

Laboratorium MATLA

Ćwiczenie 3.

Programowanie. Instrukcje sterujące. Typy danych.

Opracowali:

- dr inż. Beata Leśniak-Plewińska

Zakład Inżynierii Biomedycznej
Instytut Metrologii i Inżynierii Biomedycznej
Wydział Mechatroniki Politechniki Warszawskiej

Warszawa, 2018

Cel ćwiczenia

Celem ćwiczenia jest nabycie przez studentów umiejętności w zakresie programowania w MATLAB'ie, a w szczególności umiejętności zastosowania instrukcji sterujących oraz posługiwania się różnymi typami danych.

Referencje

Andrew P. King Paul Aljabar *MATLAB Programming for Biomedical Engineers and Scientists* 1st Ed. Academic Press, 2017

1. Przenośne urządzenie służy do monitorowania parametrów klinicznych, w tym częstości rytmu serca (HR, ang. *heartrate*). Producent urządzenia chciałby dodać nową funkcję, która byłaby odpowiedzialna za pojawianie się alarmu w sytuacji gwałtownej zmiany wartości HR.

Gwałtowna zmiana wartości HR ma miejsce wówczas, gdy bieżąca wartość HR jest mniejsza lub większa od średniej z pięciu poprzednich wartości o więcej niż 20 % wartości tej średniej.

Napisz funkcję `alerthr`, która będzie służyła do wykrywania gwałtownych zmian wartości HR. Funkcja `alerthr` będzie pobierać jako parametr wejściowy wektor wartości HR oraz częstotliwość wykonywania pomiarów wartości HR w Hz, a jako parametr wyjściowy będzie zwracać wektor chwil czasowych (w minutach), w których nastąpiły gwałtowne zmiany wartości HR oraz wektor, którego elementy będą miały wartości 1 dla chwili, w której wystąpił gwałtowny wzrost wartości HR i -1 dla jej spadku. Przyjmij, że początek pomiarów wartości HR odpowiada chwili równej 0 (pierwszy pomiar odpowiada chwili czasowej równej 0 [min]).

Zapewnij kontrolę błędów. Np. w przypadku, gdy argumentem wejściowym funkcji `alerthr` będzie pusta macierz, funkcja ta powinna zakończyć swoje działanie i wyświetlić odpowiedni komunikat o przyczynie błędu.

Pliki *heart1.txt* i *heart2.txt* zawierają wartości częstotliwości rytmu serca zarejestrowane z częstotliwością 1/60 Hz.

Napisz skrypt, w którym z plików będą ładowane dane – wektor wartości częstości rytmu serca. Następnie, w celu zbadania czy w zarejestrowanych danych występuje gwałtowna zmiana częstotliwości rytmu serca, wywoływana będzie funkcja `alerthr`. Dalej, wyniki tej funkcji posłużą do ewentualnego wyświetlenia komunikatów - dla każdej wykrytej gwałtownej zmiany HR, powinien zostać wyświetlony komunikat o treści:

UWAGA! W minucie wystąpił częstotliwości rytmu serca.

W wykropkowanych miejscach powinna zostać podana wartość chwili czasowej, w której wystąpiła gwałtowna zmiana częstotliwości rytmu serca oraz informację czy był to spadek czy wzrost tej częstotliwości.

W przypadku, gdy żadna gwałtowna zmiana wartości HR nie zostanie wykryta, powinien zostać wyświetlony komunikat o treści:

UWAGA! Nie stwierdzono gwałtownej zmiany częstotliwości rytmu serca.

Wyniki uzyskane dla obu plików danych: *heart1.txt* i *heart2.txt* wpisz w odpowiednie rubryki *Sprawozdania*. W przypadku gdy zaproponowany w *Sprawozdaniu* komunikat nie wystąpił, wykreśl go, np.

~~UWAGA! Nie stwierdzono gwałtownej zmiany częstotliwości rytmu serca.~~

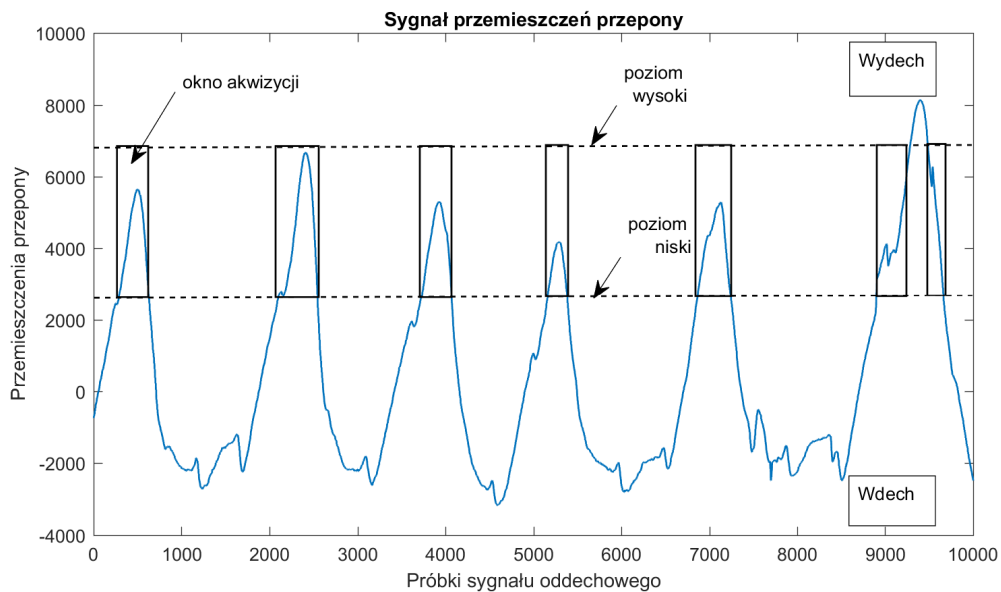
WSKAZÓWKA. W realizacji zadania mogą być przydatne wbudowane funkcje MATLAB'a: `error`, `load`, `find`.

2. Wiele technik obrazowania medycznego rejestruje sekwencję danych obrazowych w pewnym przedziale czasu. Bramkowanie jest techniką pozwalającą na zredukowanie artefaktów w wynikowym obrazie, wynikających z cyklicznych ruchów (np. uderzeń serca,

oddechu). Technika ta polega na akceptowaniu jedynie danych zarejestrowanych mniej więcej w tej samej fazie ruchu i odrzuceniu pozostałych. W celu podjęcia decyzji czy dane można zaakceptować, czy też należy je odrzucić można wykorzystać sygnał związany z danym ruchem (np. sygnał EKG, sygnał z czujnika rejestrującego ruch oddechowy). Stosunek liczby obrazów zaakceptowanych do całkowitej liczby obrazów zarejestrowanych jest nazywany efektywnością skanowania.

W przypadku tomografii z użyciem metody rezonansu magnetycznego do bramkowania obrazowania wykorzystuje się często sygnał z czujnika rejestrującego ruch oddechowy umieszczonego nad przeponą pacjenta. (Rys. 1)

Aby podjąć decyzję, które obrazy zaakceptować, a które odrzucić, należy wyznaczyć tzw. okna akwizycji. Są one często wyznaczone poprzez wybór minimalnego i maksymalnego poziomu sygnału pochodzącego z czujnika oddechu. (Rys. 1) Poziomy sygnał bramkującego ograniczające okna akwizycji dobiera się na podstawie sygnału zarejestrowanego w trakcie fazy przygotowawczej.



Rysunek 1: Proste bramkowanie oddechem. Akceptowane są jedynie te obrazy, które są rejestrowane w czasie trwania okna akwizycji wyznaczonego w fazie wydechu poprzez dwa poziome graniczne wartości sygnału z czujnika oddechu.

Napisz funkcję MATLAB'a `okno_akwizycji`, która korzystając z sygnału pochodzącego z czujnika oddechu oraz dodatkowych informacji: pożądanej wydajności skanowania oraz długości fazy przygotowawczej, wyznaczy wartości progowe dla fazy wydechu definiujące okna akwizycji.

WSKAZÓWKA: Wysoki poziom sygnału wyznaczającego granicę okna akwizycji możesz przyjąć jako równy wartości maksymalnej sygnału z czujnika oddechu zarejestrowanego w trakcie fazy przygotowawczej. Wartość poziomu minimalnego dobierz tak, aby procent danych zaakceptowanych w trakcie fazy przygotowawczej odpowiadał pożądanej wydajności skanowania.

Napisz skrypt `bramkowaieMRI`, w którym dane z pliku `.mat` zawierającego sygnały z czujnika oddechu zamocowanego nad przeponą pacjenta będą ładowane do przestrzeni roboczej. Dalej, skrypt powinien umożliwiać użytkownikowi wprowadzenie długości fazy przygotowawczej (w próbkach) oraz pożądaną wydajność skanowania.

Następnie, za pomocą funkcji `okno_akwizycji` w skrypcie wyznaczone zostaną niski i wysoki poziom definiujący okna akwizycji. Korzystając z wyników funkcji `okno_akwizycji` skrypt powinien wyszukać indeksy wszystkich próbek sygnału z czujnika oddechu zawarte w oknach akwizycji. Następnie powinna zostać wyznaczona rzeczywista wydajność skanowania. Na koniec, wartości poziomów użytych do wyznaczenia okien akwizycji oraz wartość rzeczywistej wydajności skanowania powinny zostać wyświetlone w Oknie poleceń w postaci zdania:

Wartość poziomu niskiego wynosi ..., a wysokiego Rzeczywista wydajność skanowania wynosi...

Plik `bellows.mat` zawiera sygnały z czujnika oddechu zamocowanego nad przeponą pacjenta.

Przetestuj działanie skryptu `bramkowaieMRI` i funkcji `okno_akwizycji` dla pliku `bellows.mat`. Przyjmij, że faza przygotowawcza obejmuje odcinek pierwszych 5000 próbek sygnału z czujnika oddechu, a pożądana wydajność skanowania wynosi 25%.

Uzyskane wyniki wpisz w odpowiednie rubryki *Sprawozdania*.

WSKAZÓWKA: W realizacji zadania może być przydatna wbudowane funkcje MATLAB'a: `find`, `sort`.

3. Badania kliniczne nowego środka odbywają się w kilku fazach. W trakcie I fazy badań klinicznych ocenia się bezpieczeństwo stosowania testowanego środka. Badane jest jego wchłanianie z przewodu pokarmowego, poziom jego stężenia we krwi i tkankach, wydalanie, toksyczność oraz interakcję z pożywieniem i powszechnie stosowanymi lekami. Wyniki tej części prac nad nowym lekiem pozwalają wstępnie określić jego dawkowanie poprzez badanie zależności pomiędzy dawką leku oraz odpowiedzią kliniczną.

W celu oceny skuteczności i toksyczności środka, początkowo niska dawka jest zwiększana. Większa dawka jest stosowana u nowej grupy uczestników badań. Proces zwiększania dawki jest kontynuowany dopóki toksyczność środka nie stanie się zbyt duża. Dawka bywa zwiększana kolejno o 67% dla drugiej, o 50% dla trzeciej, o 40% dla czwartej i o 33% dla każdej kolejnej grupy uczestników. Często stosowany jest następujący schemat zwiększania dawki, zwany „3+3” stosowany dla 3-osobowych grup uczestników badań:

- jeśli u żadnego z trzech uczestników z danej grupy nie wystąpił nieakceptowalny poziom toksyczności, kolejna grupa trzech uczestników będzie stosowała środek w wyższej dawce (podwyższanie dawki dla kolejnych grup przebiega zgodnie z powyższą zasadą o odpowiedni % wartość dawki stosowanej w badaniu dla poprzedniej grupy),
- jeśli dla jednego z uczestników badań z danej grupy wystąpił nieakceptowalny poziom toksyczności, wówczas dla kolejnej grupy stosowana jest taka sama dawka środka jak dla danej grupy (dawka jest powtórzona),

- procedura zwiększania dawki jest kontynuowana dopóty, dopóki u co najmniej dwóch na trzech uczestników z danej grupy nie stwierdzi się toksyczności na nieakceptowalnym poziomie. Dawka rekomendowana dla drugiej fazy badań jest definiowana jako największa dawka mniejsza od tej ostatniej.

Napisz skrypt `dawkaIIIfaza`, w którym z pliku `.mat` będą ładowane wyniki dla I fazy badań klinicznych cytotoksyny. Dane zawierają wartości dla kolejnych zastosowanych dawek oraz odpowiadające im wyniki toksyczności. Dane są zapisane w tablicy komórkowej o rozmiarze 1×2 . Pierwsza komórka zawiera macierz wartości poziomów dawek w mg (określonych zgodnie z powyższym schematem) dla kolejnych trzyosobowych grup. Druga komórka natomiast zawiera macierz wartości logicznych. Każdy wiersz tej macierzy zawiera informację o poziomie toksyczności dla poszczególnych uczestników danej grupy. Wartość 1 oznacza zbyt wysoka toksyczność, 0 – akceptowalną toksyczność. Na podstawie tych danych skrypt ma wyznaczać dawkę dla II fazy badań klinicznych tego samego środka. Informację o wyznaczonej wartości dawki ma być wyświetlana w *Oknie Poleceń* w postaci zdania:

Dawka dla II-ej fazy badań cytotoksyny, wyznaczona na podstawie wyników I-ej fazy badań wynosi

Zadbaj o to, aby kod był odporny na nietypowe sytuacje, np. co się stanie gdy już w pierwszej grupie uczestników, u dwóch z nich zostanie stwierdzony wysoki poziom toksyczności?

Przetestuj działanie skryptu dla wyników dla pierwszej fazy badań klinicznych cytotoksyny zawartych w pliku `phaseI_data1.mat` i `phaseI_data2.mat`.

Wyniki wpisz w odpowiedniej rubryce *Sprawozdania*.

4. Wskaźnik ciężkości obrażeń (ISS – ang. *Injury Severity Score*) to miara służąca do oceny jak poważnych urazów doznała dana osoba w wyniku wypadku. Wskaźnik ten jest obliczany w następujący sposób. Ciało jest dzielone na 6 obszarów: głowa, twarz, klatka piersiowa, jama brzuszna i kręgosłup, kończyny oraz tzw. obrażenia ogólne. Każdemu urazowi w obrębie danego obszaru przypisywana jest wartość punktowa wynikająca ze skróconej skali obrażeń (AIS – ang. *Abbreviated Injury Scale*), która dzieli obrażenia ciała na: lekkie (1), średnie (2), poważne nie zagrażające życiu (3), poważne zagrażające życiu (4), oraz krytyczne (5). Do wyznaczenia wskaźnika ISS, dla każdego obszaru wybierana jest tylko najwyższa spośród wartości w skali AIS przypisanych różnym urazom doznany przez pacjenta w danym obszarze. Wskaźnik ISS jest wyznaczany jako suma kwadratów 3 najwyższych spośród wartości w skali AIS przypisanych wszystkim obszarom dla danego pacjenta. Najwyższa możliwa liczba punktów w skali ISS wynosi więc $3 \times 5^2 = 75$.

Napisz funkcję `wyznaczISS`, która na podstawie 6-elementowego wektora wartości punktowych w skali AIS dla danego pacjenta, będzie wyznaczać wartość wskaźnika ISS dla tego pacjenta.

Napisz skrypt `ISScore`, w którym z pliku `.mat` będą ładowane dane zapisane w postaci wektora struktur. Pojedyncza struktura zawiera 5 pól, odpowiadających poszczególnym obszarom ciała, a każde z pól zawiera jedną, najwyższą dla tego obszaru ciała wartość punktową w skali AIS. Poszczególne pola struktury noszą nazwy: `head`, `face`, `chest`, `abdomen`, `extremities` i `external`. Dalej, za pomocą funkcji `wyznaczISS`, dla każdego pacjenta zostanie wyznaczona wartość wskaźnika ISS. Następnie, do struktury danych dla danego pacjenta zostanie dodane nowe pole o nazwie `iss` i zawartości równej

wyznaczonej wartości wskaźnika ISS. Na koniec, po wyznaczeniu i dodaniu do struktury danych wartości wskaźnika ISS dla każdego pacjenta, nowe dane zostaną zapisane w pliku *iis.mat*.

Zadbaj o kontrolę błędów. Np. sprawdź czy pola struktury danych załadowanych z pliku mają nazwy zgodne z podanymi wyżej. Jeśli nie, skrypt powinien przerwać działanie i wyświetlić odpowiedni komunikat.

Przetestuj działanie skryptu `ISscore` i funkcji `wyznaczISS` dla plik *ais.mat*.

Wartości punktowe w skali AIS i wyznaczone wartości wskaźnika ISS dla danego pacjenta (dla danych w pliku *ais.mat*) wpisz w odpowiednie rubryki *Sprawozdania*.

WSKAZÓWKA: W realizacji zadania może być przydatna wbudowane funkcje MATLAB'a: `fieldnames`, `isfield`, `strcmp`, `strfind`.

Sprawozdanie

Ćwiczenie 3. Programowanie. Instrukcje sterujące. Typy danych.

L.p.	Imię i nazwisko	Grupa	Data
Punkt cw./ L. punktów	Realizacja/wynik		Uwagi prowadzącego
1 / 1	Plik <i>heart1.txt</i>		
	UWAGA! W minucie wystąpił częstotliwości rytmu serca.		
	UWAGA! W minucie wystąpił częstotliwości rytmu serca		
	UWAGA! W minucie wystąpił częstotliwości rytmu serca.		
	UWAGA! W minucie wystąpił częstotliwości rytmu serca.		
	UWAGA! W minucie wystąpił częstotliwości rytmu serca.		
	UWAGA! W minucie wystąpił częstotliwości rytmu serca.		
	UWAGA! W minucie wystąpił częstotliwości rytmu serca.		
	UWAGA! Nie stwierdzono gwałtownej zmiany częstotliwości rytmu serca.		
	Plik <i>heart1.txt</i>		
	UWAGA! W minucie wystąpił częstotliwości rytmu serca.		
	UWAGA! W minucie wystąpił częstotliwości rytmu serca.		
	UWAGA! W minucie wystąpił częstotliwości rytmu serca.		
UWAGA! Nie stwierdzono gwałtownej zmiany częstotliwości rytmu serca.			

2 / 1,5	<p>Wartość poziomu niskiego wynosi, a wysokiego</p> <p>Rzeczywista wydajność skanowania wynosi</p>								
3 / 1	<p>Plik <i>phaseI_data1.mat</i></p> <p>Dawka dla II-ej fazy badań cytotoksyny, wyznaczona na podstawie wyników I-ej fazy badań wynosi [mg].</p> <p>Plik <i>phaseI_data2.mat</i></p> <p>Dawka dla II-ej fazy badań cytotoksyny, wyznaczona na podstawie wyników I-ej fazy badań wynosi [mg].</p>								
4 / 1,5	Numer pacjenta	AIS						ISS	