



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

<b>1</b>	<b>Wstęp</b>	<b>1</b>
1.1	Motywacja projektu . . . . .	1
1.2	Cel pracy . . . . .	1
<b>2</b>	<b>Idea regulatora MPC</b>	<b>2</b>
2.1	Wstęp . . . . .	2
2.2	Sposób działania . . . . .	2
2.3	Model obiektu . . . . .	3
2.4	Kryterium jakości regulacji . . . . .	4
2.5	Problem programowania kwadratowego . . . . .	4
2.6	Pozostałe rodzaje regulatorów klasy MPC . . . . .	4
2.7	Wady i zalety w porównaniu z regulatorem PID . . . . .	5
<b>3</b>	<b>Założenia projektowe i wykorzystane narzędzia</b>	<b>6</b>
3.1	Założenia projektowe . . . . .	6
3.2	Architektura systemu . . . . .	6
3.2.1	Platforma STM . . . . .	6
3.2.2	Procesor - architektura ARM . . . . .	6
3.3	Narzędzia programistyczne . . . . .	6
3.3.1	Języki programowania C/C++ . . . . .	6
3.3.2	Język programowania Python . . . . .	6
3.3.3	Środowisko MATLAB . . . . .	7
3.3.4	Cube . . . . .	7
3.3.5	Biblioteka HAL . . . . .	7
3.3.6	CMake . . . . .	7
3.3.7	Kompilator i linker . . . . .	7
3.3.8	Regex . . . . .	7
3.4	Przykład referencyjny . . . . .	7
3.5	Sposób testowania . . . . .	7
<b>4</b>	<b>Implementacja rozwiązania</b>	<b>8</b>
4.1	Ogólny schemat programu . . . . .	8

---

4.2	Szczegóły implementacji - STM . . . . .	8
4.2.1	Obiektowość . . . . .	8
4.2.2	Komunikacja . . . . .	8
4.2.3	Aglorytm . . . . .	8
4.3	Szczegóły implementacji - PC . . . . .	8
4.4	Problemy napotkane podczas realizacji . . . . .	8
<b>5</b>	<b>Przykładowe wyniki</b>	<b>9</b>
5.1	Różne parametry układu . . . . .	9
5.2	Różne parametry regulatora . . . . .	9
5.3	Różne wartości zadane . . . . .	9
<b>6</b>	<b>Podsumowanie</b>	<b>10</b>
6.1	Wyniki . . . . .	10
6.2	Wnioski . . . . .	10
6.3	Pomysły na rozwój projektu . . . . .	10
	<b>Dodatki</b>	<b>11</b>
<b>A</b>	<b>Porównanie matlab stm</b>	<b>12</b>

# **Rozdział 1. Wstęp**

## **1.1 Motywacja projektu**

Lorem ipsum.

## **1.2 Cel pracy**

Lorem ipsum.

# Rozdział 2. Idea regulatora MPC

## 2.1 Wstęp

Model predictive control (MPC) jest to zaawansowana metoda sterowania, która polega na takim dobraniu sterowania, aby spełniało ono szereg ograniczeń. Od lat 80. XX wieku algorytm ten wykorzystywany jest w przemyśle procesowym w zakładach chemicznych i rafineriach ropy naftowej. W ostatnich latach MPC znalazło zastosowanie także w elektrowniach i elektronice mocy. Sterowanie predykcyjne wykorzystuje dynamiczny model obiektu, najczęściej jest to empiryczny model pozyskany za pomocą identyfikacji systemów. Główną zaletą MPC jest optymalizacja obecnego przedziału czasowego, biorąc pod uwagę przyszłe stany obiektu. Jest to osiągnięte poprzez optymalizację skończonego horyzontu czasowego, ale z wykorzystaniem jedynie sterowania wyliczonego dla obecnej chwili czasu. Proces ten jest powtarzany z każdą iteracją algorytmu rozwiązującego układ równań różniczkowych opisujących dany układ. Taki schemat regulacji powoduje, że istnieje możliwość przewidzenia przyszłych zdarzeń (występujących zgodnie z podanym modelem wartości zadanej) i podjęcia odpowiednich działań regulujących pracę układem we wcześniejszych chwilach. Sterowanie predykcyjne jest zazwyczaj zaimplementowane jako dyskretny regulator, lecz obecnie prowadzone są badania mające na celu uzyskanie szybszej odpowiedzi przy użyciu specjalnie do tego przygotowanych układów analogowych.

## 2.2 Sposób działania

Zasada pracy regulatora MPC polega na minimalizacji różnic między wartościami predykowanymi:  $y(i + p|i)$  w chwili obecnej  $i$  na przyszłą  $i + p$ , a wartościami zadanymi dla tych wyjść  $r(i + p|i)$ . Przez minimalizację tychże różnic rozumiana jest minimalizacja określonego kryterium jakości  $J$ . W następnej chwili czasu  $(i + 1)$  następuje kolejny pomiar sygnału na wyjściu obiektu, a cała procedura powtarzana jest z takim samym horyzontem predykcji  $H(p = 1, 2, \dots, H_{max})$ . W tym celu stosowana jest więc zasada sterowania repetycyjnego bazującego na przesuwym horyzoncie czasu. W algorytmie regulacji MPC obecny jest także tzw. horyzont sterowania  $L$  (gdzie  $L \leq H$ ), po którego upływie przyrost sygnału sterującego wynosi zero. W ten

sposób zapewnione są własności całkujące układu regulacji predykcyjnej.

Algorytmy MPC cechują się następującymi wymogami i właściwościami:

- Wymaganie wyznaczenia wartości przyszłych sygnału sterującego.
- Sterowanie według zdefiniowanej trajektorii referencyjnej dla wielkości wyjściowej.
- Uwzględnienie przyszłych zmian wartości zadanej. Wcześniejsza reakcja regulatora na przyszłą zmianę wartości referencyjnej kompensuje negatywny wpływ opóźnienia na działanie układu.
- Stabilna regulacja obiektów, które nie są minimalnofazowe bez uwzględnienia tego faktu podczas syntezy regulatora.

Realizację metody sterowania predykcyjnego można zapisać w czterech następujących krokach:

1. Pomiar lub estymacja aktualnego stanu obiektu.
2. Obliczenie przyszłych próbek wyjść systemu.
3. Zaaplikowanie sygnałów sterujących tylko do następnej chwili czasu.
4. Powtórzenie algorytmu dla kolejnej chwili czasu.

## 2.3 Model obiektu

Do poprawności działania regulatora MPC niezbędna jest znajomość modelu obiektu, który ma byćysterowany. Obecnie wykorzystuje się model w postaci równań stanu, podczas gdy w przeszłości korzystano z modelu odpowiedzi skokowej. Takie podejście wymaga także zaprojektowania obserwatora stanu, używając do tego metod znanych z teorii sterowania. Model obiektu może być zarówno liniowy, jak i nieliniowy. Jednakże, użycie modelu nieliniowego prowadzi do nieliniowej optymalizacji, co powoduje zwielokrotnienie trudności obliczeniowej. Przekłada się to na zwiększenie wymagań odnośnie częstotliwości taktowania procesora w implementacji sprzętowej. Wynika z tego stwierdzenie, że modele liniowe mają największe znaczenie praktyczne z uwagi na możliwość przeprowadzenia obliczeń w czasie rzeczywistym nawet bez wygórowanych wymagań hardware'owych. Rozwiązaniem tego problemu jest zastosowanie regulatora predykcyjnego w połączeniu z linearyzacją modelu obiektu w konkretnym punkcie pracy. Następnie wyznaczone są sterowania

tak jak dla liniowego przypadku. Tak zrealizowany algorytm gwarantuje jedynie rozwiązanie suboptymalne, jednak nie rzutuje to w żaden sposób na przydatność jego realizacji.

## 2.4 Kryterium jakości regulacji

Jak już wcześniej pokazano w rozdziale 2.2 w celu wyznaczenia wartości sterowań w obecnej i następnych chwilach wyznacza się minimum funkcji celu. Funkcja ta określa jakość pracy regulatora na horyzoncie predykcji. Można stwierdzić, że wartość sygnału sterującego jest wyznaczana poprzez minimalizację wskaźnika jakości regulacji, który jest inherentnie związany z predykcją wyjścia obiektu.

W przypadku skalarnym funkcję celu można opisać następującym równaniem:

$$J = \sum_{i=1}^N w_{x_i} (r_i - x_i)^2 + \sum_{i=1}^N w_{u_i} u_i^2 \quad (2.1)$$

$x_i$  =  $i$ -ta zmienna sterowana

$r_i$  =  $i$ -ta zmienna referencyjna

$u_i$  =  $i$ -ta zmienna sterująca

$w_{x_i}$  = współczynnik wagowy zmiennej  $x_i$

$w_{u_i}$  = współczynnik wagowy  $u_i$

## 2.5 Problem programowania kwadratowego

$$R = R_1 * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

## 2.6 Pozostałe rodzaje regulatorów klasy MPC

- Nonlinear MPC
- Explicit MPC
- Robust MPC



## 2.7 Wady i zalety w porównaniu z regulatorem PID

Dorobić tabelę

Porównanie regulacji PID i MPC	Cecha	Regulator PID	Regulator predykcyjny
ograniczenia	brak informacji o ograniczeniach	ograniczenia uwzględnione w projekcie	wartość zadana bliska ograniczeniom
wartość zadana	wartość zadana daleka od ograniczeń	wartość zadana bliska ograniczeniom	optymalność sterowania
ograniczeniom	optymalność sterowania	nie ma charakteru optymalnego	sterowanie ma charakter optymalny
sterowanie	nie ma charakteru optymalnego	sterowanie ma charakter optymalny	liczba wejść i wyjść układu
sterowanie	ma charakter optymalny	liczba wejść i wyjść układu	jedno wejście i jedno wyjście
liczba wejść i wyjść układu	jedno wejście i jedno wyjście	wiele wejść i wiele wyjść	model matematyczny
model matematyczny	model matematyczny	nie jest konieczny	

# Rozdział 3. Założenia projektowe i wykorzystane narzędzia

## 3.1 Założenia projektowe

- Układ liniowy, deskryptywny
- Odpowiedź MCU w czasie rzeczywistym
- Poprawność wyliczonych wartości
- Opisany przypadek MPC do implementacji -> fast gradient method

## 3.2 Architektura systemu

### 3.2.1 Platforma STM

STM32F4 series of high-performance MCUs with DSP and FPU instructions

### 3.2.2 Procesor - architektura ARM

ARM, previously Advanced RISC Machine, originally Acorn RISC Machine, is a family of reduced instruction set computing (RISC) architectures for computer processors, configured for various environments.

## 3.3 Narzędzia programistyczne

### 3.3.1 Języki programowania C/C++

Paradygmaty, co, jak i dlaczego.

### 3.3.2 Język programowania Python

Wysokopoziomowe ułatwienie testowania z PC.

### **3.3.3 Środowisko MATLAB**

Sprawdzone środowisko do obliczeń macierzowych i układów dynamicznych.

### **3.3.4 Cube**

Generator kodu do biblioteki HAL.

### **3.3.5 Biblioteka HAL**

W miarę wysokopziomowe rozwiązanie jak na standardy mikroprocków, jednak zostały napotkane problemy.

### **3.3.6 CMake**

Cross platform make, możliwość kompilacji z różnych systemów operacyjnych.

### **3.3.7 Kompilator i linker**

arm none eabi gcc, coś o opensource

### **3.3.8 Regex**

Krótko o wyrażeniach regularnych.

## **3.4 Przykład referencyjny**

Ten z simulinka + testowanie w matlabie.

## **3.5 Sposób testowania**

Skrypt w Pythonie

# **Rozdział 4. Implementacja rozwiązania**

## **4.1 Ogólny schemat programu**

Przydałby się rysunek z przepływem informacji + jak działa STM w połączeniu z PC.

## **4.2 Szczegóły implementacji - STM**

### **4.2.1 Obiektowość**

C++, singleton, hermetyzacja danych, clean code

### **4.2.2 Komunikacja**

Ramka, schemat, itp

### **4.2.3 Aglorytm**

Jak zostało zrealizowane założenie

## **4.3 Szczegóły implementacji - PC**

Krótko o skrypcie w Pythonie.

## **4.4 Problemy napotkane podczas realizacji**

HAL - przerwania, ramka UARTa Model układu - feasibility

# **Rozdział 5. Przykładowe wyniki**

**5.1 Różne parametry układu**

**5.2 Różne parametry regulatora**

**5.3 Różne wartości zadane**

# **Rozdział 6. Podsumowanie**

## **6.1 Wyniki**

Krótkie podsumowanie

## **6.2 Wnioski**

Zgodność z założeniem

## **6.3 Pomysły na rozwój projektu**

Implementacja linearyzacji

## **Dodatki**

## **Dodatek A. Porównanie matlab stm**



# **Spis rysunków**

# Spis tablic

# **Spis listingów**

# Bibliografia

- [1] Model predictive control. [https://en.wikipedia.org/wiki/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](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](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.