

Igor Popławski

Aproksymacja funkcji

Metody numeryczne (MNUB), semestr 24L

Prowadzący: Paweł Mazurek

Politechnika Warszawska, Wydział Elektroniki i Technik
Informacyjnych

Warszawa 11.05.2024r.

Oświadczam, że niniejsza praca stanowiąca podstawę do uznania osiągnięcia efektów uczenia się z przedmiotu Metody numeryczne została wykonana przeze mnie samodzielnie.

Spis treści

1.Lista symboli matematycznych i akronimów	3
2.Wprowadzenie	4
Metoda różnicy wstecznej	4
Aproksymacja średniokwadratowa funkcji na podstawie jej dyskretnych wartości.....	5
Wyznaczanie wielomianów Legendre’a.....	5
3.Metodyka i wyniki doświadczeń.....	6
Zad.1	6
Zad.2	7
Zad.3	8
Zad.4	9
4.Dyskusja wyników eksperymentów numerycznych	12
Zad.1	12
Zad.2	12
Zad.3	12
Zad.4	13
5.Wnioski ogólne	14
6.Lista źródeł.....	14
7.Listing programów	15

1. Lista symboli matematycznych i akronimów

x, y, z – zmienne skalarne

$\mathbf{x}, \mathbf{y}, \mathbf{z}$ – wektory zmiennych skalarnych

$\Phi, \mathbf{X}, \mathbf{Y}$ – macierze zmiennych skłarnych

2.Wprowadzenie

Zadanie projektowe polega na aproksymacji współrzędnych oraz prędkości środka ciężkości osoby monitorowanej przez czujniki nadające krótkie impulsy elektromagnetyczne małej mocy, które umożliwiają estymację odległości między osobą monitorowaną a czujnikiem. Do wyznaczania odległości między czujnikami a osobą użyte zostały dwa czujniki. Do wyznaczania współrzędnych dla każdej chwili $n=1,\dots,N$ należy rozwiązać układ nieliniowych równań algebraicznych:

$$x_n = \frac{1}{8}(d_{1,n}^2 - d_{2,n}^2 + 16) \quad (1)$$

$$y_n = \sqrt{d_{1,n}^2 - x_n^2} \quad (2)$$

Do projektu wykorzystujemy dane otrzymane z serii pomiarów umieszczone w pliku MNUB_24L_P2_dane32.mat, znajduje się tam:

- tablica komórkowa **t** o 20 elementach zawierająca wektory o różnej długości, których wartości reprezentują momenty pomiaru odległości między osobą a czujnikami;
- tablica komórkowa **D** o 20 elementach zawierająca macierze o dwóch kolumnach i różnej liczbie wierszy, której wartości reprezentują zmierzone odległości między osobą a dwoma czujnikami; kolumny odpowiadają kolejnym czujnikom radarowym, a wiersze – kolejnym chwilom;
- tablica komórkowa **xref** i **yref** o 20 elementach zawierająca wektory o różnej długości, których wartości reprezentują referencyjne wartości współrzędnych x i y osoby w kolejnych chwilach ruchu;
- zmienna **vref**, której wartość reprezentuje referencyjną wartość prędkości ruchu osoby.

Metody użyte w zadaniu projektowym:

Metoda różnicy wstecznej

Z definicji pochodnej:

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h} \quad (3)$$

Możemy przybliżyć pochodną za pomocą wzoru:

$$D_B(f, x, h) = \frac{f(x) - f(x-h)}{h} \quad (4)$$

Gdzie:

$f(x)$ – funkcja, której pochodnej szukamy

x - argument funkcji

h - krok z jakim następuje zmiana argumentu funkcji

D_B - przybliżona wartość pochodnej

Aproksymacja średniokwadratowa funkcji na podstawie jej dyskretnych wartości

By wyznaczyć wektor parametrów \mathbf{p} dla którego liniowa kombinacja liniowo niezależnych funkcji dana równaniem:

$$\hat{f}(x; \mathbf{p}) = \sum_{k=1}^K p_k \cdot \phi_k(x) \quad (5)$$

Gdzie:

\hat{f} - przybliżenie funkcji $f(x)$

p_k - k -ta wartość wektora \mathbf{p}

$\phi_k(x)$ – funkcja liniowo niezależna

jest najlepszym przybliżeniem ciągu wartości nieznanej funkcji $f(x)$ korzystamy ze wzoru:

$$\Phi^T \cdot \Phi \cdot \mathbf{p} = \Phi^T \cdot \mathbf{y} \quad (6)$$

Gdzie:

Φ - macierz o wymiarze N na K (gdzie N to liczba dyskretnych punktów x_N a K to liczba niezależnych funkcji ϕ_k) zawierająca w kolumnach wartości funkcji ϕ_k dla kolejnych wartości x_N

\mathbf{y} – wektor o rozmiarze N zawierający wartości funkcji $f(x_N)$

\mathbf{p} – szukany wektor

Przekształcając równanie (2) otrzymujemy równanie postaci:

$$\hat{\mathbf{p}} = (\Phi^T \cdot \Phi)^{-1} \cdot \Phi^T \cdot \mathbf{y} \quad (7)$$

Które możemy rozwiązać np. za pomocą programu MATLAB.

Otrzymujemy w ten sposób wektor współczynników wagowych decydujących o tym jaki udział w naszej funkcji $f(x)$ mają niezależne liniowo funkcje $\phi_k(x)$. Możemy więc na tej podstawie przybliżyć naszą badaną funkcję.

Wyznaczanie wielomianów Legendre'a

W projekcie jako bazę funkcji liniowo niezależnych w zadaniu 2 i 3 przyjmujemy wielomiany Legendre'a opisane wzorami:

$$P_0(t) = 1 \quad (8)$$

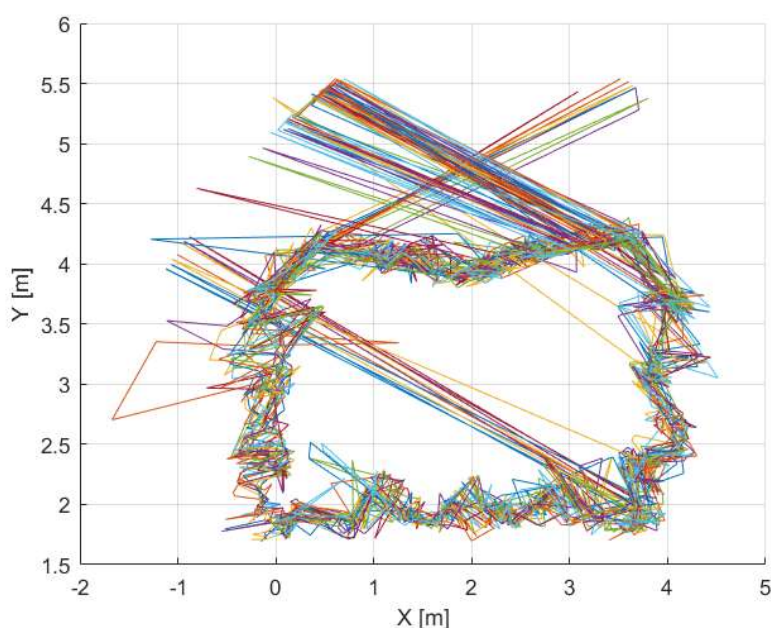
$$P_1(t) = t \quad (9)$$

$$P_k(t) = \frac{2k-1}{k} \cdot t \cdot P_{k-1}(t) - \frac{k-1}{k} \cdot P_{k-2}(t) \text{ dla } k = 2, \dots, K-1 \quad (10)$$

3. Metodyka i wyniki doświadczeń

Zad.1

By wyznaczyć trajektorie ruchu osoby korzystamy z równania (1) obliczając współrzędną x oraz z równania (2) obliczając współrzędną y dla każdej chwili w czasie w której dokonano pomiaru i zapisujemy te wartości do wektorów \mathbf{x} i \mathbf{y} . Powtarzamy więc to po całej długości wektora \mathbf{t} rysujemy dwuwymiarowy wykres x od y – używając środowiska MATLAB za pomocą funkcji `plot`. Czynność tą powtarzamy 20 razy, jako że mamy 20 serii pomiarowych i wpisujemy wektory do tablicy komórkowej. Otrzymane trajektorie przedstawiono na rys. 1:



Rys.1 Dwadzieścia dwuwymiarowych trajektorii ruchu osoby

By wyznaczyć prędkość ruchu używamy metody różnicy wstecznej (patrz wprowadzenie). W tym celu dla każdej chwili w jakiej dokonano pomiaru wyznaczamy prędkość obliczając pierwiastek z sumy kwadratów różnic współrzędnych x i y między obecną, a poprzednią chwilą, w której dokonano pomiaru, a następnie dzielimy ten pierwiastek na różnice chwil w czasie. W ten sposób otrzymujemy wektor wartości prędkości \mathbf{v} o długości o jeden mniejszej niż długość wektora \mathbf{t} . Powtarzamy ten proces dla każdej z 20 serii pomiarów i wpisujemy wektory do tablicy komórkowej.

Mając już tablice komórkowe zawierające wektory \mathbf{x} , \mathbf{y} i \mathbf{v} dla każdej z 20 serii pomiarowych wyznaczamy wartości bezwzględne różnicy wektorów \mathbf{x}_{ref} i \mathbf{x} oraz \mathbf{y}_{ref} i \mathbf{y} . Następnie sumujemy wszystkie wartości wektorów różnicy ze wszystkich serii i przypisujemy do zmiennych x_{sum} i y_{sum} . By otrzymać średni błąd bezwzględny wyznaczenia współrzędnej x i y dzielimy x_{sum} i y_{sum} na łączną długość wektorów \mathbf{x} i \mathbf{y} ze wszystkich 20 serii. Ponad to musimy obliczyć średni błąd bezwzględny wyznaczenia prędkości. By to zrobić wyznaczamy prędkość średnią – mnożymy wartości średnie poszczególnych wektorów \mathbf{v} przez ich długość, następnie sumujemy otrzymane iloczyny i dzielimy tą sumę na łączną długość wszystkich wektorów \mathbf{v} . Mając już prędkość średnią odejmujemy ją od \mathbf{v}_{ref} i bierzemy moduł, jest to nasz średni błąd bezwzględny wyznaczenia prędkości.

Wartości średniego błędu bezwzględnego wyznaczenia współrzędnej x , współrzędnej y oraz prędkości ruchu przedstawiono w tabeli:

Tabela 1. Wartości średniego błęd bezwzględny wyznaczenia współrzędnej x, współrzędnej y oraz prędkości z jednostkami

średni błąd bezwzględny wyznaczenia:	prędkości	Współrzędnej x	Współrzędnej y
wartość	2.86523	0.22942	0.14955
jednostki	m/s	m	m

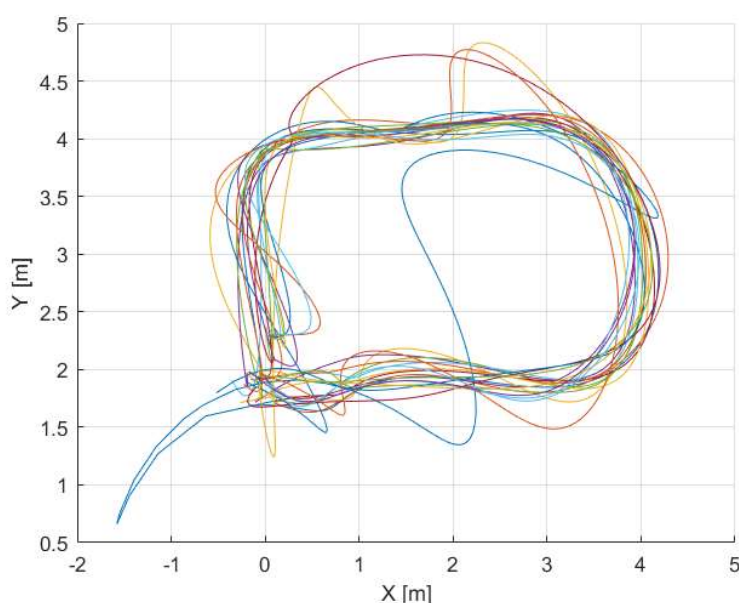
Zad.2

By dokonać aproksymacji funkcji ruchu badanej osoby na podstawie $M=15$ równomiernie rozłożonych punktów dla każdej serii pomiarów tworzymy wektor rozłożonych równomiernie punktów w czasie od 0 do ostatniej wartości wektora \mathbf{t} (w programie wektor nazwano **tperfM**). W kolejnym kroku znajdujemy wartości zapisane w wektorze \mathbf{t} które są najbliższe wartościom idealnie rozłożonym. W programie została użyta do tego funkcja, która dla każdej kolejnej wartości **tperfM** liczy różnice z wartościami wektora \mathbf{t} i wpisuje do wektora **tM** wartość wektora \mathbf{t} jeśli jest ona mniejsza niż poprzednia różnica. W takim przypadku wpisujemy również wartości macierzy \mathbf{D} z obu kolumn do macierzy \mathbf{Y} . Po dokonaniu skalujemy wektor **tM** dla każdej serii pomiarów tak by miał zakres $[-1,1]$. Skalujemy również wektor \mathbf{t} w ten sam sposób.

Przechodząc dalej tworzymy macierz Φ przyjmując za bazę funkcji liniowo niezależnych wielomiany Legendre'a. Dla każdej z 15 wartości wektora **tM** i dla każdego K (gdzie K należy od 1 do 10) używamy funkcji, która zwraca nam 0 dla $K=0$, wartość z wektora **tM** gdy $K=1$ oraz odpowiednią wartość dla K od 2 do 10 (patrz wyznaczanie wielomianów Legendre'a). Mając już macierz Φ używamy równania (7) by wyznaczyć dwa wektory $\hat{\mathbf{p}}$ dla wartości x i y (jako wektor \mathbf{y} przyjmujemy odpowiednie kolumny macierzy \mathbf{Y}).

Po wyznaczeniu tych wektorów powtarzamy schemat z Zad.2 z tym że w pętli idącej po całej długości wektora \mathbf{t} wyznaczamy dodatkowo dwie estymaty (będące wektorami o długości wektora czasu) stosując wzór (5) dla współrzędnych x i y . Zmieniamy również wartości macierzy \mathbf{D} przy wyliczaniu wektorów \mathbf{x} i \mathbf{y} oraz stosowaniu wzorów (1) i (2) na odpowiednie wartości z wektorów estymat. Dalej proces jest identyczny jak w Zad.2.

Wykresy 20 trajektorii osoby zostały przedstawione na rys.2.



Rys.2 Dwadzieścia zaproksymowanych dwuwymiarowych trajektorii ruchu osoby

Wyniki średniego błędu bezwzględnego wyznaczenia współrzędnej x , współrzędnej y oraz prędkości ruchu z Zad.3 przedstawiono w następującej tabeli:

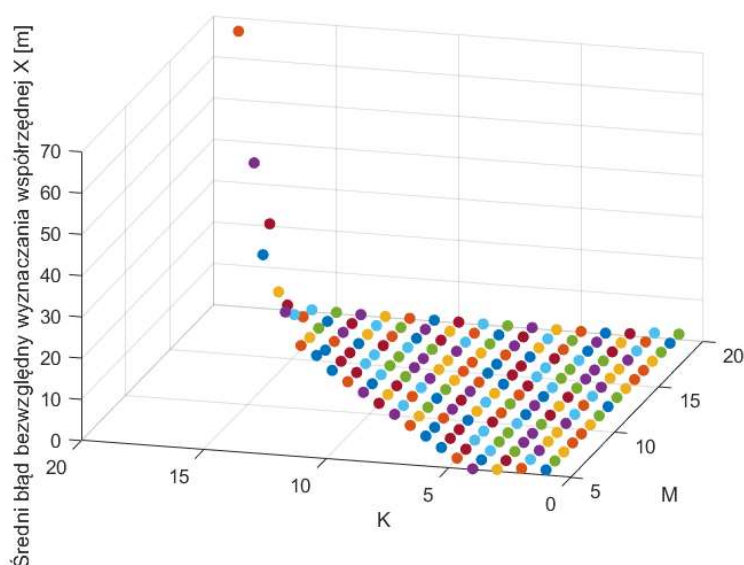
Tabela 2. Wartości średniego błęd bezwzględny wyznaczenia współrzędnej x , współrzędnej y oraz prędkości z jednostkami

średni błąd bezwzględny wyznaczenia:	prędkości	Współrzędnej x	Współrzędnej y
wartość	0.03991	0.19955	0.15585
jednostki	m/s	m	m

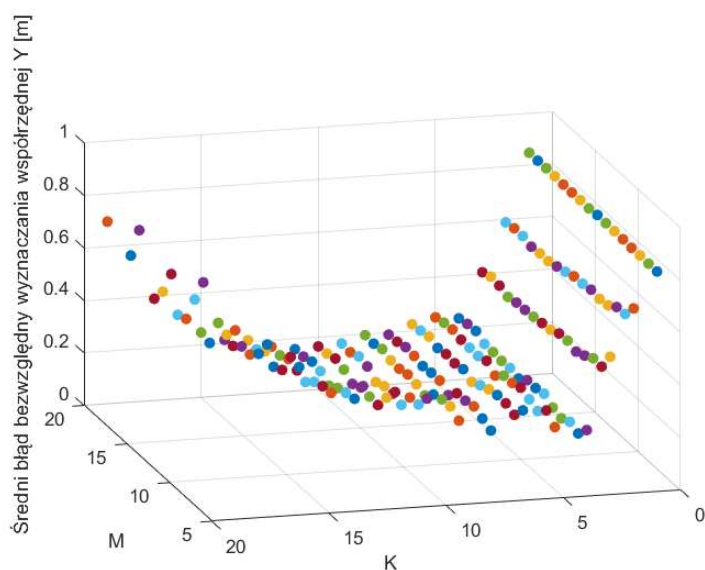
Zad.3

W celu przeprowadzenia badania zależności wartości średniego błędu bezwzględnego wyznaczenia współrzędnej x , współrzędnej y oraz prędkości ruchu od parametrów M i K powtarzamy schemat z Zad.2 dla każdej wartości M w zakresie 5-20 i każdej wartości $K < M$. W programie MATLAB zrobimy to poprzez umieszczenie naszego algorytmu w dwóch pętlach „for”, jednej inkrementującą wartość M od 5 do 20 następną pętlę zawartą w poprzedniej inkrementującą wartość K od 1 do $M-1$. W każdym obiegu drugiej pętli zapisujemy wartości naszych średnich błędów bezwzględnych wyznaczania wartości współrzędnej x , y i prędkości. W odpowiednich macierzach (**KMXerr**, **KMYerr**, **KMVerr**) o wymiarach 20 na 19. Po zastosowaniu algorytmu do wszystkich kombinacji M i K tworzymy trójwymiarowe wykresy zależności średniego błędu bezwzględnego od M i K (w programie MATLAB realizujemy to za pomocą dwóch pętli oraz funkcji „scatter3”).

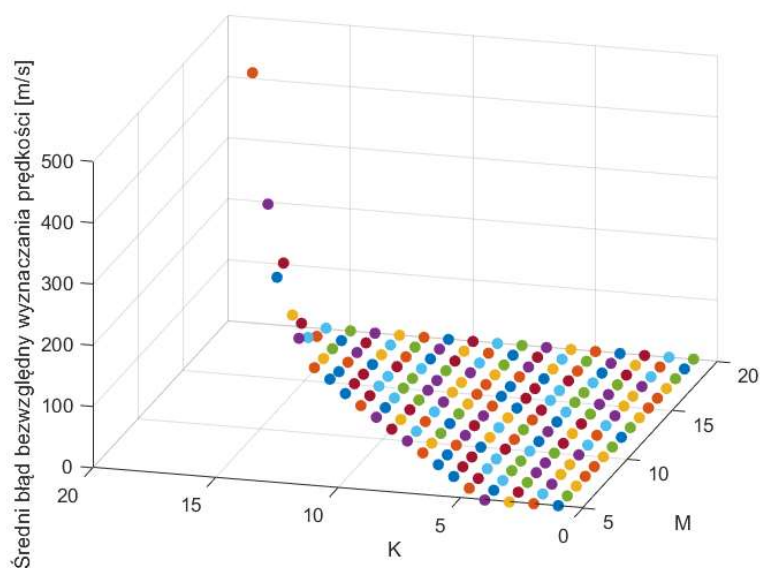
Wykresy zależności zostały przedstawione na rys.3, rys.4 oraz rys.5.



Rys.3 Wykres zależności wartości średniego błędu bezwzględnego wyznaczenia współrzędnej x od parametrów M i K



Rys.4 Wykres zależności wartości średniego błędu bezwzględnego wyznaczenia współrzędnej y od parametrów M i K



Rys.5 Wykres zależności wartości średniego błędu bezwzględnego wyznaczenia prędkości ruchu od parametrów M i K

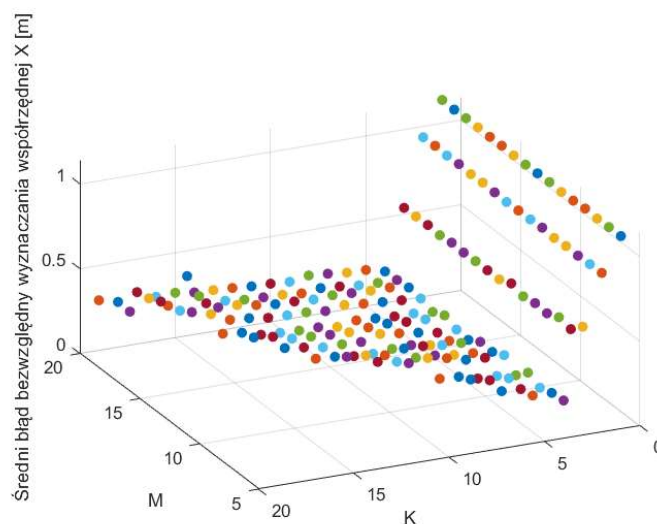
Zad.4

W Zad.4 przeprowadzamy te same operacje co w Zad.3 z tym, że oprócz rozkładania wektora czasu na M punktów rozkładamy też wektor \mathbf{t} na K punktów postępujemy przy tym tak samo jak przy wyznaczaniu wektora \mathbf{tM} – znajdujemy najpierw idealnie równomiernie rozłożone punkty po całym przedziale czasu, a następnie znajdujemy najbliższe im wartości zawarte w wektorze \mathbf{t} i wpisujemy te wartości do wektora \mathbf{tK} . Co więcej tworząc macierz Φ przyjmujemy funkcje:

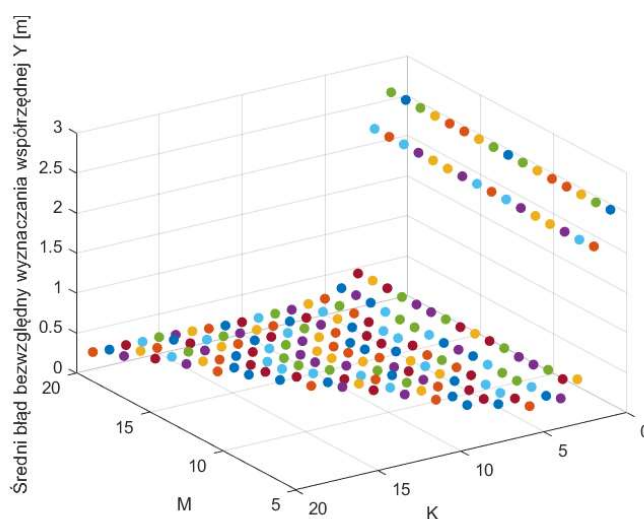
$$G_k(t) = \frac{1}{3 \cdot \sqrt{2 \cdot \pi}} \cdot \exp \left\{ -\frac{1}{18} \cdot (t - t'_k)^2 \right\}$$

jako bazę funkcji liniowo niezależnych. Pod t i t_k podstawiamy odpowiednie wartości z wektorów \mathbf{tM} oraz \mathbf{tK} . Obliczając estymaty także podstawiamy funkcje G_k do wzoru (5).

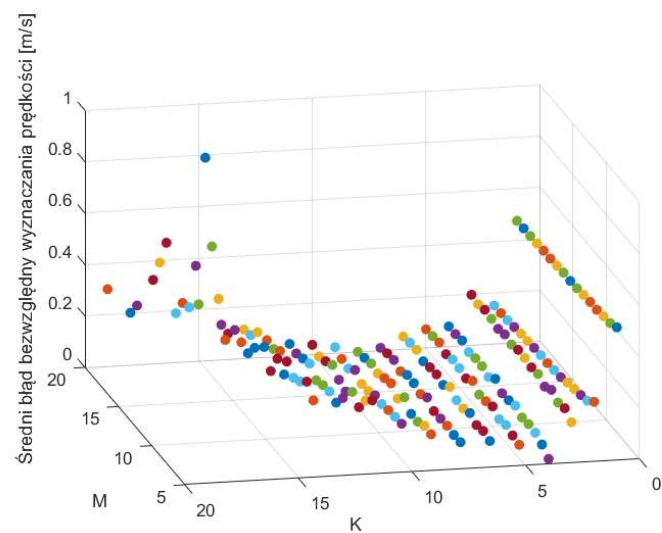
Wykresy zależności wartości średniego błędu bezwzględnego wyznaczenia współrzędnej x , współrzędnej y oraz prędkości ruchu od parametrów M i K zostały przedstawione na rys.6 rys.7 i rys.8.



Rys.6 Wykres zależności wartości średniego błędu bezwzględnego wyznaczenia współrzędnej x od parametrów M i K



Rys.7 Wykres zależności wartości średniego błędu bezwzględnego wyznaczenia współrzędnej y od parametrów M i K



Rys.8 Wykres zależności wartości średniego błędu bezwzględnego wyznaczenia prędkości ruchu od parametrów M i K

4. Dyskusja wyników eksperymentów numerycznych

Zad.1

Już na pierwszy rzut oka widać, że pomiary, gdy badany znajdował się w okolicach współrzędnych (0,2), (0,4) i (4,2) znacząco odbiegają od reszty zniekształcając obraz.

Wyznaczanie trajektorii ruchu bez aproksymacji otrzymanych funkcji może zawierać wiele błędów i niedokładności co może powodować błędne wnioski. W tym przypadku np. analizując nagłą zmianę położenia osoby można by pomyśleć, że przewróciła się.

Wyniki średniego błędu bezwzględnego wyznaczenia współrzędnej x oraz współrzędnej y to 23cm i 15cm, a jako że osoba poruszała się po prostokącie 2m na 4m to całkiem znaczący błąd. Średni błąd bezwzględny wyznaczenia prędkości to 2,87 m/s co jest ogromny błędem biorąc pod uwagę to, że przeciętny człowiek chodzi z prędkością około 1,25 m/s. Możemy z tego wywnioskować że bez aproksymacji szacowanie prędkości nie ma większego sensu, bo jest obarczone błędem przekraczającym 200%. Tak duży błąd wynika z dużych przeskoków między punktami widocznymi na rogach wykresu, analizując kolejne punkty gdy algorytm trafi na tak mocno zakłócony pomiar oblicza prędkości między pomiarem w miarę poprawnym a zakłóconym i podaje o wiele za duże wartości prędkości, kiedy w rzeczywistości badany w żadnej chwili nie poruszał się tak szybko.

Zad.2

Po przeprowadzeniu aproksymacji na podstawie 10 równomiernie rozłożonych punktów przyjmując jako bazę funkcji liniowo niezależnych wielomiany Legendre'a dla $K=10$ trajektorie widoczne na rys.2 są dużo bardziej jednolite i wygładzone, mają tylko nieliczne znaczące odkształcenia (np. pik nakreślony niebieską linią). To przybliżenie dużo lepiej oddaje tor ruchu, po którym w rzeczywistości którym się badany (był to prostokąt).

Wykresy są lekko zaokrąglone na rogach, ponieważ żaden z 10 punktów nie trafił w chwilę, gdy dana osoba stała na „rogu” prostokąta po którym się poruszała. Chociaż większość z 20 przebiegów nie ma znaczących zakłóceń to np. przebieg zaznaczony linią niebieską ma wyraźny pik i zniekształcenia co wynika z faktu, że równomiernie rozłożone punkty trafiły akurat w wartości o największych zakłóceniach. Widzimy więc, że chodź ta metoda aproksymacji w większości przypadków jest skuteczna to w sytuacji, gdy źle rozłożymy nasze punkty służące do aproksymacji, możemy otrzymać zupełnie błędny kształt wykresu.

Wartości średniego błędu bezwzględnego wyznaczania współrzędnej x poprawiły się o około 3cm za to średniego błędu bezwzględnego wyznaczania współrzędnej y bardzo nieznacznie się zmieniły (nawet nie 1cm) z czego możemy wywnioskować, że aproksymacja oparta na metodzie najmniejszych kwadratów pozwala z trochę większą dokładnością odwzorować ciąg pozycji w jakich znajdował się pacjent. Jeśli chodzi o wyznaczanie prędkości teraz błąd to jedynie około 0.04 m/s, czyli 3,2% przeciętnej prędkości człowieka. Widzimy więc, że po zastosowaniu aproksymacji jesteśmy w stanie bardzo dobrze przybliżyć prędkość z jaką poruszał się badany. Wynika to z faktu, że analizując naszą zaproksymowaną funkcję nie natrafiamy na wartości odstające więc prędkość liczona między punktami nie skacze nagle do dużych, błędnych wartości.

Zad.3

Z wykresów widzimy, że przy wyznaczaniu współrzędnej x błędy utrzymują się na dość niskim poziomie do momentu, gdy K nie przekroczy 15 wtedy błędy znacząco się nasilają, ta sama sytuacja występuje przy wyznaczaniu prędkości jednak po wartościach $K>15$ średnie błędy bezwzględne są jeszcze większe (rzęd setek).

Zależność średnich błędów bezwzględnych wyznaczania wartości współrzędnej y ma jednak inny kształt. Najmniejsze błędy pojawiają się, gdy M jest w zakresie 5-10. Gdy wartość M rośnie (powyżej 10) lub maleje (poniżej 5) to błąd stale rośnie, jednak nie dochodzi on do rzędu dziesiątek lub setek przy żadnej z danych wartości M i K .

Z zależności średniego błędu bezwzględnego wyznaczania współrzędnej x oraz wyznaczania prędkości od M i K widzimy, że nasza aproksymacja oparta na metodzie najmniejszych kwadratów z wykorzystaniem wielomianów Legendre'a nie sprawdza się dla dużych wartości K i M . Dla niektórych przypadków nasze przybliżenie z powodu dużej liczby M może brać pod uwagę więcej punktów odstających. Dla wszystkich przypadków natomiast wysoka wartość K powoduje, że zaczyna występować efekt Rungego. Jest to efekt, który zachodzi przy próbie przybliżania funkcji pierwotnej za pomocą zbyt dużej liczby wielomianów. W efekcie zastosowania zbyt dużej liczby wielomianów nasza estymata niepotrzebnie „oscyluje” po drodze do węzłów, przez co posiada ona w sobie jeszcze więcej punktów odstających od rzeczywistych wartości niż w pierwotnym zbiorze pomiarów.

Analizując wykres zależności średniego błędu bezwzględnego współrzędnej y od M i K możemy również wywnioskować że dla mniejszych wartości K gorzej przybliżamy rzeczywiste współrzędne y co wynika z faktu, że używamy mniejszej liczby funkcji liniowo niezależnych, więc nie możemy dokładnie oddać kształtu zależności y od t .

Zad.4

Przybliżając nasze funkcje używając funkcji G_k jako bazy funkcji liniowo niezależnych średnie błędy bezwzględne wyznaczania współrzędnej x i prędkości maleją dla dużych K i M . Błąd wyznaczania współrzędnej x spadł drastycznie dla dużych wartości M i K . Teraz maksymalna jego wartość to około jeden, obecnie największe błędy obserwujemy dla $K=1$, $K=2$ i $K=3$. Niewielkie falowanie wykresu widać też dla maksymalnych wartości K . Błędy wyznaczania współrzędnej y utrzymują się na mniej więcej tym samym poziomie co poprzednio (około 0,3), jedynie dla $K=1$ i $K=2$ obserwujemy zwiększenie się błędu prawie do wartości 3. Średni błąd wyznaczania prędkości przy dużym M i K także spadł bardzo drastycznie – z rzędu setek na wartość poniżej 1. Dla błędów prędkości najniższe błędy pojawiają się dla K w zakresie od 5 do 10 i rosną im bardziej wykraczają za ten zakres.

Fakt zmniejszenia się błędów dla K i M powyżej 15 pokazuje nam, że użycie aproksymacji za pomocą metody najmniejszych kwadratów, używając Gaussoidy jako bazy funkcji liniowo niezależnych jest dużo bardziej skuteczne przy dużej liczbie węzłów.

Dla tej metody również dostajemy duże błędy przy małych wartościach K (wyznaczając współrzędne x , y oraz prędkość), ponieważ mała liczba Gaussoid również nie jest w stanie dobrze przybliżyć zależności wartości naszych współrzędnych lub prędkości od czasu.

5.Wnioski ogólne

Dzięki aproksymacji funkcji możemy wyeliminować błędy pomiarowe w naszych danych oraz lepiej oddać rzeczywisty przebieg badania jakie przeprowadziliśmy. Eliminując skoki i nagłe zniekształcenia danych możemy także z większą dokładnością wyznaczać inne wartości, które nie były bezpośrednio mierzone, ale są powiązane z wielkościami mierzonymi podczas badania (np. w naszym przypadku prędkość).

Stosując aproksymacje za pomocą metody najmniejszych kwadratów ważny jest jednak odpowiedni dobór ilości węzłów i ich rozmieszczenie – ważne jest by nie znalazły się na pomiarach, które zostały znacząco zakłócone.

Używając tej metody ważny jest również dobór odpowiedniej bazy, na podstawie której będziemy tworzyć funkcje liniowo niezależne, ważna jest również ilość tych funkcji. Używając zbyt dużej ilości niektórych funkcji (np. wielomianów Legendre’a lub sinusoid) możemy wywołać efekt Rungego, a używając zbyt małej ilości nie odwzorujemy wystarczająco poprawnego przebiegu aproksymowanej funkcji.

Z naszego badania wynika, że Gaussoidy dużo lepiej radzą sobie z zakłóceniami związanymi z efektem Rungego niż wielomiany Legendre’a. Stosując obie z tych funkcji najlepiej użyć liczby funkcji liniowo niezależnych w zakresie od 5 do 10.

6.Lista źródeł

[1] Paweł Mazurek – MNUB 24L, wykłady Wydział Elektroniki i Technik Informacyjnych, Politechnika Warszawska

[2] Wykład 1 Interpolacja wielomianowa, Wydział Mechaniczny Energetyki i Lotnictwa, Politechnika Warszawska

7.Listing programów

- plik Zad1_Zad2

```
load MNUB_24L_P2_dane32.mat;

%deklaracje zmiennych, macierzy i
komórek
X=cell(1,20); %komórki wyników X i Y
Y=cell(1,20);
Vav=zeros(1,20); %prędkość średnia
Xerrabs=cell(1,20); %bezw. błędy
względne
Yerrabs=cell(1,20);
Verrabs=cell(1,20);
V=cell(1,20); %prędkość
lengthall=0; %ilosc wszystkich
pomiarów
xsum=0;
ysum=0;
figure;
xlabel("X [m]");
ylabel("Y [m]");
grid on;
hold on;
for i=1:20
    X{i}=zeros(1,length(t{i}));
    Y{i}=zeros(1,length(t{i}));
    V{i}=zeros(1,length(t{i})-1);
    for j=1:length(t{i})
        X{i}(j)=(D{i}(j,1)*D{i}(j,1)-
D{i}(j,2)*D{i}(j,2)+16)/8;

Y{i}(j)=sqrt(D{i}(j,1)*D{i}(j,1)-
X{i}(j)*X{i}(j));
        if j>=2
            V{i}(j-1)=(sqrt((X{i}(j)-
X{i}(j-1))^2+(Y{i}(j)-Y{i}(j-
1))^2))/(t{i}(j)-t{i}(j-1));
%wyznaczanie prędkości
        end
    end
    Xerrabs{i}=abs(xref{i}'-X{i});
%obliczenie błędów względnych
Yerrabs{i}=abs(yref{i}'-Y{i});
Vav(i)=mean(V{i})*length(V{i});
xsum=sum(Xerrabs{i})+xsum;
%sumowanie wszystkich X oraz Y
ysum=sum(Yerrabs{i})+ysum;
lengthall=lengthall+length(V{i});
%sumowanie długości wszystkich
wektorów
plot(X{i},Y{i});
end
vav=sum(Vav)/lengthall;
verr=abs(vav-vref); %obliczanie
średnich błędów bezwzględnych
wyznaczania X i Y i prędkości
xerrall=xsum/(lengthall+20);
```

```
yerrall=ysum/(lengthall+20);
hold off;

%Zad.2
M=15;
K=10;
tM=cell(1,20);
tscale=cell(1,20);
tperfM = cell(1,20);
Fi=cell(1,20);
y=cell(1,20);
p1=cell(1,20);
p2=cell(1,20);

for i=1:20
    tperfM{i} = linspace(0,
t{i}(length(t{i})), M); %idealny
rozklad wektora t na M punktów
    tM{i}=zeros(1,M);
    y{i}=zeros(M,2); %zmienna
odpowiedzialna za przechowanie
odległości od czujników w wybranych
punktach czasu
    tscale{i}=zeros(1,length(t{i}));
    for j=1:M %funkcja do znajdowania
najbliższych wartości idealnego
rozkładu
        difftmax=0;
        for m=1:length(t{i})
            difft=abs(tperfM{i}(j)-
t{i}(m));
            if m==1
                difftmax=difft;
                tM{i}(j)=t{i}(m);
                y{i}(j,1)=D{i}(m,1);
                y{i}(j,2)=D{i}(m,2);
            end
            if difft<difftmax
                difftmax=difft;
                tM{i}(j)=t{i}(m);
                y{i}(j,1)=D{i}(m,1);
                y{i}(j,2)=D{i}(m,2);
            end
        end
        end
        tM{i}=tM{i}/tM{i}(M)*2-1;
%przeskalowanie wektroów tak by
wartości zawierały się od -1 do 1

tscale{i}=t{i}/t{i}(length(t{i}))*2-1;
end

for i=1:20 %obliczenie macierzy Fi
    Fi{i}=zeros(M,K);
    for j=1:M
```

```

        for k=0:K-1

Fi{i}(j,k+1)=PolyLeg(tM{i}(j),k);
        end
    end

p1{i}=(Fi{i}'*Fi{i})\Fi{i}'*y{i}(:,1);

p2{i}=(Fi{i}'*Fi{i})\Fi{i}'*y{i}(:,2);
end

est1=cell(1,20);
est2=cell(1,20);
X2=cell(1,20);
Y2=cell(1,20);
V2av=zeros(1,20);
X2errabs=cell(1,20);
Y2errabs=cell(1,20);
V2errabs=cell(1,20);
V2=cell(1,20);
lengthall2=0;
x2sum=0;
y2sum=0;
figure
hold on;
xlabel("X [m]");
ylabel("Y [m]");
grid on;
for i=1:20
    X2{i}=zeros(1,length(t{i}));
    Y2{i}=zeros(1,length(t{i}));
    V2{i}=zeros(1,length(t{i})-1);
    est1{i}=zeros(1,length(t{i}));
    est2{i}=zeros(1,length(t{i}));
    for j=1:length(t{i})
        for k=1:K

est1{i}(j)=est1{i}(j)+p1{i}(k)*PolyLeg
(tscale{i}(j),k-1);

est2{i}(j)=est2{i}(j)+p2{i}(k)*PolyLeg
(tscale{i}(j),k-1);


```

- plik Zad3

```

load MNUB_24L_P2_dane32.mat;

%Zad.3
KMxerr=zeros(20,19);
KMyerr=zeros(20,19);
KMVerr=zeros(20,19);
for M=5:20 %kod wzięty z dodatkowe
pętle by zmieniać wartości M i K
    for K=1:M-1
        tM=cell(1,20);
        tscale=cell(1,20);
        tperfM = cell(1,20);
        Fi=cell(1,20);

```

```

        end

X2{i}(j)=(est1{i}(j)*est1{i}(j)-
est2{i}(j)*est2{i}(j)+16)/8;

Y2{i}(j)=sqrt(est1{i}(j)*est1{i}(j)-
X2{i}(j)*X2{i}(j));
        if j>=2
            V2{i}(j-
1)=(sqrt((X2{i}(j)-X2{i}(j-
1))^2+(Y2{i}(j)-Y2{i}(j-
1))^2))/(t{i}(j)-t{i}(j-1));
        end
    end
    X2errabs{i}=abs(xref{i}'-X2{i});
    Y2errabs{i}=abs(yref{i}'-Y2{i});
    V2av(i)=mean(V2{i})*length(V2{i});
    x2sum=sum(X2errabs{i})+x2sum;
    y2sum=sum(Y2errabs{i})+y2sum;

lengthall2=lengthall2+length(V2{i});
    plot(X2{i},Y2{i});
end
v2av=sum(V2av)/lengthall2;
v2err=abs(v2av-vref);
x2errall=x2sum/(lengthall2+20);
y2errall=y2sum/(lengthall2+20);

function poly = PolyLeg(t,k) %funkcja
odpowiadająca za wyznaczenie
wielomianów Legendre
    if k==0
        poly=1;
    elseif k==1
        poly=t;
    else
        poly=(2*k-1)/k*t*PolyLeg(t,k-1)-
(k-1)/k*PolyLeg(t,k-2);
    end
end

```

```

        y=cell(1,20);
        p1=cell(1,20);
        p2=cell(1,20);

        for i=1:20
            tperfM{i} = linspace(0,
t{i}(length(t{i})), M);
            tM{i}=zeros(1,M);
            y{i}=zeros(M,2);

tscale{i}=zeros(1,length(t{i}));
            for j=1:M
                diffTmax=0;

```



```

        for m=1:length(t{i})
diffft=abs(tperfm{i}(j)-t{i}(m));
            if m==1

diffftmax=diffft;

tM{i}(j)=t{i}(m);

y{i}(j,1)=D{i}(m,1);
y{i}(j,2)=D{i}(m,2);
                end
            if diffft<diffftmax

diffftmax=diffft;

tM{i}(j)=t{i}(m);

y{i}(j,1)=D{i}(m,1);
y{i}(j,2)=D{i}(m,2);
                    end
                end
            tM{i}=tM{i}/tM{i}(M)*2-1;

tscale{i}=t{i}/t{i}(length(t{i}))*2-1;
        end

        for i=1:20
            Fi{i}=zeros(M,K);
            for j=1:M
                for k=0:K-1

Fi{i}(j,k+1)=PolyLeg(tM{i}(j),k);
                    end
                end

p1{i}=(Fi{i}'*Fi{i})\Fi{i}'*y{i}(:,1);
p2{i}=(Fi{i}'*Fi{i})\Fi{i}'*y{i}(:,2);
            end

            est1=cell(1,20);
            est2=cell(1,20);
            X2=cell(1,20);
            Y2=cell(1,20);
            V2av=zeros(1,20);
            X2errabs=cell(1,20);
            Y2errabs=cell(1,20);
            V2errabs=cell(1,20);
            V2=cell(1,20);
            lengthall2=0;
            x2sum=0;
            y2sum=0;
            hold on;
            for i=1:20

```

```

X2{i}=zeros(1,length(t{i}));
Y2{i}=zeros(1,length(t{i}));
V2{i}=zeros(1,length(t{i})-1);
est1{i}=zeros(1,length(t{i}));
est2{i}=zeros(1,length(t{i}));
                for j=1:length(t{i})
                    for k=1:K

est1{i}(j)=est1{i}(j)+p1{i}(k)*PolyLeg
(tscalet{i}(j),k-1);

est2{i}(j)=est2{i}(j)+p2{i}(k)*PolyLeg
(tscalet{i}(j),k-1);
                        end

X2{i}(j)=(est1{i}(j)*est1{i}(j)-
est2{i}(j)*est2{i}(j)+16)/8;

Y2{i}(j)=real(sqrt(est1{i}(j)*est1{i}(
j)-X2{i}(j)*X2{i}(j)));
                            if j>=2
                                V2{i}(j-
1)=(sqrt((X2{i}(j)-X2{i}(j-
1))^2+(Y2{i}(j)-Y2{i}(j-
1))^2))/(t{i}(j)-t{i}(j-1));
                                    end
                                end
                            X2errabs{i}=abs(X2{i}-
xref{i}');
                            Y2errabs{i}=abs(Y2{i}-
yref{i}');

V2av(i)=mean(V2{i})*length(V2{i});

x2sum=sum(X2errabs{i})+x2sum;

y2sum=sum(Y2errabs{i})+y2sum;

lengthall2=lengthall2+length(V2{i});
                end
            v2av=sum(V2av)/lengthall2;
            KMVerr(M,K)=abs(v2av-vref);

            KMXerr(M,K)=x2sum/(lengthall2+20);

            KMYerr(M,K)=y2sum/(lengthall2+20);
                end
            end

            figure;
            xlabel("M");
            ylabel("K");
            zlabel("Średni błąd bezwzględny
wyznaczania prędkości [m/s]");

```



```

        end
    end
    hold off;

    figure;
    xlabel("M");
    ylabel("K");
    zlabel("Średni błąd bezwzględny
    wyznaczania współrzędnej X [m]");
    grid on;
    hold on;
    for M=5:20
        for K=1:M-1

scatter3(M,K,KMXerr(M,K), 'filled');
            end
        end
    hold off;

    figure;

```

```

        xlabel("M");
        ylabel("K");
        zlabel("Średni błąd bezwzględny
    wyznaczania współrzędnej Y [m]");
        grid on;
        hold on;
        for M=5:20
            for K=1:M-1

scatter3(M,K,KMYerr(M,K), 'filled');
                end
            end
        hold off;

        function g = Gk(t,tk) %funkcja
    używana zamiast PolyLeg
            g=1/3/sqrt(2*pi)*exp((-
    1)/18*(t-tk)^2);
        end

```