

Organizacja pracy w grupie

Zespół 9 - Paweł Bugyi, Marcin Michalski, Krzysztof Pierczyk
2020

Projektowanie układów sterowania (projekt grupowy)

Agenda

1. Przygotowanie do projektu, przydział zadań
2. Implementacja
3. Wykonywanie eksperymentów
4. Opracowanie dokumentacji

Przygotowanie do projektu, przydział zadań

Wstępne przygotowania, przydział zadań

- Wykorzystane narzędzia (MATLAB, github, LaTeX, Jenkins, ...)

Wstępne przygotowania, przydział zadań

- Wykorzystane narzędzia (MATLAB, github, LaTeX, Jenkins, ...)
- Ściśle określona struktura drzewa projektowego

Wstępne przygotowania, przydział zadań

- Wykorzystane narzędzia (MATLAB, github, LaTeX, Jenkins, ...)
- Ściśle określona struktura drzewa projektowego
 1. kod źródłowy

Wstępne przygotowania, przydział zadań

- Wykorzystane narzędzia (MATLAB, github, LaTeX, Jenkins, ...)
- Ściśle określona struktura drzewa projektowego
 1. kod źródłowy
 2. konfiguracja projektu

Wstępne przygotowania, przydział zadań

- Wykorzystane narzędzia (MATLAB, github, LaTeX, Jenkins, ...)
- Ściśle określona struktura drzewa projektowego
 1. kod źródłowy
 2. konfiguracja projektu
 3. dokumentacja pomocnicza

Wstępne przygotowania, przydział zadań

- Wykorzystane narzędzia (MATLAB, github, LaTeX, Jenkins, ...)
- Ściśle określona struktura drzewa projektowego
 1. kod źródłowy
 2. konfiguracja projektu
 3. dokumentacja pomocnicza
 4. dokumentacja projektu

Wstępne przygotowania, przydział zadań

- Wykorzystane narzędzia (MATLAB, github, LaTeX, Jenkins, ...)
- Ściśle określona struktura drzewa projektowego
 1. kod źródłowy
 2. konfiguracja projektu
 3. dokumentacja pomocnicza
 4. dokumentacja projektu
 5. ...

Wstępne przygotowania, przydział zadań

```
> bin
> config
> data
> doc
> external
> include
> lib
> obj
> scripts
> spike
> src
> test
> tools
> web
◆ .gitignore
◆ .gitmodules
📄 install.sh
📄 LICENSE
📄 README.md
📄 SConstruct
```

Rysunek 1: Przykładowe drzewo projektowe

Wstępne przygotowania, przydział zadań

- Wykorzystane narzędzia (MATLAB, github, LaTeX, Jenkins, ...)
- Ściśle określona struktura drzewa projektowego
 1. kod źródłowy
 2. konfiguracja projektu
 3. dokumentacja pomocnicza
 4. dokumentacja projektu
 5. ...
- Stały wzorzec dokumentacji

- Napisanie kodu do zadania (MATLAB)

- Napisanie kodu do zadania (MATLAB)
- Przeprowadzenie testów (MATLAB)

- Napisanie kodu do zadania (MATLAB)
- Przeprowadzenie testów (MATLAB)
- Generacja sprawozdania (MATLAB, LaTeX)

Implementacja

1. Analiza teoretyczna

1. Analiza teoretyczna

- jaki efekt chcemy osiągnąć?

1. Analiza teoretyczna

- jaki efekt chcemy osiągnąć?
- jakie mamy ograniczenia?

1. Analiza teoretyczna

- jaki efekt chcemy osiągnąć?
- jakie mamy ograniczenia?
- jakich narzędzi możemy użyć?

1. Analiza teoretyczna

- jaki efekt chcemy osiągnąć?
- jakie mamy ograniczenia?
- jakich narzędzi możemy użyć?

2. Rozbicie zadania na podproblemy



1. Analiza teoretyczna

- jaki efekt chcemy osiągnąć?
- jakie mamy ograniczenia?
- jakich narzędzi możemy użyć?

2. Rozbicie zadania na podproblemy

1. Analiza teoretyczna
 - jaki efekt chcemy osiągnąć?
 - jakie mamy ograniczenia?
 - jakich narzędzi możemy użyć?
2. Rozbicie zadania na podproblemy
3. Ustalenie konwencji dotyczących tworzenia kodu

Implementacja

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Configuration %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
  
% Sampling period [s]  
SAMPLING_PERIOD = 0.5;  
  
% Length of the simulation  
SAMPLES_NUMBER = 200;  
  
% Desired output trajectory structure (@see make_trajectory function's d  
Y_zad_trajectory_struct.size = SAMPLES_NUMBER;  
Y_zad_trajectory_struct.steps = [10, 1.0];  
  
% Desired noise parameters (frequency in [Hz])  
Z_amplitude = 3;  
Z_frequency = 0.1;  
  
% DMC parameters  
DMC_struct.N = 180;  
DMC_struct.Nu = 180;  
DMC_struct.D = 180;  
DMC_struct.Dz = 1;  
DMC_struct.lambda = 0.0327;  
  
% Plotting process output  
PLOT_OUTPUT = true;  
  
% Plotting input values (U and Z)  
PLOT_INPUT = true;  
  
% Error printing  
PRINT_ERROR = true;  
  
% Saving plots to txt files  
SAVE_DATA = false;
```

Rysunek 2: Przykładowa sekcja konfiguracyjna

Implementacja

1. Analiza teoretyczna
 - jaki efekt chcemy osiągnąć?
 - jakie mamy ograniczenia?
 - jakich narzędzi możemy użyć?
2. Rozbicie zadania na podproblemy
3. Ustalenie konwencji dotyczących tworzenia kodu

Implementacja

1. Analiza teoretyczna
 - jaki efekt chcemy osiągnąć?
 - jakie mamy ograniczenia?
 - jakich narzędzi możemy użyć?
2. Rozbicie zadania na podproblemy
3. Ustalenie konwencji dotyczących tworzenia kodu
4. Ciągłe testowanie oprogramowania

Implementacja

1. Analiza teoretyczna
 - jaki efekt chcemy osiągnąć?
 - jakie mamy ograniczenia?
 - jakich narzędzi możemy użyć?
2. Rozbicie zadania na podproblemy
3. Ustalenie konwencji dotyczących tworzenia kodu
4. Ciągłe testowanie oprogramowania
 - testy jednostkowe

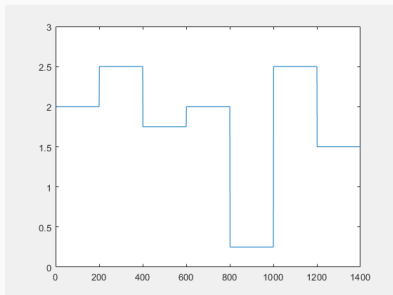
Implementacja

1. Analiza teoretyczna
 - jaki efekt chcemy osiągnąć?
 - jakie mamy ograniczenia?
 - jakich narzędzi możemy użyć?
2. Rozbicie zadania na podproblemy
3. Ustalenie konwencji dotyczących tworzenia kodu
4. Ciągłe testowanie oprogramowania
 - testy jednostkowe
 - scenariusze testowe

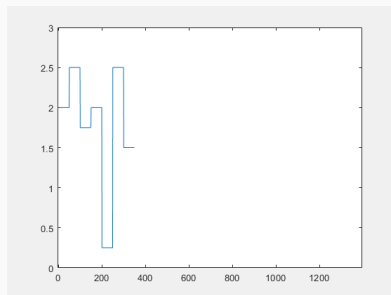
Wykonywanie eksperymentów



Przygotowanie scenariuszy testowych



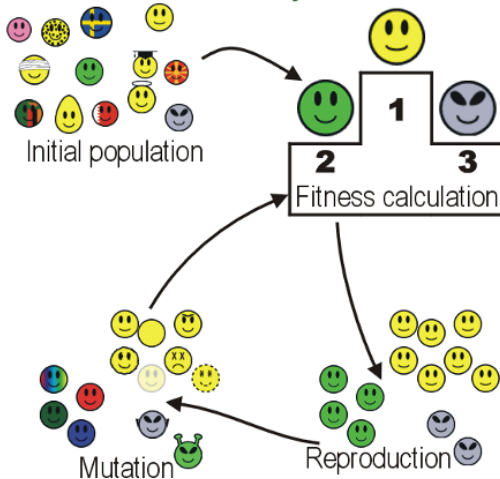
Dobry przebieg



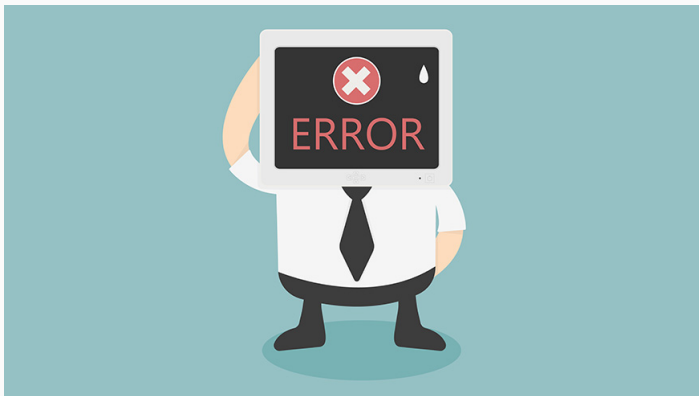
Zły przebieg

Zautomatyzowanie testowania

Evolutionary search



Przygotowanie się na potencjalne błędy podczas testowania



Podsumowanie części związanej z eksperymentami



Opracowanie dokumentacji

- Część teoretyczna sprawozdania

- Część teoretyczna sprawozdania
- Przygotowanie wykresów z danych wygenerowanych w MATLABie

- Część teoretyczna sprawozdania
- Przygotowanie wykresów z danych wygenerowanych w MATLABie
- Opis wyników

- Część teoretyczna sprawozdania
- Przygotowanie wykresów z danych wygenerowanych w MATLABie
- Opis wyników
- Złożenie sprawozdania

- Generowanie wykresów

- Generowanie wykresów
 1. Tworzenie skryptów automatyzujących generację plików pdf

- Generowanie wykresów
 1. Tworzenie skryptów automatyzujących generację plików pdf
 2. Tworzenie skryptów automatyzujących generację kodu tworzącego wykresy w dokumentach

- Generowanie wykresów
 1. Tworzenie skryptów automatyzujących generację plików pdf
 2. Tworzenie skryptów automatyzujących generację kodu tworzącego wykresy w dokumentach
 3. Formatowanie wykresów w dokumentach

- Generowanie wykresów - nazewnictwo

- Generowanie wykresów - nazewnictwo
 1. G1_D_180_N_60_Nu_25_lambda_25.txt

- Generowanie wykresów - nazewnictwo
 1. G1_D_180_N_60_Nu_25_lambda_25.txt
 2. G1 D=180, N=60, $N_u=25$, $\lambda=25$

- Tworzenie treści merytorycznej

- Tworzenie treści merytorycznej
 1. Analiza problemu

- Tworzenie treści merytorycznej
 1. Analiza problemu
 2. Opis teoretyczny potencjalnych rozwiązań

- Tworzenie treści merytorycznej
 1. Analiza problemu
 2. Opis teoretyczny potencjalnych rozwiązań
 3. Analiza wyników

- Tworzenie treści merytorycznej
 1. Analiza problemu
 2. Opis teoretyczny potencjalnych rozwiązań
 3. Analiza wyników
 4. Wnioski, zauważenie błędów, pomysły na poprawę w przyszłych projektach

- Formatowanie dokumentów

- Formatowanie dokumentów
 1. Dobór rozmiaru obrazów, dobór kolorystyki wykresów

- Formatowanie dokumentów
 1. Dobór rozmiaru obrazów, dobór kolorystyki wykresów
 2. Rozmieszczenie obrazów z uwzględnieniem zachowania stałej gęstości tekstu na stronie

Tworzenie kodu prostego w użyciu przez pozostałych członków zespołu bez zagłębiania się w implementację

- Tworzenie kodu prostego w użyciu przez pozostałych członków zespołu bez zagłębiania się w implementację

Tworzenie kodu prostego w użyciu przez pozostałych członków zespołu bez zagłębiania się w implementację

- Tworzenie kodu prostego w użyciu przez pozostałych członków zespołu bez zagłębiania się w implementację
 1. Dbanie o ścisłe komentowanie tworzonego oprogramowania

Tworzenie kodu prostego w użyciu przez pozostałych członków zespołu bez zagłębiania się w implementację

- Tworzenie kodu prostego w użyciu przez pozostałych członków zespołu bez zagłębiania się w implementację
 1. Dbanie o ścisłe komentowanie tworzonego oprogramowania
 2. Tworzenie intuicyjnego interfejsu