made step of reference signal at 0.01s.

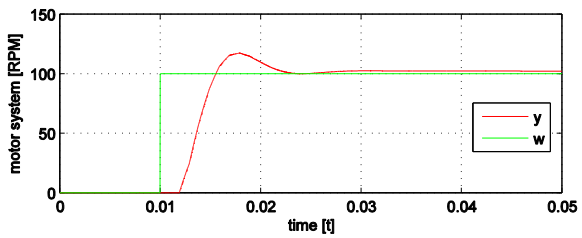

Figure 5. Time response of output and reference variable under robust controller

The corresponding closed-loop step response under the feedback controller (34) and feed-forward controller $G_{FF}(z^{-1})=S(z^{-1})/P(z^{-1})=2$ is in Figure 5.

V. CONCLUSION

The paper deals with the development of robust control algorithm based on reflection vectors methodology. The proposed algorithm can be effective realized using FPGA structure end guaranteeing stability, robustness and high performance. Theoretical results were verified on the example for feedback and feedforward control structures. The methods were also successfully tested for stable and unstable processes.

The illustrative example was solved using quadratic programming for suitably defined performance function. Simulation results prove applicability of the proposed robust controller design theory for systems with parametric uncertainty.

Digital controller was successfully implemented and hardware co-simulated on Spartan-6 FPGA board. From the obtained results, it is evident that the application of FPGA structure is very suitable for high speed processes. In this paper, we presented the basic necessary principles how to realize and modified existing digital robust control algorithms. The co-simulation option can be useful to accelerate simulation of advanced control algorithms.

ACKNOWLEDGMENT

This paper is supported by APVV project No. APVV-0772-12.

REFERENCES

[1] J. Ackermann, “Robust Control” Systems with Uncertain Physical Parameters. Springer-Verlag. London, UK, 1993.

[2] J. Ciganek and S. Kozak, „Robust Controller Design Based on Reflection Vectors”. Control Applications, (CCA) & Intelligent Control, (ISIC), 2009, pp. 938–943.

[3] J. Ciganek and F. Noge,” Fuzzy Logic Control of Mechatronic Systems”, IEEE Int. conf. on Process Control, Strbske Pleso, Slovak Republic, 2013, pp. 309–313.

[4] Š. Kozak “Development of control engineering methods and their applications in industry” In 5th Int. Scientific-Technical Conference Process Control 2002. Kouty nad Desnou, Czech Rep., 2002, pp. 218-222.

[5] M. Kocur, “HW Realization of PID algorithm based on FPGA,” Slovak University of Technology in Bratislava, Bratislava.

[6] V. Viswanathan “Embedded Control Using FPGA”. Indian Institute of Technology, Bombay, 2005

[7] L. H. Keel and S. P. Bhattachayya, “A linear programming approach to controller design.” Automatica, vol. 35, 1999, pp. 1717–1724.

[8] J. L. Diaz-Barrero and J. J. Egozcue, “Characterization of polynomials using reflection coefficients.” Appl. Math. E-Notes, vol. 4, 2004, pp. 114–121.

[9] B. Picinbono and M. Bendir, “Some properties of lattice autoregressive filters”, IEEE Trans. Acoust. Speech Signal Process, 34, 1986, pp. 342–349.

[10] A. M. Oppenheim and R. W. Schaffer, “Discrete-Time Signal Processing.” Prentice-Hall, Englewood Cliffs,1989.

[11] S. M. Kay, “Modern Spectral Estimation,” Prentice Hall, New Jersey, 1988.

[12] D. Bishop. “Fixed point package user’s guide”. http://www.vhdl.org, [retrieved: 11, 2014]

[13] Xilinx, “SP601 Hardware User Guide”

 ISBN: 978-1-61208-394-0