MIPRO 2016, May 30 - June 3, 2016, Opatija, Croatia

# Buck converter controlled by Arduino Uno

H. Kovačević, Ž. Stojanović
Department of Electrical Engineering, Zagreb, Croatia
Polytechnic of Zagreb, Zagreb, Croatia
hkovacevi@tvz.hr, zeljko.stojanovic@tvz.hr

**Abstract – The paper presents a buck converter controlled with microcontroller integrated on Arduino Uno board. The control is implemented by use of PI controller embedded on Arduino Uno board. Open loop control-to-output transfer function is obtained from measured step response. The PI controller and feedback divider transfer functions are synthesized to get desired loop gain. As a verification of analysis, input voltage step responses of regulated and unregulated buck converter are compared and improvements are identified.**

## I. INTRODUCTION

Buck converters are widely used in various types of electrical equipment and described at large in literature [1, 2]. The control of a buck converter is essentially important. There are many power converter control methods like linear control, fuzzy logic, predictive control, etc. [1, 3, 4]. Each of those has its advantages and disadvantages [3-5]. One of the most commonly used techniques of linear control is PI control because of its simplicity in design, implementation and understanding of operation [5, 6].

This article deals with a buck converter controlled by microcontroller ATmega 328 based on Arduino Uno board. It has been chosen because of its popularity and simplicity of use [7-10]. In this article, PI controller is realized by programming of Arduino Uno board. The converter is analyzed as a feedback system. Elements of feedback system are determined in order to achieve satisfying stability and time response. Simulation of input voltage step response for regulated and unregulated converter is performed. Programming code is shown in the appendix. Conclusions and proposals for next researches are done.

## II. BUCK CONVERTER SCHEMATIC

Physical realization of the buck converter is shown in Figure 1. The basic circuit of the converter consists of a DC power supply $E$, switches $V1$ and $V2$, output filter $L_d$ - $C_d$ and load $R_d$. In the drive unit transistor $V3$ drives controlled switch $V1$. Resistors $R_4$ and $R_5$ are used for current adjustment. Capacitors $C_1$ and $C_2$ and resistor $R_3$ are used to improve switching characteristics. As a control device a microcontroller is used. The output voltage is sensed by microcontroller analog input pin A1. The driving transistor $V3$ is PWM controlled by digital pin D9.

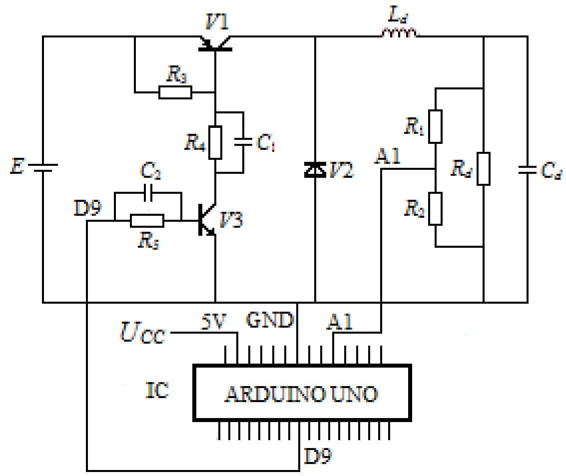


Figure 1 Buck converter schematic

A list of components and basic parameters of buck converter is shown in Table 1. The physical realization of buck converter is presented in Figure 2.

TABLE I. LIST OF COMPONENTS AND BASIC PARAMETERS

| Component | Value | Component | Value |
|---|---|---|---|
| $E$ | 20 V | $C_1$ | 5,6 nF |
| $U_d(0)$ | 13,2 V | $C_2$ | 2,2 nF |
| $I_d(0)$ | 0,34 A | $C_d$ | 100 µF |
| $d = U_d(0)/E$ | 0,66 | $L_d$ | 2,17 mH |
| $R_1$ | 470 kΩ | $V1$ | MJE1501G |
| $R_2$ | 100 kΩ | $V2$ | BC107B |
| $R_3$ | 27 Ω | $V3$ | 1N5408 |
| $R_4$ | 220 Ω | Microcont. | ATmega 328 |
| $R_5$ | 1 kΩ | Board | Arduino Uno |
| $R_d$ | 39 Ω | $U_{CC}$ | 5 V |
| Switching frequency | 8 kHz | Sampling period | 2 µs |

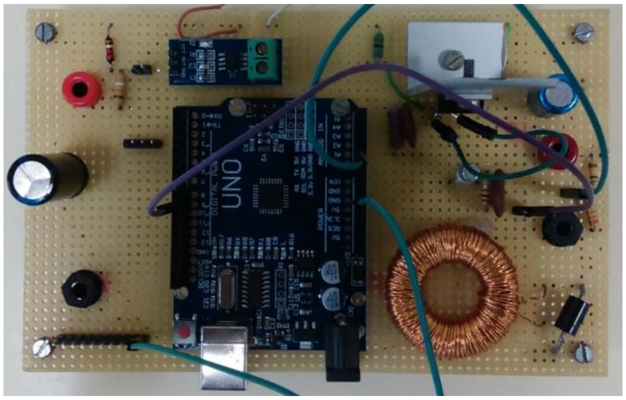

Figure 2 Physical realization of buck converter

## III. BUCK CONVERTER CONTROL

Buck converter control can be realized with a digital or analog controller. A digital controller was chosen because of its parameter changes flexibility such as desired output voltage, switching frequency and PI controller proportional and integral parameters.

### *A. Block diagram of regulator system small-signal model*

The buck converter is operating in voltage mode control. Feedback loop regulation of buck converter is shown in Figure 3. The output voltage $v(t)$ is measured by the microcontroller using a voltage divider with gain $H(s)$. The error signal

$$v_e(s) = H(s)v(s) - v_{ref}(s)$$

sets the duty cycle of PWM signal. Control algorithm was accomplished by using PI controller implemented in Arduino Uno board. For the analysis, whole schematic is presented by small-signal block diagram of the feedback system in Figure 4. List of all variables and functional block parameters of the feedback system and its descriptions is shown in Table 2.

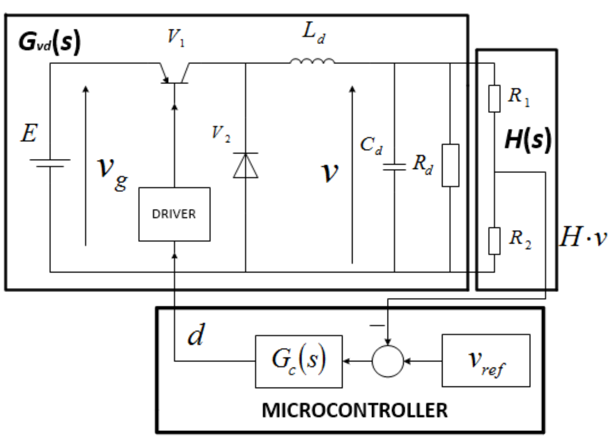


Figure 3 Feedback loop regulation of buck converter

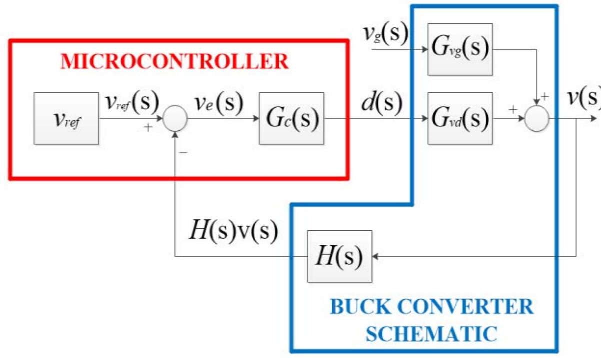


Figure 4 Small-signal block diagram of the feedback system

### *B. Obtained transfer functions of the buck converter*

In order to control the converter successfully, transfer functions of all blocks have to be determined.

Buck converter open loop control-to-output transfer function

$$G_{vd}(s) = \frac{v(s)}{d(s)}$$

is determined experimentally by step change of the duty cycle $d(t)$ and measuring the transient of output voltage $v(t)$, Figure 5. The data is processed by System Identification Tool in Matlab. Obtained expression for control-to-output transfer function $G_{vd}(s)$ is shown in Table 3 and Bode plot is shown in Figure 7.

Buck converter open loop line-to-output transfer function

$$G_{vg}(s) = \frac{v(s)}{v_g(s)}$$

is determined experimentally by step change of the input voltage $v_g(t)$ and measuring the transient of output voltage

TABLE II. PARAMETERS AND VARIABLES OF SMALL-SIGNAL BLOCK DIAGRAM

| **Parameter** | **Description** |
|---|---|
| $v_{ref}$ | Reference input |
| $v_e$ | Error signal |
| $d(s)$ | PWM signal |
| $v_g(s)$ | Input voltage |
| $v(s)$ | Output voltage |
| $G_c(s)$ | Transfer function of PI controller |
| $G_{vd}(s)$ | Control-to-output transfer functions |
| $G_{vg}(s)$ | Line-to-output transfer functions |
| $H(s)$ | Sensor gain |
| $T(s)$ | Loop gain $G_{vd}(s)\ H(s)\ G_c(s)$ |

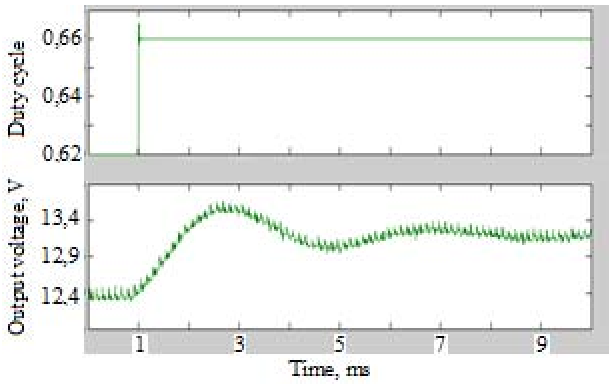


Figure 5 Unregulated output voltage step response on change of duty cycle $d(t)$

$v(t)$, Figure 6. Obtained expression for control-to-output transfer function $G_{vg}(s)$ is shown in Table 3. The shape of transfer function $G_{vg}(s)$ is the same as transfer function $G_{vd}(s)$ but they differ for constant gain.

Sensor gain of the output voltage divider is

$$H(s) = \frac{R_2}{R_1 + R_2}$$

Criteria for choosing the values of output voltage divider are not to exceed maximum analog input voltage of Arduino Uno (5 V) and not to load the output significantly. Chosen sensor gain is shown in Table 3 and in Figure 7.

Reference voltage $v_{ref} = 2{,}3$ V is chosen to be in the middle of the range of analog input voltage of Arduino Uno.

Transfer function of PI controller is

$$G_c(s) = \frac{d(s)}{v_e(s)}$$

The parameters of PI controller are determined by the Bode plots using Simulink. PI controller parameters are chosen to

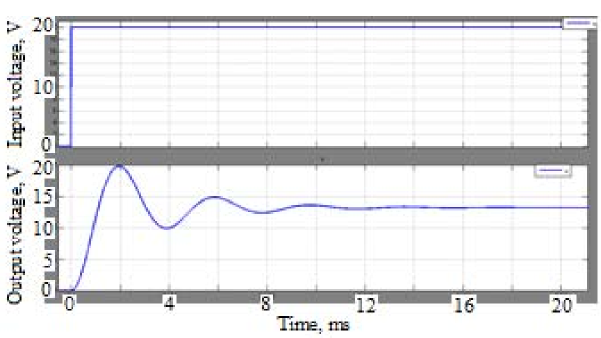


Figure 6 Free response of unregulated output voltage on step change of input voltage $v_g(t)$

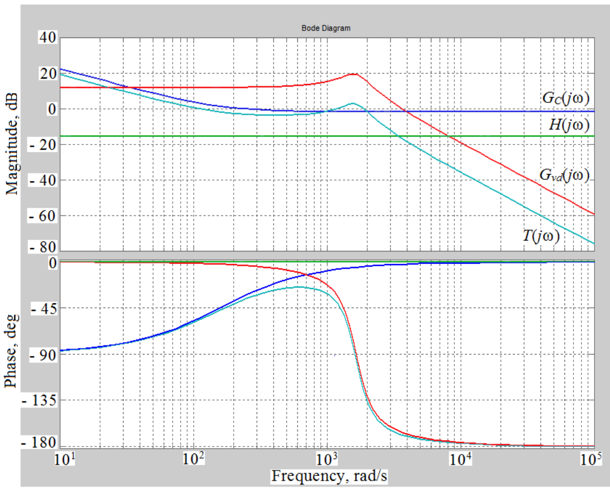


Figure 7 Bode plots of $G_{vd}(s)$, $H(s)$, $G_c(s)$ and $T(s)$

achieve phase margin of desired loop gain

$$T(s) = G_{vd}(s)H(s)G_c(s)$$

about 60°. PI controller's crossover frequency (determining the $G_c(s)$) is chosen to be 5-10 times lower than crossover frequency of the buck converters open loop system (derived from $G_{vd}(s)$). Obtained expression for transfer function of PI controller $G_c(s)$ is shown in Table 3. Bode plots of $G_{vd}(s)$, $H(s)$, $G_c(s)$ and $T(s)$ are shown in Figure 7.

## IV. Verification by Simulation

Simulation is done in Simulink. The simulation model is shown in Figure 8. Input voltage step response for unregulated and regulated output voltage is simulated.

Free response of regulated output voltage on step change of input voltage $v_g(t)$ from zero to full value ($E = 20$ V) is shown in Figure 9. Comparing this results to the unregulated output voltage response in Figure 6 it is visible that feedback prevents overshoot of output voltage.

TABLE III. Expressions for parametrers and variables of small-signal block diagram

| Parameter | Expression |
|---|---|
| $v_{ref}$ | 2,3 V |
| $G_c(s)$ | $G_c(s) = 0{,}836 \cdot \frac{6{,}33 \cdot 10^{-3} s + 1}{6{,}33 \cdot 10^{-3} s}$ |
| $G_{vd}(s)$ | $G_{vd}(s) = \frac{3{,}99}{3{,}65 \cdot 10^{-7} s^2 + 260 \cdot 10^{-6} s + 1}$ |
| $G_{vg}(s)$ | $G_{vd}(s) = \frac{0{,}66}{3{,}65 \cdot 10^{-7} s^2 + 260 \cdot 10^{-6} s + 1}$ |
| $H(s)$ | 0,175 |

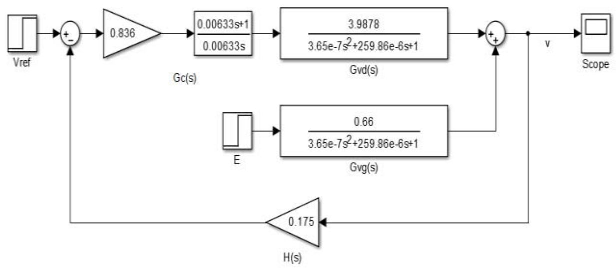


Figure 8 Simulation model for step changes of input voltage

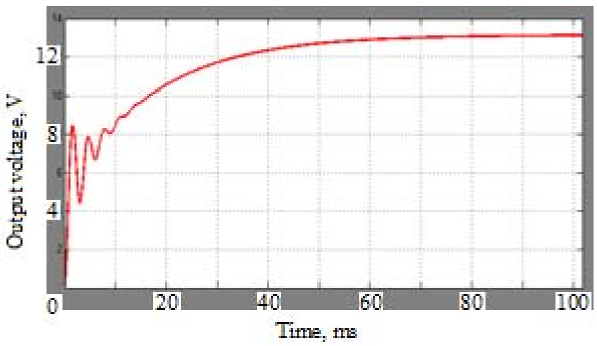


Figure 9 Free response of regulated output voltage on step change of input voltage $v_g(t)$

Simulation is also done for the case when converter is in steady state and there is increase of input voltage for 1 V. In this case, the waveforms of output voltages are shown in Figure 10 and Figure 11.

In a case of unregulated output voltage the steady-state error is significant and it can be calculated by multiplying change of input voltage by duty cycle. In the case of regulated output voltage the steady-state error is much smaller. Some of the parameters of quality of the transition process in the time domain read off from Figure 11 are shown in Table 4.

## V. CONCLUSION

The buck converter controlled by Arduino Uno microcontroller board is realized. The converter control is implemented by PI controller which is built in Arduino Uno board. PI controller is synthesized to accomplish desired closed loop gain. Simulation of closed loop line-to-output step response validated proposed design. Impact of proportional and integral parameters of PI regulator on converter control is significant. In a future work, proposed design will be validated by measurements and compared with analog PI control.

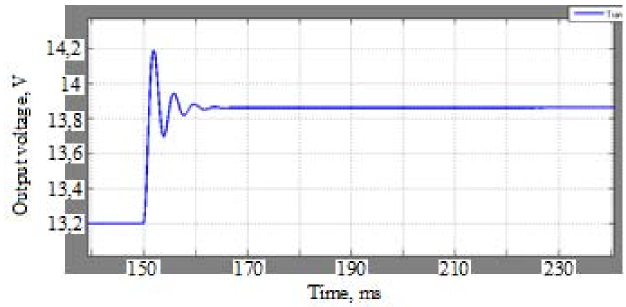


Figure 10 Unregulated output voltage by step of input voltage for 1 V

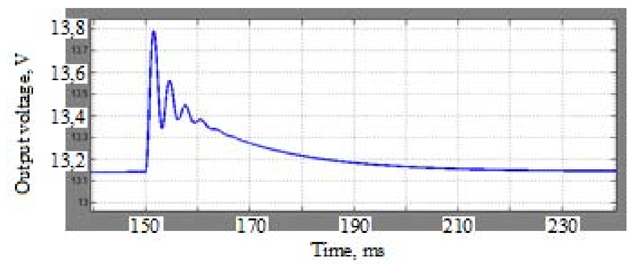


Figure 11 Regulated output voltage by step of input voltage for 1 V

TABLE IV. PARAMETERS OF THE QUALITY OF THE TRANSITION PROCESS IN THE TIME DOMAIN

| Parameter | Overshoot, V | Peak time, ms | Settling time ms |
|---|---|---|---|
| Value | 0,65 | 1,5 | 90 |

## REFERENCES

[1] R. W. Erickson, D. Maksimović, Fundamentals of Power Electronics. 2nd ed. Kluwer, 2001

[2] I. Flegar, Elektronički energetski pretvarači, Kigen, Zagreb, 2010

[3] K. Wan, Advanced current-mode control techniques for dc-dc power electronic converters, Missouri university of science & technology, PhD Thesis, 2009

[4] P. Cortes, M. P. Kazmierkowski, R. M. Kennel, D. E. Quevedo, J. Rodríguez, Predictive Control in Power Electronics and Drives, IEEE Trans. on ind. el., vol. 55, no. 12, pp. 4312-4324, December 2008

[5] S. Verma, S. K. Singh, A. G. Rao, Overview of control Techniques for DC-DC converters, Res. J. Engineering Sci., vol. 2, no. 8, pp. 18-21, August 2013

[6] F. Tahri, A. Tahri, A. Allali, S. Flazi, The Digital Self-Tuning Control of Step a Down DC-DC Converter, Acta Polytechnica Hungarica, vol. 9, no. 6, 2012

[7] A. Sinha, M. Pavithra, K. R. Sutharshan, S. K. Sahoo, Arduino based pulse width modulated output voltage control of a dc-dc boost converter using Proportional, Integral and Derivative control strategy, Australian Journal of Basic and Applied Sciences, vol. 7, no. 11, pp. 104-108, September 2013,

[8] S. Dutta, S. Agarwal, S. Gadgil, J. N. Hemalatha, A Flexible and Reconfigurable Controller Design for Buck Conveters, AICERA 2014 - International Conference on Magnetics, Machines & Drives, July 2014

[9] Arduino Uno datasheet

[10] Atmega 328 datasheet

## APPENDIX

```
#include <PID_v1.h>
#include <TimerOne.h>   // Including library

#define output_voltage A1
#define PWM 9

// Variable declaration
double Kp=0.836, Ki=132, Kd=0; // PI controller
parameters
double
output=0,desired_voltage=13.2,voltage=0,duty_cycle=0,

PID myPID(&napon, &output, &desired_voltage, Kp, Ki,
Kd, DIRECT); // Initialization of PID controller

void setup() {
  pinMode(PWM,OUTPUT);
  Timer1.initialize(125); // Frequency 8 kHz
  myPID.SetOutputLimits(0 , 1023); // PID controller limits
  myPID.SetSampleTime(0.0735); // PID sample time
  myPID.SetMode(AUTOMATIC);
}

void loop() {
   myPID.Compute();
   Timer1.pwm(PWM,output); // Duty cycle set up

   //Measuring output voltage
   voltage =((analogRead(output_voltage)/1023.)*5)/0.21;
}
```