



**Politechnika
Śląska**

PROJEKT INŻYNIERSKI

Analiza sposobu uderzenia pałeczką w klepkę na stanowisku do gry na
cymbałkach

Piotr DINGES

Nr albumu: 300717

Kierunek: Automatyka i Robotyka

Specjalność: Technologie Informacyjne

PROWADZĄCY PRACĘ

dr hab. inż. Krzysztof Stebel

KATEDRA Katedra Automatyki i Robotyki

Wydział Automatyki, Elektroniki i Informatyki

Gliwice 2024

Tytuł pracy

Analiza sposobu uderzenia pałeczką w klepkę na stanowisku do gry na cymbałkach

Streszczenie

(Streszczenie pracy – odpowiednie pole w systemie APD powinno zawierać kopię tego streszczenia.)

Słowa kluczowe

(2-5 słów (fraz) kluczowych, oddzielonych przecinkami)

Thesis title

Analysis of how to hit the stave with the baton on the cymbal stand

Abstract

(Thesis abstract – to be copied into an appropriate field during an electronic submission – in English.)

Key words

(2-5 keywords, separated by commas)

Spis treści

1	Wstęp	1
2	[Analiza tematu]	3
3	Wymagania i narzędzia	5
4	[Właściwy dla kierunku – np. Specyfikacja zewnętrzna]	7
5	[Właściwy dla kierunku – np. Specyfikacja wewnętrzna]	9
6	Weryfikacja i walidacja	11
7	Podsumowanie i wnioski	13
	Bibliografia	15
	Spis skrótów i symboli	19
	Źródła	21
	Lista dodatkowych plików, uzupełniających tekst pracy	23
	Spis rysunków	25
	Spis tabel	27

Rozdział 1

Wstęp

W XXI wieku mamy do czynienia z coraz to nowszą technologią zarówno w domu, jak i miejscu naszej pracy. Era nowoczesnych technologii dała nam rozwiązania które możemy wykorzystać wszędzie: życiu codziennym, do nauki lub w przemyśle. Szczególnie dużym osiągnięciem dla człowieka jest wymyślenie, opracowanie i skonstruowanie urządzenia, które może go zastąpić przy niektórych czynnościach – robota. Roboty pomagają nam każdego dnia, dzięki nim nasze życie jest prostsze i nie musimy poświęcać czasu na niektóre proste zadania. W dzisiejszym świecie, w prawie każdym domu można zauważyć windę, pralkę czy też zmywarke, a na ulicach działa sygnalizacja świetlna.

W coraz szybciej rozwijającym się świecie potrzeba coraz więcej różnych robotów do coraz to bardziej skomplikowanych zadań, najczęściej by przyspieszyć proces oraz odciążyć ludzi od ciężkiej pracy. Tu przykładem mogą być manipulatory – dzięki nim nie trzeba podnosić ciężkich rzeczy, wystarczy łatwo pokierować robotem, a ten wykona za nas całą ciężką pracę. Jeśli używa się robota zgodnie z zaleceniami producenta, to jesteśmy w stanie zastąpić na dobre człowieka, który musiał by się bardziej męczyć, narażać swoje zdrowie lub też i życie.

Wraz z przyrostem ilości zrobotyzowanych czynności, bez ciągłego udziału człowieka mamy do czynienia z automatyzacją procesu. Czynności powtarzalne, ruchowe, nie wymagające udziału świadomości mogą być wykonywane bez przerwy, w dzień i w nocy. Często mogą też pracować przy zerowym oświetleniu, a nawet w temperaturach, które byłyby niekomfortowe dla człowieka.

Jednocześnie gdy rośnie ilość zautomatyzowanych procesów, wzrasta zapotrzebowanie na ludzi, którzy będą w stanie zaprogramować urządzenia oraz uporządkować systemy które muszą ze sobą współpracować. W szczególności tam, gdzie wiele czynności musi wystąpić w sposób synchroniczny.

Tu z pomocą przychodzą programowalne sterowniki logiczne (PLC). PLC-ki to tak właściwie komputery, które są przystosowane do pracy w przemyśle (ciężkich warunkach?). Wykorzystuje się je praktycznie wszędzie w przemyśle. Są one odpowiedzialne za sterowanie robotami, przesyłanie informacji, obliczanie wartości sygnałów, sygnalizację oraz stały

monitoring urządzeń. Modułowe sterowniki PLC składają się z następujących elementów:

- jednostki centralnej (CPU),
- wejść cyfrowych i analogowych,
- wyjść cyfrowych i analogowych,
- modułu zasilania,
- modułu komunikacji,
- pamięci

Sterowniki są w stanie wysyłać i odbierać sygnały z bardzo dużą częstotliwością. W zależności od tego, jak skomplikowany jest wgrywany program, jeden cykl sterownika może wynosić od kilku do kilkuset milisekund, a w najgorszym przypadku opóźnienie wnoszone przez sterownik może osiągnąć do 2 cykli sterownika.

Jak widać jest to dobry i stosunkowo bezlatencyjny (bez opóźnienia) system do programowania oraz przesyłania danych i informacji do robotów oraz innych urządzeń. To dlatego jest on na „standardowym wyposażeniu” każdej fabryki, gdzie automatyzuje się procesy. Jest to pewien proces optymalizacji. Nie trzeba wykorzystywać tyłu ludzkich rąk do pracy (mniejsze wydatki na pracowników). Sterownik po wgraniu programu może działać non-stop, przez dwadzieścia cztery godziny na dobę, siedem dni w tygodniu, jeśli tylko ma stały dostęp do zasilania. Dlatego właśnie stwierdziłem, że jest to dobry pomysł do wykorzystania w mojej pracy inżynierskiej.

Podsumowując, sterowniki są popularne, występują w naszym życiu codziennym (nawet gdy ich nie są dla nas widoczne i nie jesteśmy tego świadomi). Są to niezwykle ważne narzędzia w szeroko pojętej automatyce przemysłowej i nie tylko. Ułatwiają one procesy, a zarazem są w stanie zapewnić bezpieczeństwo na wysokim poziomie. Obecnie, w czasach gdy automatyzacja oraz robotyzacja odgrywają coraz większą rolę w naszym świecie są uznawane za przydatne, a nierzadko również i niezbędne. Ich zróżnicowanie i szeroka gama wyboru specyfikacji pozwala na trafne dobranie konkretnego modelu do odpowiedniego zadania, przez co znacznie może to ułatwić wykonanie danego projektu.

Rozdział 2

[Analiza tematu]

- sformułowanie problemu
- osadzenie tematu w kontekście aktualnego stanu wiedzy (*state of the art*) o poruszonym problemie
- studia literaturowe [3, 4, 2, 1] - opis znanych rozwiązań (także opisanych naukowo, jeżeli problem jest poruszany w publikacjach naukowych), algorytmów,

Wzory

$$y = \frac{\partial x}{\partial t} \tag{2.1}$$

jak i pojedyncze symbole x i y składa się w trybie matematycznym.

Rozdział 3

Wymagania i narzędzia

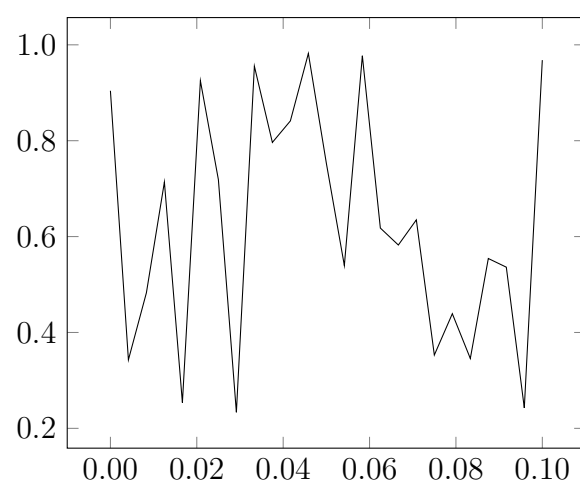
- wymagania funkcjonalne i нефункционалне
- przypadki użycia (diagramy UML) – dla prac, w których mają zastosowanie
- opis narzędzi, metod eksperymentalnych, metod modelowania itp.
- metodyka pracy nad projektowaniem i implementacją – dla prac, w których ma to zastosowanie

Rozdział 4

[Właściwy dla kierunku – np. Specyfikacja zewnętrzna]

Jeśli „Specyfikacja zewnętrzna”:

- wymagania sprzętowe i programowe
- sposób instalacji
- sposób aktywacji
- kategorie użytkowników
- sposób obsługi
- administracja systemem
- kwestie bezpieczeństwa
- przykład działania
- scenariusze korzystania z systemu (ilustrowane zrzutami z ekranu lub generowanymi dokumentami)



Rysunek 4.1: Podpis rysunku po rysunkiem.

Rozdział 5

[Właściwy dla kierunku – np. Specyfikacja wewnętrzna]

Jeśli „Specyfikacja wewnętrzna”:

- przedstawienie idei
- architektura systemu
- opis struktur danych (i organizacji baz danych)
- komponenty, moduły, biblioteki, przegląd ważniejszych klas (jeśli występują)
- przegląd ważniejszych algorytmów (jeśli występują)
- szczegóły implementacji wybranych fragmentów, zastosowane wzorce projektowe
- diagramy UML

Krótką wstawka kodu w linii tekstu jest możliwa, np. **int a;** (biblioteka `listings`). Dłuższe fragmenty lepiej jest umieszczać jako rysunek, np. kod na rys 5.1, a naprawdę długie fragmenty – w załączniku.

```
1 class test : public basic
2 {
3     public:
4         test (int a);
5         friend std::ostream operator<<(std::ostream & s,
6                                         const test & t);
7     protected:
8         int _a;
9
10 };
```

Rysunek 5.1: Pseudokod w `listings`.

Rozdział 6

Weryfikacja i walidacja

- sposób testowania w ramach pracy (np. odniesienie do modelu V)
- organizacja eksperymentów
- przypadki testowe zakres testowania (pełny/niepełny)
- wykryte i usunięte błędy
- opcjonalnie wyniki badań eksperymentalnych

Tabela 6.1: Nagłówek tabeli jest nad tabelą.

ζ	metoda						
	alg. 1	alg. 2	alg. 3			alg. 4, $\gamma = 2$	
			$\alpha = 1.5$	$\alpha = 2$	$\alpha = 3$	$\beta = 0.1$	$\beta = -0.1$
0	8.3250	1.45305	7.5791	14.8517	20.0028	1.16396	1.1365
5	0.6111	2.27126	6.9952	13.8560	18.6064	1.18659	1.1630
10	11.6126	2.69218	6.2520	12.5202	16.8278	1.23180	1.2045
15	0.5665	2.95046	5.7753	11.4588	15.4837	1.25131	1.2614
20	15.8728	3.07225	5.3071	10.3935	13.8738	1.25307	1.2217
25	0.9791	3.19034	5.4575	9.9533	13.0721	1.27104	1.2640
30	2.0228	3.27474	5.7461	9.7164	12.2637	1.33404	1.3209
35	13.4210	3.36086	6.6735	10.0442	12.0270	1.35385	1.3059
40	13.2226	3.36420	7.7248	10.4495	12.0379	1.34919	1.2768
45	12.8445	3.47436	8.5539	10.8552	12.2773	1.42303	1.4362
50	12.9245	3.58228	9.2702	11.2183	12.3990	1.40922	1.3724

Rozdział 7

Podsumowanie i wnioski

- uzyskane wyniki w świetle postawionych celów i zdefiniowanych wyżej wymagań
- kierunki ewentualnych danych prac (rozbudowa funkcjonalna ...)
- problemy napotkane w trakcie pracy

Bibliografia

- [1] Imię Nazwisko i Imię Nazwisko. *Tytuł strony internetowej*. 2021. URL: <http://gdzies/w/internecie/internet.html> (term. wiz. 30.09.2021).
- [2] Imię Nazwisko, Imię Nazwisko i Imię Nazwisko. „Tytuł artykułu konferencyjnego”. W: *Nazwa konferencji*. 2006, s. 5346–5349.
- [3] Imię Nazwisko, Imię Nazwisko i Imię Nazwisko. „Tytuł artykułu w czasopiśmie”. W: *Tytuł czasopisma* 157.8 (2016), s. 1092–1113.
- [4] Imię Nazwisko, Imię Nazwisko i Imię Nazwisko. *Tytuł książki*. Warszawa: Wydawnictwo, 2017. ISBN: 83-204-3229-9-434.

Dodatki

Spis skrótów i symboli

DNA kwas deoksyrybonukleinowy (ang. *deoxyribonucleic acid*)

MVC model – widok – kontroler (ang. *model-view-controller*)

N liczebność zbioru danych

μ stopnień przyleżności do zbioru

\mathbb{E} zbiór krawędzi grafu

\mathcal{L} transformata Laplace’a

Źródła

Jeżeli w pracy konieczne jest umieszczenie długich fragmentów kodu źródłowego, należy je przenieść w to miejsce.

```
1 if (_nClusters < 1)
2     throw std::string ("unknown_number_of_clusters");
3 if (_nIterations < 1 and _epsilon < 0)
4     throw std::string ("You should set a maximal number of
        iteration or minimal difference — epsilon.");
5 if (_nIterations > 0 and _epsilon > 0)
6     throw std::string ("Both number of iterations and minimal
        epsilon set — you should set either number of iterations
        or minimal epsilon.");
```

Lista dodatkowych plików, uzupełniających tekst pracy

W systemie do pracy dołączono dodatkowe pliki zawierające:

- źródła programu,
- dane testowe,
- film pokazujący działanie opracowanego oprogramowania lub zaprojektowanego i wykonanego urządzenia,
- itp.

Spis rysunków

4.1	Podpis rysunku po rysunkiem.	8
5.1	Pseudokod w <code>listings</code>	10

Spis tabel

6.1	Nagłówek tabeli jest nad tabelą.	12
-----	--	----