



# AERO SHIELD: MINIATÚRNY EXPERIMENTÁLNY MODUL AEROKYVADLA

Bakalárska práca

Vedúci práce: Ing. Mgr. Anna Vargová

Konzultant: Ing. Erik Mikuláš

Oponent: prof. Ing. Gergely Takács, PhD.

## Základný opis

- AeroShield - polohovateľné kyvadlo poháňané motorom.

Spätná väzba realizovaná pomocou:

- Rotačného enkóderu - bezkontaktné meranie uhlu kyvadla pomocou Hallovho senzora.

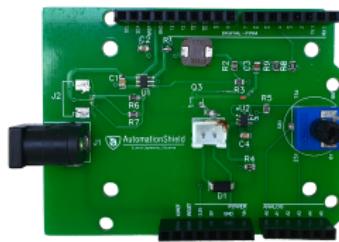
Súčasť projektu AutomationShield [1].



Obr. 1: Logo AutomationShieldu [1].

## Verzie AeroShieldu

- R1 - záverečný projekt na predmet Mikropočítače a mikroprocesorová technika.
- R2 - oprava chýb dosky R1, zmena zapojenia a rozloženia komponentov.
- R3 - vylepšenie verzie R2 - zmena spôsobu napájania.



(a) R1.



(b) R2.



(c) R3.

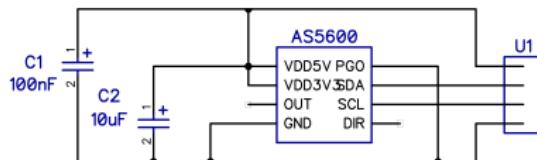
Obr. 2: Verzie AeroShieldu.

# Zmeny oproti pôvodnej verzii AeroShieldu R1

- Odstránenie napájacieho konektora na 12 V, z hlavnej PCB dosky
- Zmena schémy zapojenia znižovacieho meniča
- Vytvorenie komunikačného prepojenia senzora AS5600 s hlavnou PCB doskou
- Zmena v schéme zapojenia mosfetu
- Vytvorenie nového designu držiaku kyvadla
- Tvorba softvérového rozhrania AeroShieldu

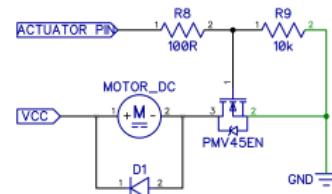
# Základné hardvérové komponenty AeroShieldu

## Rotačný enkóder



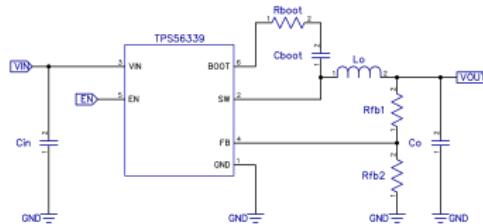
Obr. 3: Schéma zapojenia snímača Hallovho javu.

## Akčný člen



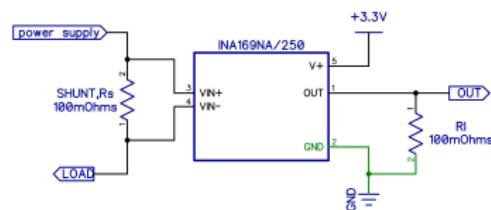
Obr. 5: Schéma zapojenia motorčeka.

## Znižovací menič



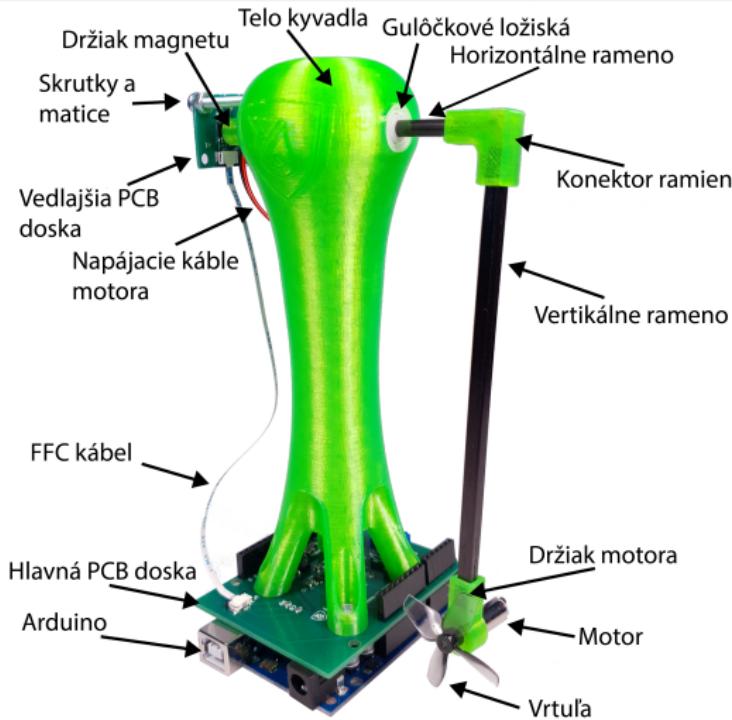
Obr. 4: Schéma zapojenia znižovacieho meniča.

## Monitor prúdu



Obr. 6: Schéma zapojenia monitoru prúdu. ↗ ↘ ↙ ↘

# Model AeroShieldu



Obr. 7: Zostavený model AeroShieldu.

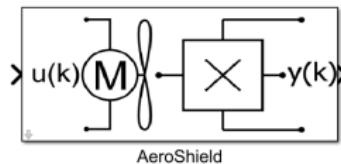
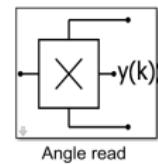
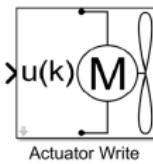
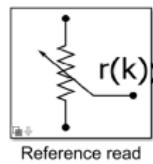
# Softvér

Vytvorenie knižníc pre API

- Arduino IDE
- MATLAB
- Simulink

Najdôležitejšie funkcie AeroShieldu

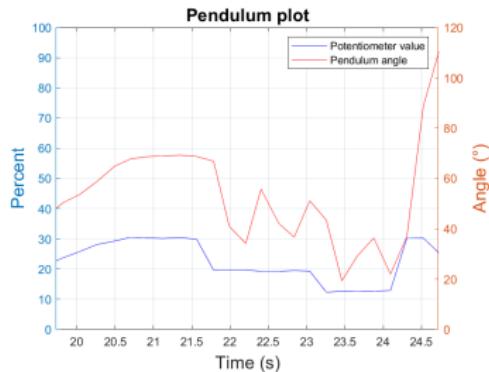
- Reference read
- Actuator write
- Angle read
- Calibration



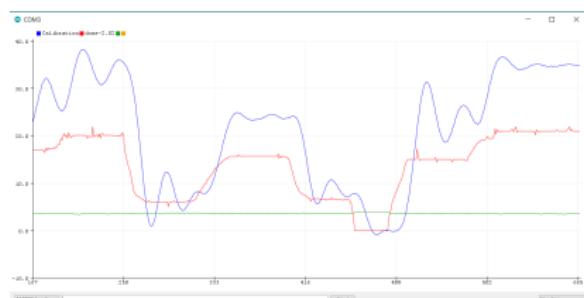
Obr. 8: Knižnica AeroLibrary Simulink.

## Bez spätej väzby - otvorená slučka

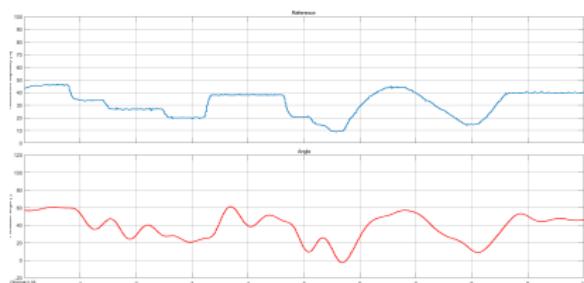
Príklady s otvorenou slučkou  
boli vytvorené pre Arduino  
IDE Obr.10, MATLAB  
Obr.9, ako aj pre Simulink  
Obr.11.



Obr. 9: Výstup MATLAB.

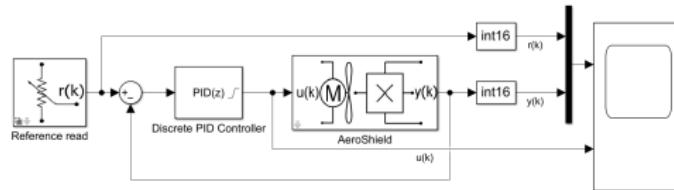


Obr. 10: Výstup Arduino IDE.

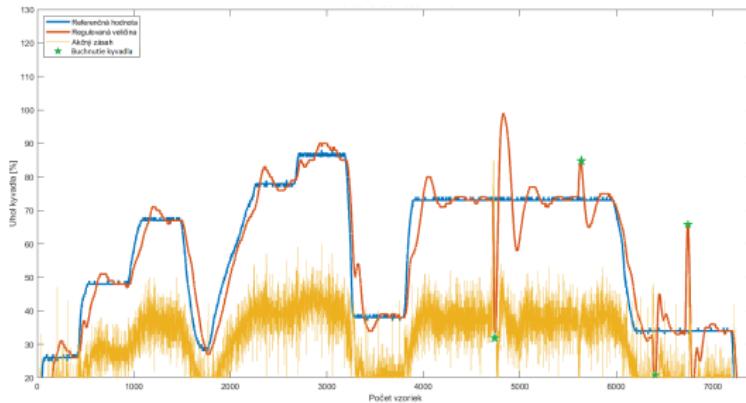


Obr. 11: Výstup Simulink.

## So spätnou väzbou - PID

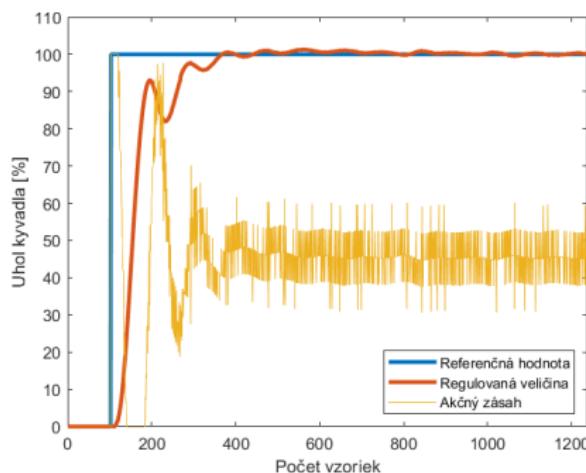


Obr. 12: Príklad PID regulácie - Simulink.

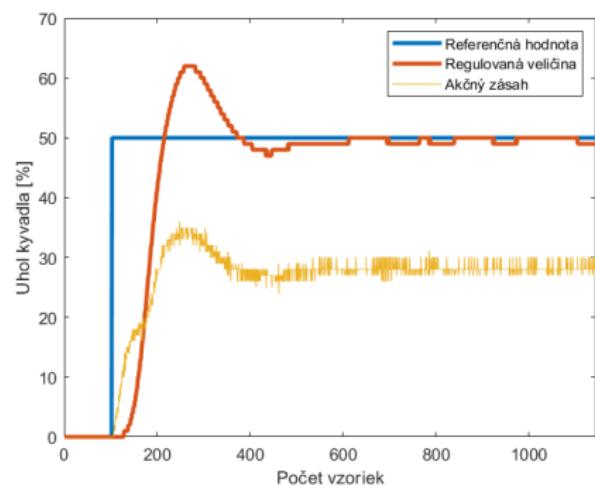


Obr. 13: Výstup Simulink- manuálna trajektória.

# Skoková zmena referenčnej hodnoty



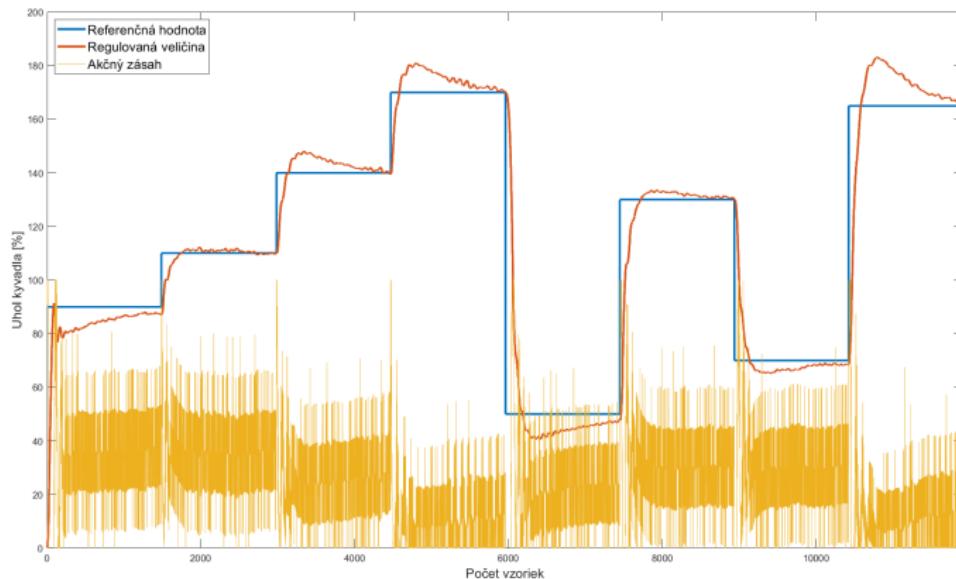
Obr. 14: Výstup Arduino IDE.



Obr. 15: Výstup Simulink.

## Možnosti regulácie

- Možnosť regulácie do  $165^\circ$  s výraznejším preregulovaním ( $175^\circ$  pri postupnom zvyšovaní referenčnej hodnoty).



Obr. 16: Výstup Arduino IDE.

# Prínos a návrhy na vylepšenia

## Prínos

- Vytvorenie učebnej pomôcky z dostupných komponentov, za nízku cenu
- Vytvorenie API Arduino IDE, MATLAB, Simulink
- Funkčné didaktické príklady, slúžiace pre jednoduchšie zoznámenie s AeroShieldom

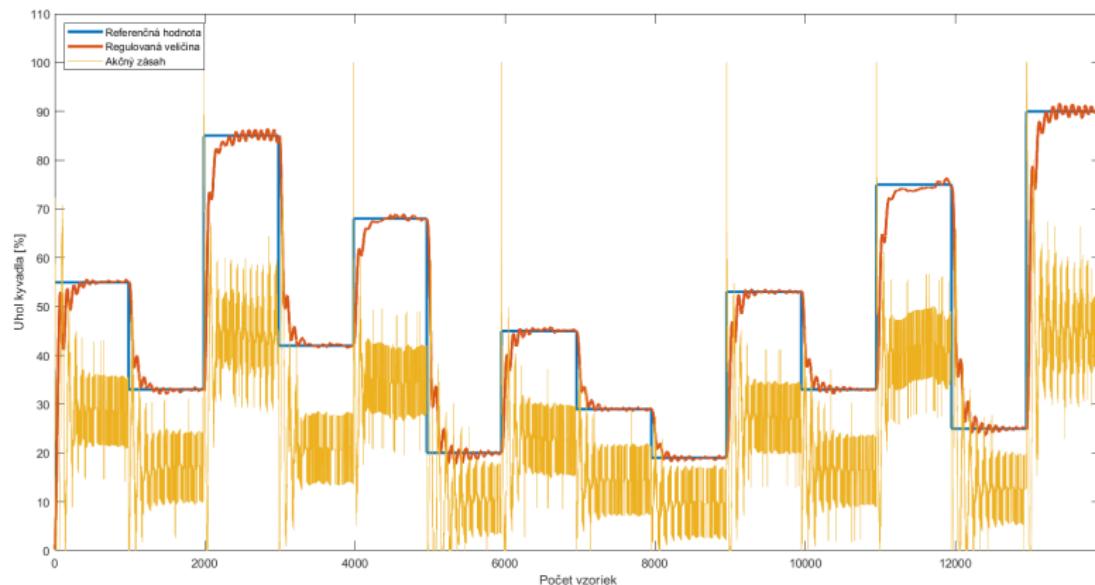
## Návrhy na vylepšenia

- Vylepšenie monitorovania odoberaného prúdu
- Zmena spôsobu napájania motora
- Použitie rôznych druhou algoritmov spätnoväzbového riadenia

Ďakujem za pozornosť

## 1. otázka

Ako by ste odstránili kmitavý priebeh viditeľný na Obr. 17?



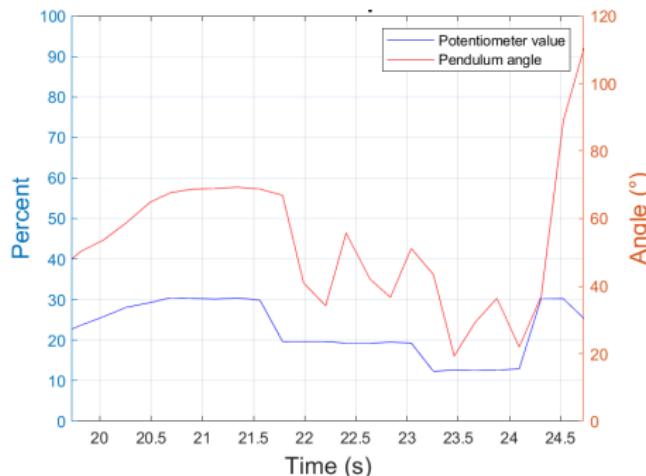
Obr. 17: Automatická trajektória Arduino IDE.

## Odpoveď

- Znížením proporcionálnej zložky  $K_p$
  - Znížením integračnej časovej zložky  $T_i$
  - Zvýšením derivačnej časovej zložky  $T_d$
- alebo/aj
- Skrátením periódy vzorkovania

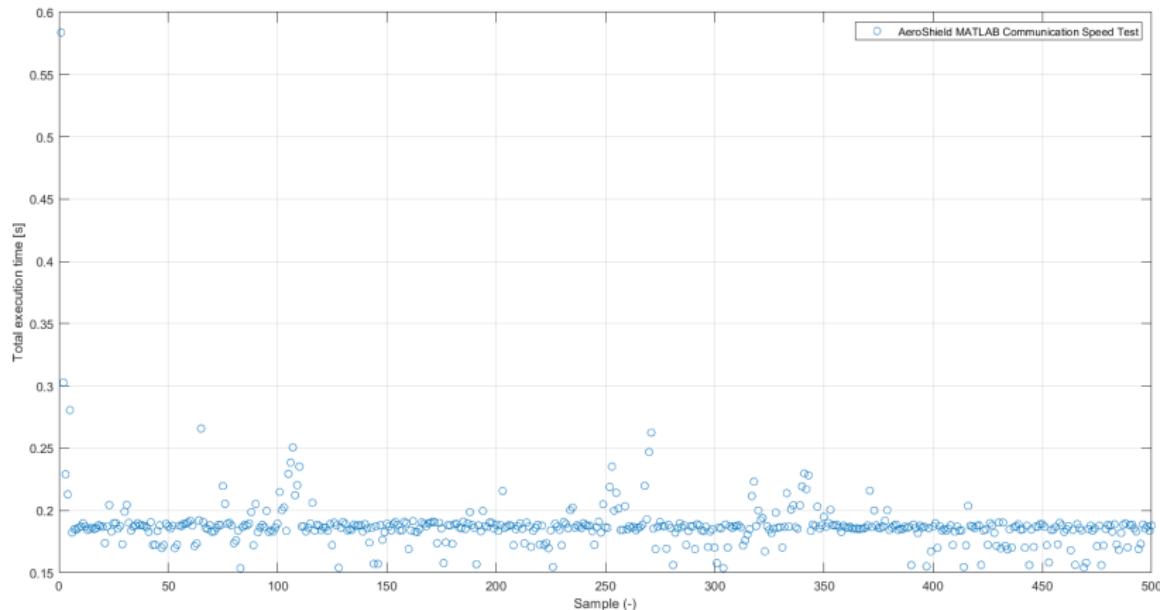
## 2. otázka

V prípade PID riadenia podľa kódu na s. 43 ste používali vzorkovanie  $T = 3$  ms. Aké bolo vzorkovanie v prípade MATLAB a Simulink? Prečo vyzerá výstup na Obr. 18 podvzorkované?



Obr. 18: Výstup MATLAB- OpenLoop.

# Odpoved'



### 3. otázka

Z hľadiska vzorkovania, považujete implementáciu aspoň PID riadenia v interpretovanom jazyku Python za reálne? (napr. na procesore ATSAMD51 @ 160 MHz)

# Odpoveď

ATSAMD51

- 160 MHz

ATSAM3X8E

- 84 MHz

ATmega328P

- 16 MHz

ATmega2560

- 16 MHz

## 4. otázka

**Čo by ste ešte rád dokončili alebo vylepšili z a.) hardvérového b.) softvérového c.) dokumentačného hľadiska ak by ste mali viac času?**

# Odpoveď

## Hardvér

- Použitie konektora so zbernými krúžkami
- Zmena designu kyvadla

## Softvér

- Filtrácia meranej hodnoty odoberaného prúdu
- Zrýchlenie periódy vzorkovania v MATLABE
- Identifikácia sústavy

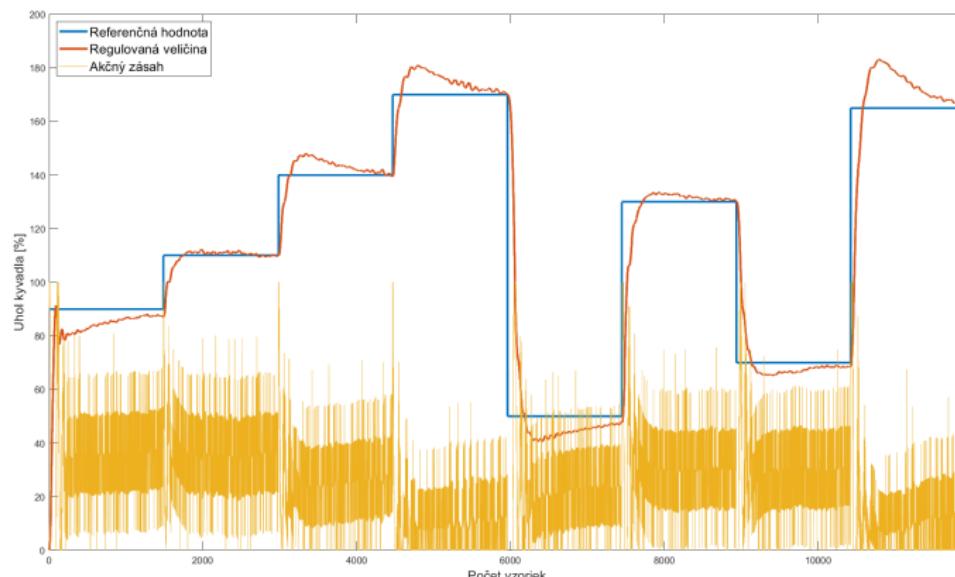
## Dokument

- Lepší opis a porozumenie OOP
- LQ, MPR riadenie

## 5. otázka

**Vysvetlite ako by ste implementovali fuzzy PID riadenie AeroShieldu.**

# Odpoveď



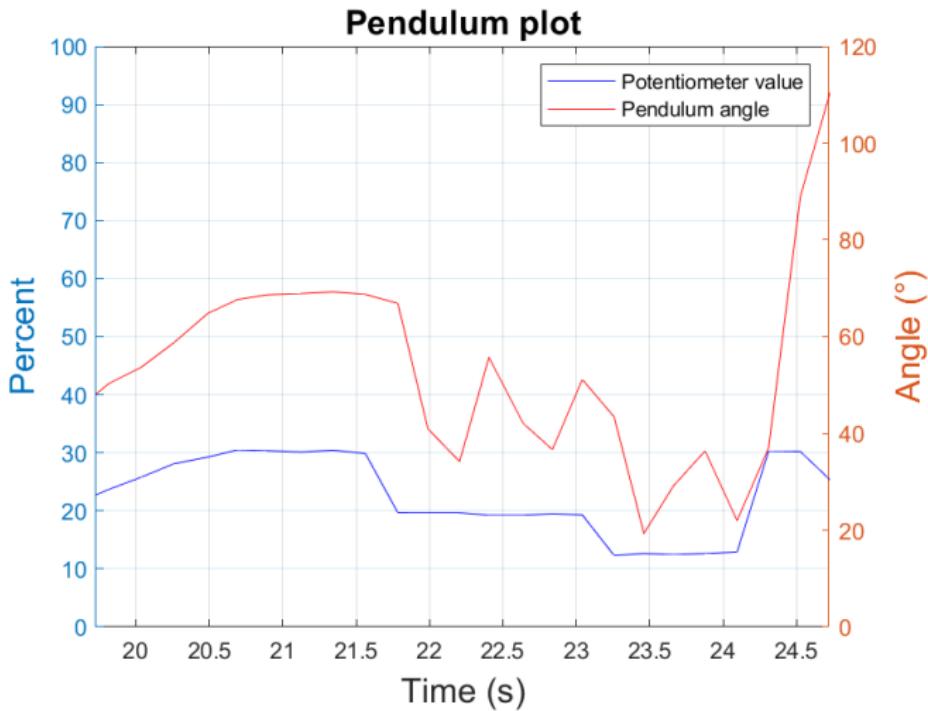
Obr. 19: Výstup Arduino IDE.

- FUZZY PID CONTROLLER FOR PROPELLER PENDULUM [2]

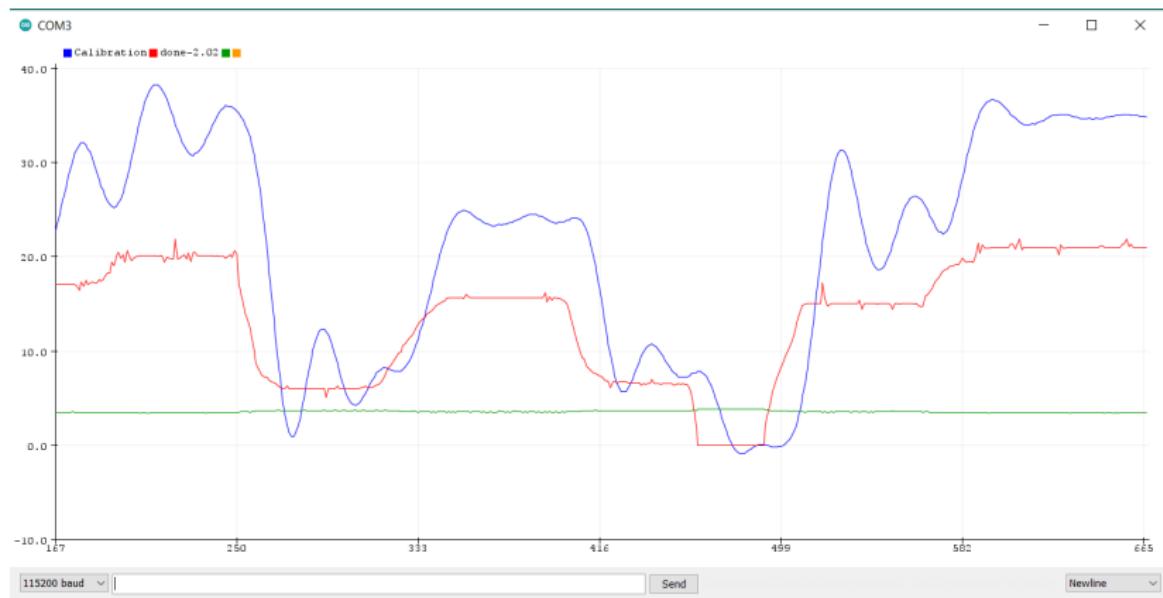
 Gergely Takacs.  
Automationshield.  
Code. Online., 2021.  
23.12.2021,  
<https://github.com/gergelytakacs/AutomationShield>.

 Yener Taskin.  
Fuzzy pid controller for propeller pendulum.  
*Istanbul University - Journal of Electrical and Electronics Engineering*, 17:3201–3207, 2017.

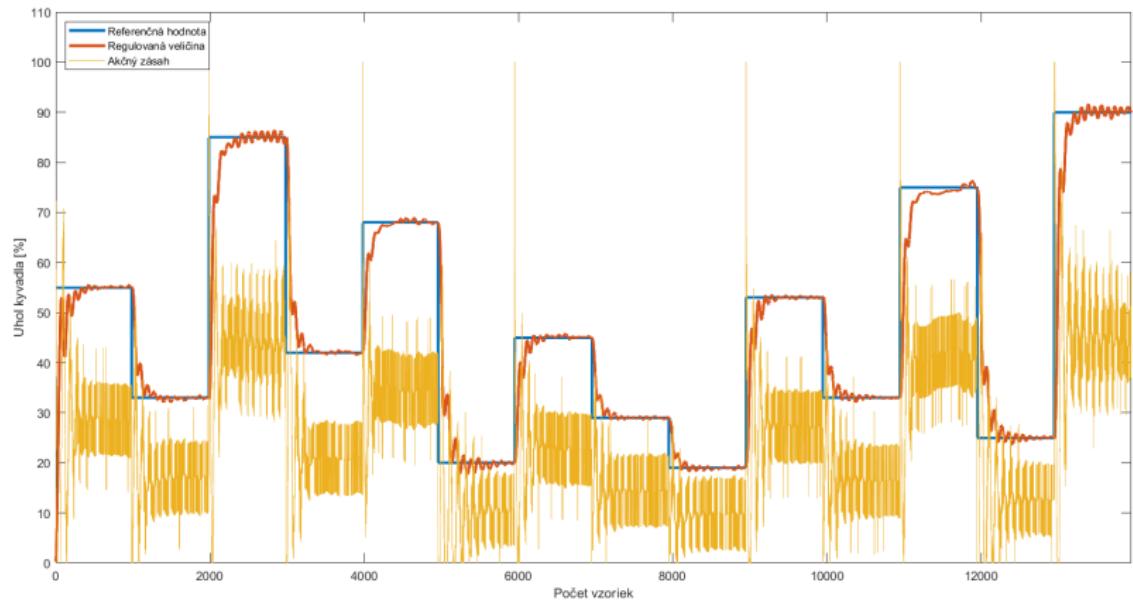
# Výstup MATLAB OL



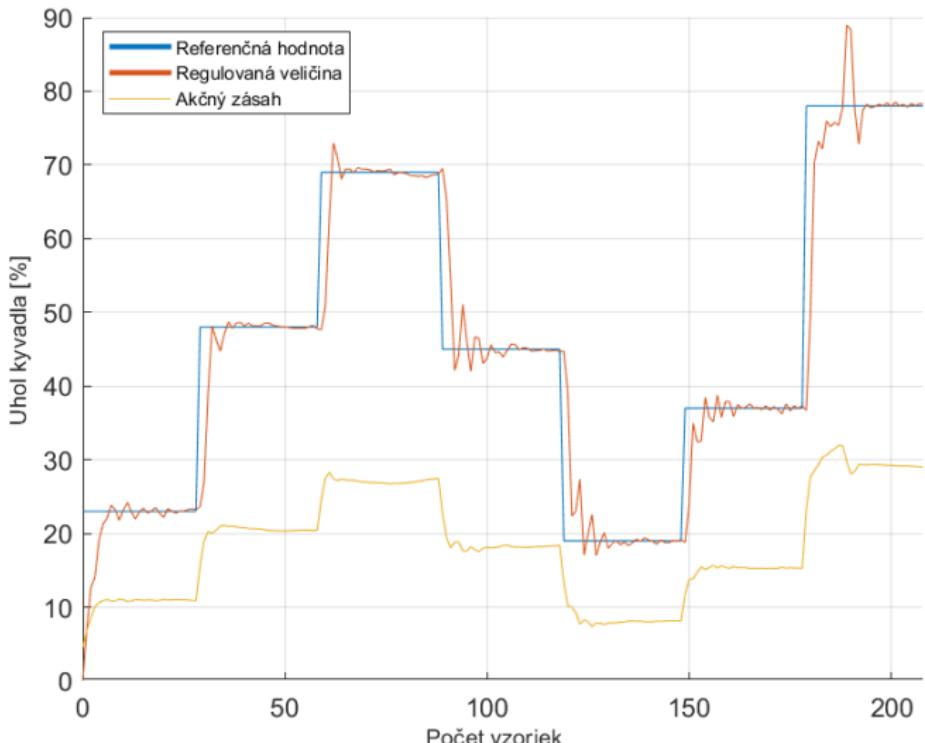
# Výstup Arduino IDE OL



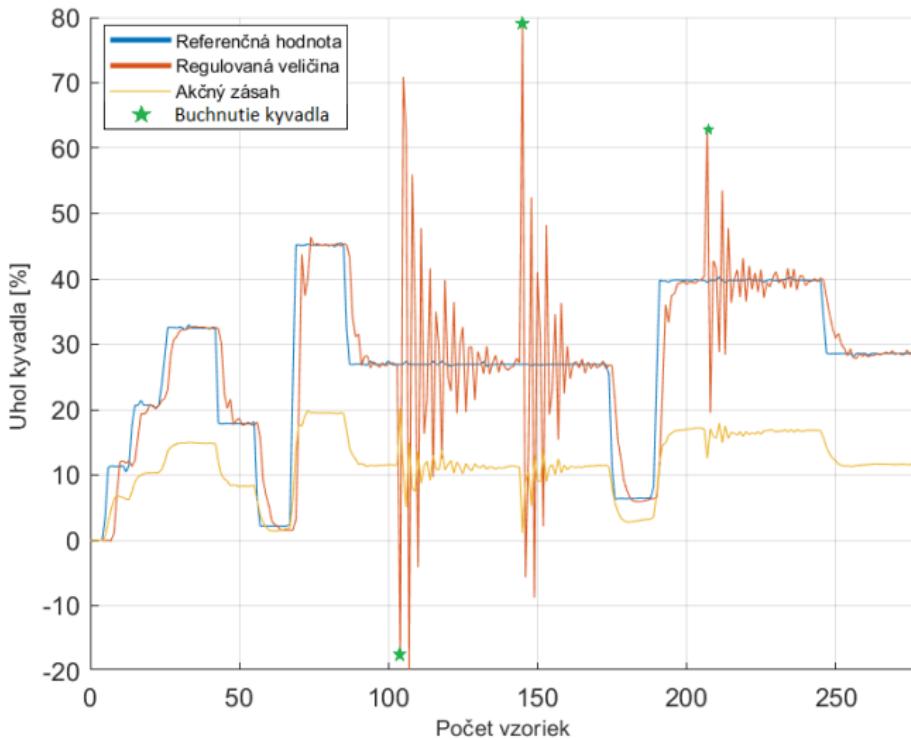
# Výstup Arduino IDE PID



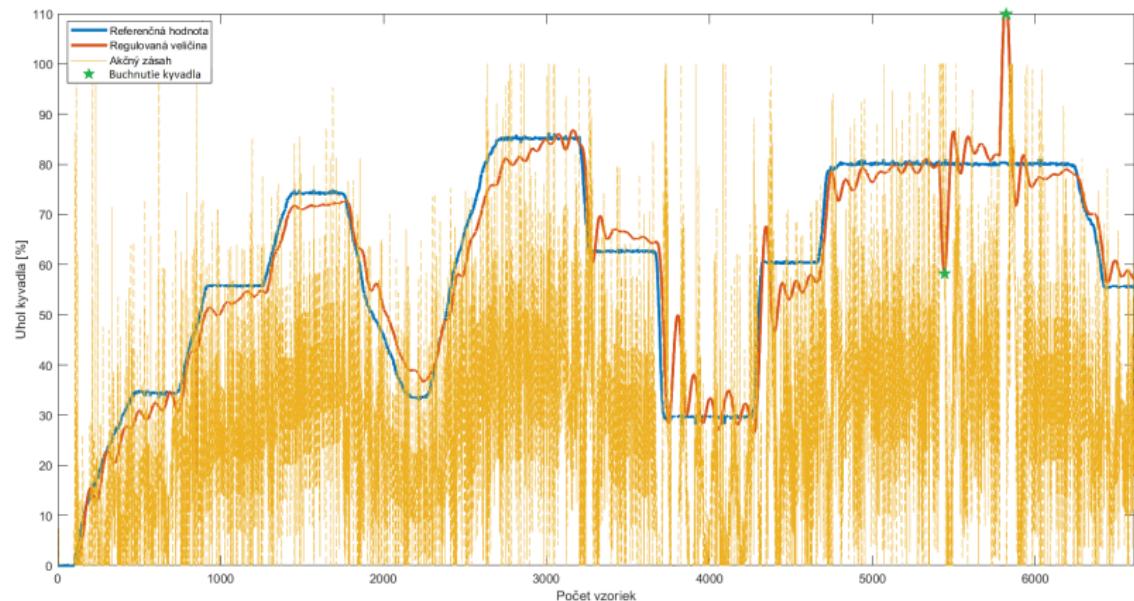
# Výstup MATLAB PID



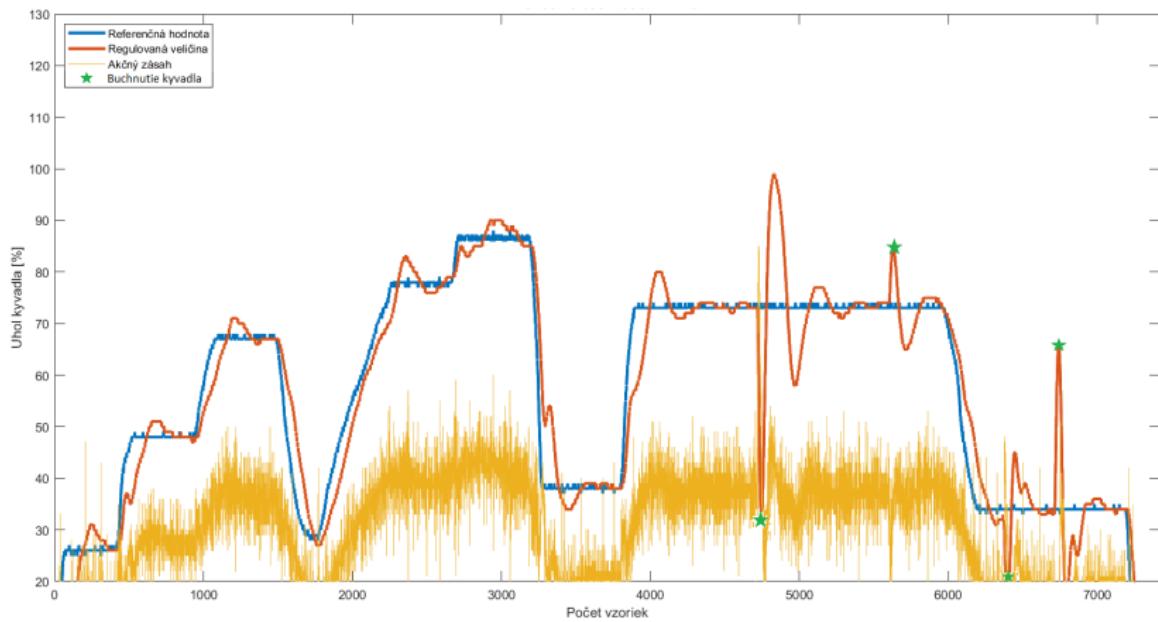
# Výstup MATLAB PID



# Výstup Arduino IDE PID manual



# Výstup Simulink PID



# Masky blokov

