



Politechnika
Śląska

POLITECHNIKA ŚLĄSKA
WYDZIAŁ AUTOMATYKI, ELEKTRONIKI I INFORMATYKI
KIERUNEK AUTOMATYKA I ROBOTYKA

Projekt inżynierski

Sprzętowa implementacja regulatora MPC

Autor: Szymon Zosgórnik

Kierujący pracą: dr hab. inż., prof. PŚ Jarosław Śmieja

Gliwice, styczeń 2020

Streszczenie

Lorem ipsum.

Spis treści

1	Wstęp	1
1.1	Motywacja projektu	1
1.2	Cel pracy	1
2	Idea regulatora MPC	2
2.1	Wstęp	2
2.2	Model obiektu	2
2.3	Kryterium jakości regulacji	2
2.4	Problem programowania kwadratowego	2
2.5	Pozostałe rodzaje regulatorów klasy MPC	2
2.6	Wady i zalety w porównaniu z regulatorem PID	2
3	Założenia projektowe i wykorzystane narzędzia	3
3.1	Założenia projektowe	3
3.2	Architektura systemu	3
3.2.1	Platforma STM	3
3.2.2	Procesor - architektura ARM	3
3.3	Narzędzia programistyczne	3
3.3.1	Języki programowania C/C++	3
3.3.2	Język programowania Python	3
3.3.3	Środowisko MATLAB	3
3.3.4	Biblioteka HAL	3
3.3.5	CMake	3
3.3.6	Kompilator i linker	4
3.4	Przykład referencyjny	4
3.5	Sposób testowania	4
4	Implementacja rozwiązania	5
4.1	Ogólny schemat programu	5
4.2	Problemy napotkane podczas realizacji	5
5	Przykładowe wyniki	6
5.1	Dużo wyników	6
6	Podsumowanie	7
6.1	Wyniki	7
6.2	Wnioski	7
6.3	Pomysły na rozwój projektu	7
	Dodatki	8
A	Jak zrobię jakieś fajen porównania to tu dam	9

Rozdział 1. Wstęp

1.1 Motywacja projektu

Lorem ipsum.

1.2 Cel pracy

Lorem ipsum.

Rozdział 2. Idea regulatora MPC

2.1 Wstęp

2.2 Model obiektu

Lorem ipsum.

2.3 Kryterium jakości regulacji

Lorem ipsum.

2.4 Problem programowania kwadratowego

Lorem ipsum.

2.5 Pozostałe rodzaje regulatorów klasy MPC

2.6 Wady i zalety w porównaniu z regulatorem PID

Rozdział 3. Założenia projektowe i wykorzystane narzędzia

3.1 Założenia projektowe

liniowy układ

3.2 Architektura systemu

Lorem ipsum. Cokolwiek o STMie / ARMie.

3.2.1 Platforma STM

Lorem ipsum.

3.2.2 Procesor - architektura ARM

Lorem ipsum.

3.3 Narzędzia programistyczne

3.3.1 Języki programowania C/C++

A gdzie Rust?!

3.3.2 Język programowania Python

pytong

3.3.3 Środowisko MATLAB

matlablabla

3.3.4 Biblioteka HAL

hal

3.3.5 CMake

cmake

3.3.6 Kompilator i linker

arm none eabi gcc

3.4 Przykład referencyjny

Lorem ipsum.

3.5 Sposób testowania

Lorem ipsum.

Rozdział 4. Implementacja rozwiązania

4.1 Ogólny schemat programu

Todooo.

4.2 Problemy napotkane podczas realizacji

Ło panie.

Rozdział 5. Przykładowe wyniki

5.1 Dużo wyników

Wiyncyj wyników.

Rozdział 6. Podsumowanie

6.1 Wyniki

No działa.

6.2 Wnioski

Jak wyżej.

6.3 Pomysły na rozwój projektu

Jak wyżej.

Dodatki

Dodatek A. Jak zrobię jakieś fajen porównania to tu dam

Lorem ipsum.

Spis rysunków

Spis tablic

Spis listingów

Bibliografia

- [1] Model predictive control. https://en.wikipedia.org/wiki/Model_predictive_control, 04.01.2020.
- [2] Power method. http://ergodic.ugr.es/cphys/LECCIONES/FORTRAN/power_method.pdf, 04.01.2020.
- [3] Understanding model predictive control. <https://www.mathworks.com/videos/series/understanding-model-predictive-control.html>, 04.01.2020.
- [4] Stm32 nucleo-64 boards (mb1136) user manual. https://www.st.com/content/ccc/resource/technical/document/user_manual/98/2e/fa/4b/e0/82/43/b7/DM00105823.pdf/files/DM00105823.pdf/jcr:content/translations/en.DM00105823.pdf, 04.2019.
- [5] Stm32f401xd stm32f401xe datasheet. <https://www.espruino.com/datasheets/STM32F401xD.pdf>, 2015.
- [6] G. Wang et al. State-space model predictive control method for core power control in pressurized water reactor nuclear power stations. *Nuclear Engineering and Technology*, strony 3–4, 2016.
- [7] Rolf Findeisen Markus Kögel. A fast gradient method for embedded linear predictive control. *Proceedings of the 18th World Congress The International Federation of Automatic Control*, strony 1362–1367, 28.08 - 02.09.2011.