

Politechnika Wrocławska
Instytut Techniki Ciepłej i Mechaniki Płynów

PREPRINT: NR 3/2008

PRACA DOKTORSKA

Przyrząd do bezinwazyjnego pomiaru
fali tętna krwi

mgr inż. Jacek Kosek

Promotor: Prof. dr hab. inż. Mirosław Werszko

słowa kluczowe:
wzmacniacz typu dysza-przysłona, pneumatyczny czujnik
ciśnienia, fala tętna krwi

krótkie streszczenie:

w pracy przedstawiono metodę bezinwazyjnego pomiaru fali tętna krwi czujnikiem pneumatycznym oraz budowę prototypu przyrządu do pomiaru, rejestracji i analizy tej fali.

Wrocław 2008

Spis treści

Wykaz symboli:	5
1. Wstęp	7
2. Cel, teza i zakres pracy	13
3. Budowa przyrządu	14
3.1. Opis ogólny przyrządu i sposobu pomiaru	14
3.2. Budowa i zasada działania czujnika	16
3.3. Moduł zasilający	19
3.4. Interfejs do współpracy czujnika pneumatycznego z komputerem	21
3.4.1. Część elektroniczna	21
3.4.2. Oprogramowanie mikrokontrolera	22
3.5. Oprogramowanie komputera	25
3.5.1. Zadania programu komputerowego	25
3.5.2. Pętla główna programu	26
3.5.3. Analiza punktu pomiarowego	27
3.5.4. Analiza pojedynczej fali tętna	29
4. Model matematyczny czujnika	31
4.1. Cel i metoda modelowania	31
4.2. Identyfikacja wzmacniacza typu dysza-przysłona	33
4.2.1. Opis wzmacniacza typu dysza-przysłona	33
4.2.2. Metoda opracowania modelu	34
4.2.3. Model kapilary	35
4.2.4. Model elementu dysza-przysłona	37
4.2.5. Model własności statycznych wzmacniacza typu dysza-przysłona	39
4.2.6. Model własności dynamicznych wzmacniacza typu dysza-przysłona	41
4.3. Identyfikacja membrany	44
4.3.1. Charakterystyka statyczna	45

4.3.2.	Charakterystyka impulsowa	45
4.4.	Model czujnika i jego weryfikacja	47
4.4.1.	Charakterystyka statyczna	48
4.4.2.	Charakterystyka dynamiczna	49
5.	Dobór parametrów czujnika	52
5.1.	Kryterium oceny czujnika	52
5.2.	Symulacje pracy czujnika	54
5.3.	Ocena poprawnej pracy czujnika na podstawie modelu	55
5.4.	Dobór parametrów konstrukcyjnych czujnika	56
6.	Ocena przydatności przyrządu do pomiaru fali tętna krwi	58
6.1.	Badanie na modelu fizycznym tętnicy	58
6.2.	Badania laboratoryjne na pacjentach	62
6.3.	Badania kliniczne	66
7.	Podsumowanie i wnioski	70
A.	Źródła programu zainstalowanego w mikrokontrolerze	72
A.1.	Lista zmian w pliku: „autofiles/usbcfg.h”	72
A.2.	Plik nagłówkowy: „user/user.h”	72
A.3.	Implementacja, plik: „user/user.c”	73
A.4.	Plik nagłówkowy: „realtime.h”	79
A.5.	Implementacja, plik: „realtime.c”	79
B.	Źródła programu zainstalowanego w komputerze	83
C.	Źródła S-funkcji wykorzystanych w modelu czujnika	89
C.1.	S-funkcja modelu membrany: „membrana_s.m”	89
C.2.	S-funkcja modelu wzmacniacza: „wzmacniacz_s.m”	91
C.3.	Funkcja licząca pojemność pneumatyczną: „cfun.m”	93
C.4.	Funkcja licząca strumień kapilary: „kapilfun.m”	93
C.5.	Funkcja licząca strumień elementu dysza-przysłona: „dyszprzyfun.m”	95
C.6.	Funkcja aproksymująca współczynnik α : „fun_aprok_alfa3.m”	96
D.	Skrypty wykorzystane przy doborze parametrów czujnika	97
D.1.	Skrypt uruchamiający symulacje: „eksperymenty.m”	97
D.2.	Skrypt obliczający kryterium: „bledy_2.m”	98

<i>Spis treści</i>	4
<hr/>	
Bibliografia	100
Spis rysunków	102
Spis tabel	104

Wykaz symboli:

- α – współczynnik przepływu,
- $\hat{\alpha}$ – aproksymowany współczynnik przepływu,
- ΔF – różnica sił działających na membranę,
- Δp – różnica ciśnień działających na membranę,
- Δp_k – spadek ciśnienia na kapilarze,
 - λ – współczynnik strat liniowych,
- μC – mikrokontroler,
- μ – współczynnik lepkości dynamicznej powietrza,
- σ – błąd modelu,
- ξ – współczynnik tłumienia membrany,
- ζ – współczynnik strat miejscowych (na wylocie i wlocie),
- A_1 – pole czynnej powierzchni membrany od strony ciśnienia p_1 ,
- A_2 – pole czynnej powierzchni membrany od strony ciśnienia p_2 ,
- $a_{0...3}$ – współczynniki aproksymacji,
 - a – pierwsza amplituda oscylacji membrany,
 - b – druga amplituda oscylacji membrany,
- C – sztywność membrany,
- d_1 – średnica kapilary,
- d_2 – średnica dyszy,
- d_3 – średnica przewodu zasilającego,
- F – siła docisku czujnika,
- F_1 – siła wywołana ciśnieniem p_1 ,

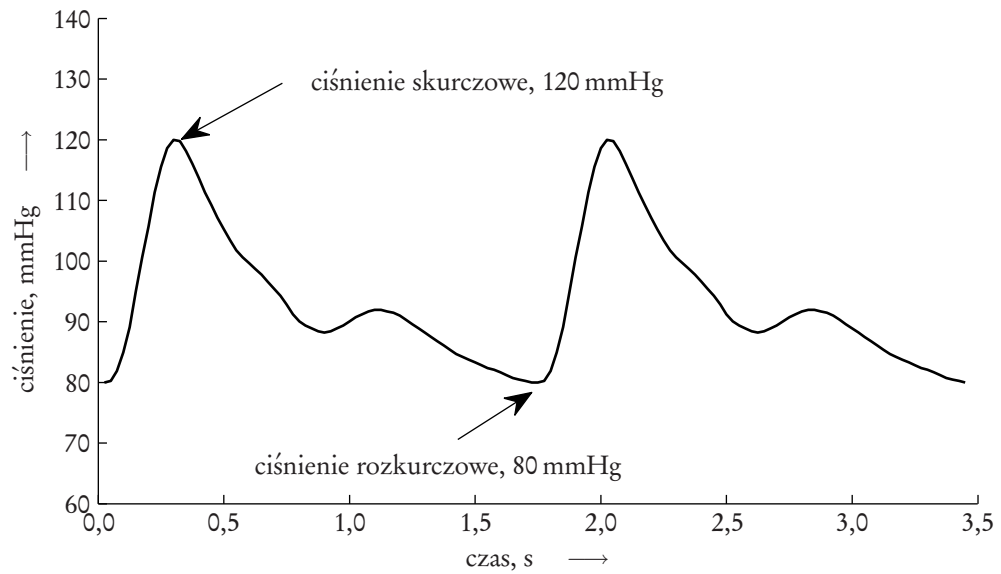
-
- F_2 — siła wywołana ciśnieniem p_2 ,
 f_0 — zależność ugięcia membrany od różnicy sił działających na nią,
 f_1 — zależność ciśnienia wzmacniacza typu dysza-przysłona od otwarcia dyszy,
 f_2 — zależność ugięcia membrany od różnicy ciśnień działających na nią,
 l — długość dławika,
 l_1 — długość przewodu zasilającego,
 p_0 — ciśnienie atmosferyczne,
 p_1 — ciśnienie działające na czujnik z zewnątrz,
 p_2 — ciśnienie wyjściowe czujnika,
 p_z — ciśnienie zasilania,
 p_t — ciśnienie tętnicze,
 q_{m1} — strumień masy przepływający przez kapilarę,
 q_{m2} — strumień masy przepływający przez zespół dysza przysłona,
 R — indywidualna stała gazowa powietrza,
 Re — liczba Reynoldsa,
 T — temperatura otoczenia w Kelwinach,
 T_s — stała czasowa membrany,
 T_w — stała czasowa wzmacniacza typu dysza-przysłona,
 T_{os} — okres oscylacji membrany,
 U — napięcie wyjściowe przetwornika pneumoelektrycznego,
 V — objętość przewodu zasilającego,
 x — otwarcie dyszy,
 x_0 — odsadzenie dyszy,
 y — ugięcie membrany,

1. Wstęp

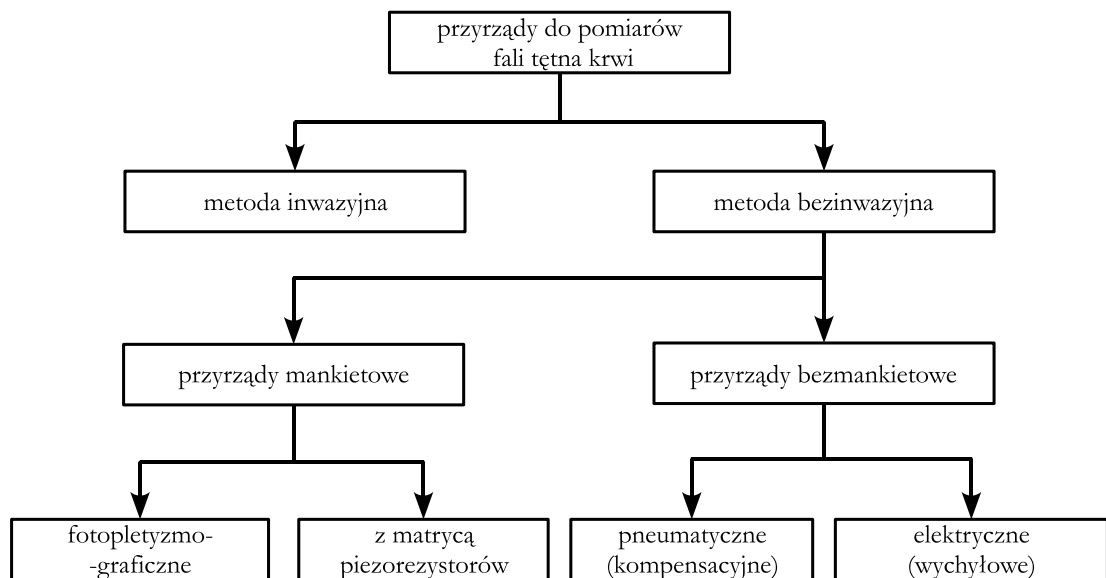
Choroby układu krążenia są obecnie na pierwszym miejscu pod względem przedwczesnych zgonów. W Polsce, choroby te są przyczyną 48% ogólnej liczby zgonów. Dla porównania — choroby nowotworowe powodują 23% całkowitej liczby zgonów. Nic też dziwnego, że profilaktyce, leczeniu oraz diagnozowaniu chorób układu krążenia poświęca się na całym świecie bardzo dużo uwagi.

Obecnie najczęściej stosowanym przyrządem do diagnozowania układu krążenia jest przyrząd mankietowy wykorzystujący tzw. metodę tonów Koratkowa. Przyrząd ten mierzy dwie wartości ciśnienia krwi: ciśnienie skurczowe oraz ciśnienie rozkurczowe. Te dwie wartości są bardzo istotne, zwłaszcza przy wykrywaniu choroby nadciśnienia ewentualnie niedociśnienia, ale nie daje pełnego obrazu stanu naczyń krwionośnych oraz serca pacjenta. Stwierdzenie na przykład nadciśnienia nie pozwala na podjęcie skutecznego leczenia do póki nie zostanie wykryta przyczyna tej choroby: wada lewej komory serca, cukrzyca, stwardnienie tętnic i inne. Znacznie więcej informacji w tym zakresie uzyskuje się z przebiegu w czasie ciśnienia krwi, czyli z tzw. fali tętna krwi (rys. 1.1). Maksimum funkcji ciśnienia w czasie nazywamy ciśnieniem skurczowym, natomiast minimum — ciśnieniem rozkurczowym.

Pomiar fali tętna krwi może odbywać się metodą inwazyjną (krwawą) lub bezinwazyjną (bez naruszania całości tętnicy) (rys. 1.2). Przyrządy do pomiarów inwazyjnych wymagają wprowadzenia do wnętrza badanej tętnicy sondy pomiarowej, której drugi koniec łączy się z przetwornikiem pomiarowym. Przyrządy tego typu mierzą falę tętna krwi ze stosunkowo najlepszą wiarygodnością. Jednak ze względu na ryzyko infekcji oraz możliwość pojawienia się zakrzepów, pomiary inwazyjne przeprowadza się wyłącznie w szpitalach.



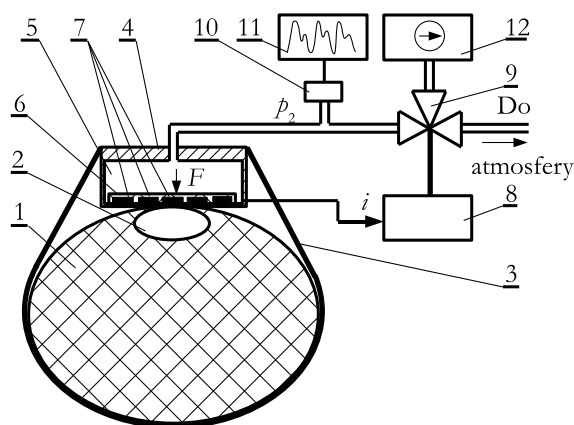
Rysunek 1.1. Przykładowa fala tętna krwi w tętnicy promieniowej pacjenta z ciśnieniem skurczowym i rozkurczowym 120/80.



Rysunek 1.2. Podział przyrządów do pomiarów fali tętna krwi.

Do wyznaczania fali tętna w sposób bezinwazyjny, czyli bez nakłuwania bądź przecinania tętnicy, służą obecnie przyrządy dwóch typów: mankietowe i bezmankietowe (dotykowe). Przyrządy mankietowe oparte są głównie na fotopletyzmografie [1] i matrycy piezorezystorów [2]. Natomiast przyrządy dotykowe mają czujniki piezorezystorowe lub pneumatyczne.

Przyrządy mankietowe, służące do pomiaru fali tętna krwi, działają na zasadzie kompensacji ciśnień i wymagają zakładania na kończynę pacjenta mankietu zasilanego sprężonym powietrzem. Przyrządy te miały swe początki już w latach 70 zeszłego wieku.

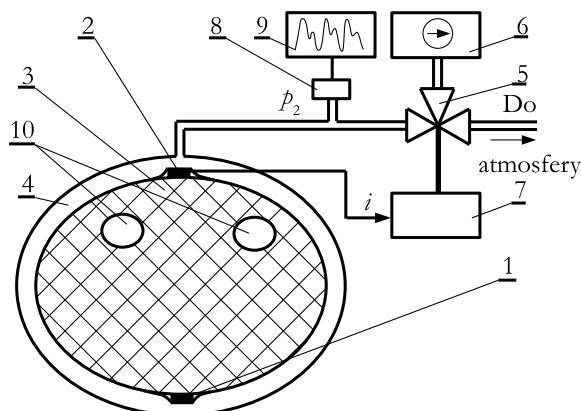


Rysunek 1.3. Uproszczony schemat przyrządu z matrycą piezorezystorów, 1 — nadgarstek, 2 — tętnica, 3 — bransoleta, 4 — czujnik odkształcenia, 5 — elastyczna komora pneumatyczna, 6 — membrana, 7 — piezorezystory, 8 — regulator przemieszczenia (odkształcenia), 9 — zawór trójdrożny, 10 — pneumoelektryczny przetwornik pomiarowy ciśnienia, 11 — mikroprocesorowy układ przetwarzający z wyświetlaczem, 12 — źródło sprężonego powietrza.

Przykładowym przyrządem mankietowym do pomiaru fali tętna krwi jest przyrząd z matrycą piezorezystorów. Uproszczony schemat tego przyrządu, pokazano na rysunku 1.3. Na rysunku tym jest widoczny przekrój nadgarstka 1 wraz z tętnicą promieniową 2, do której, za pośrednictwem bransolety 3, jest dociśnięty czujnik 4. Czujnik ten posiada elastyczną komorę 5 zamkniętą szczelnie membraną 6 z piezorezystorami 7, połączonymi z wejściem regulatora 8, który za pośrednictwem zaworu

trójdrożnego 9 steruje ciśnieniem p_2 , w ten sposób aby nie dopuścić do odkształcenia ścianki tętnicy 2 względem jej neutralnego (spłaszczonego) położenia. W wyniku tego zmiany, ciśnienia p_2 podążają za zmianami ciśnienia krwi w tętnicy. Ciśnienie p_2 (ciśnienie kompensujące) jest przetwarzane w przetworniku 10 na sygnał elektryczny i przesyłane do rejestratora 11 z wyświetlaczem graficznym.

Różnice w konstrukcji przyrządów mankietowych zależą głównie od typu czujnika służącego do pomiaru odkształcenia tętnicy, i tak w przyrządzie z matrycą piezorezystorów czujnik ten stanowi krzemowa wydłużona płytką (membrana), zawierająca na swej powierzchni około 100 odpowiednio rozmieszczonych miniaturowych piezorezystorów 7, połączonych w mostki Wheatstone'a [3]. Dzięki takiemu „zagęszczeniu” piezorezystorów, przykładając czujnik z matrycą piezorezystorów do nadgarstka, można mieć pewność, że przynajmniej jeden z nich znajdzie się dokładnie nad badaną tętnicą. Sygnał wyjściowy tego piezorezystora jest wykorzystywany do dalszej analizy.



Rysunek 1.4. Uproszczony schemat przyrządu z fotopletyzmoграфem, 1 — dioda świecąca LED, 2 — fotorezystor, mierzący natężenie światła przechodzącego przez tkanki, 3 — palec, 4 — mankiet pneumatyczny, 5 — zawór trójdrożny, 6 — źródło sprężonego powietrza, 7 — regulator, 8 — pneumatyczny przetwornik pomiarowy ciśnienia, 9 — mikroprocesorowy układ przetwarzający z wyświetlaczem, 10 — tętnice w palcu.

W przyrządzie z fotopletyzmoграфem czujnik odkształcenia stanowią: źródło światła podczerwonego 1 i fotorezystor 2 (rys. 1.4) umieszczone po przeciwnych stro-

nach palca 3. Strumień światła, przechodząc przez ten palec, jest pochłaniany głównie przez krew zawartą w naczyniach krwionośnych; sygnał wyjściowy *i* fotorezystora jest maksymalny gdy naczynia krwionośne na odcinku pomiarowym są całkowicie zaciśnięte (zatrzymanie przepływu krwi), a minimalny gdy naczynia krwionośne utrzymują swoją maksymalną objętość (podczas działania ciśnienia skurczowego). Sygnał wyjściowy *i* fotorezystora 2 stanowi więc miarę odkształceń ścian naczyń krwionośnych, podczas ich uciskania za pomocą mankietu pneumatycznego 4. Gdy amplituda wahań sygnału *i* osiąga maksymalną wartość oznacza to, że ścianka tętnicy jest spłaszczona właściwie, a zmiany ciśnienia p_2 podążają za zmianami ciśnienia w tętnicach.

Przyrządy mankietowe mają charakter aparatury stacjonarnej, używanej do monitorowania pacjentów w szpitalu głównie po operacjach serca. Wymagają one wyspecjalizowanej aparatury i są stosunkowo drogie [4].

W przyrządach bezmankietowych, czujnik siły lub ciśnienia przykłada się bezpośrednio do ciała pacjenta w miejscu, gdzie pod skórą znajduje się tętnica z wyraźnym tętnem. Przyrządy bezmankietowe mogą być typu kompensacyjnego lub wychyłowego (rys. 1.2). Te pierwsze oparte są na czujniku pneumatycznym, a drugie na czujniku piezorezystorowym. Przykładowy czujnik piezorezystorowy ma postać krótkiego cylinderka, z jednego końca którego wystaje tłoczek o średnicy ok. 9 mm [5]. W czasie pomiaru, na tłoczek ten przenoszona jest siła, wywołana działaniem ciśnienia krwi na spłaszczony przez czujnik odcinek badanej tętnicy. Siła ta, mierzona za pośrednictwem piezorezystora, jest miarą przebiegu ciśnienia krwi. Ponieważ jednak pole powierzchni, na której powstaje siła, jest nieokreślone, przyrząd wymaga przed każdym pomiarem wzorcowania na pacjencie. Dokonuje się tego z reguły za pomocą tradycyjnego przyrządu mankietowego do pomiaru ciśnienia skurczowego i rozkurczowego.

Przyrządy bezmankietowe znane są jedynie z doniesień literaturowych. Przyrządy z czujnikami piezorezystorowymi były testowane i stosowane w szpitalach

w Sidney [6] oraz w Singapurze [5]. Zaś przyrząd z czujnikiem pneumatycznym powstał w Zakładzie Automatyki Instytutu Techniki Ciepłej i Mechaniki Płynów i jest przedmiotem niniejszej pracy doktorskiej.

Przyrządy bezmankietowe wypełniają lukę jaka istnieje pomiędzy stosunkowo tanimi i łatwo dostępnymi przyrządami do pomiaru ciśnienia skurczowego i rozkurczowego, a drogimi, szpitalnymi przyrządami do bezinwazyjnego pomiaru fali tętna krwi. Przyrządy bezmankietowe mierzą bowiem falę tętna krwi, jak przyrządy szpitalne ale ich cena jest na poziomie ceny przyrządów do pomiaru ciśnienia skurczowego i rozkurczowego. Przyrządy bezmankietowe mogą być stosowane nie tylko w szpitalach ale również w ambulatoriach, przychodniach i domach pacjentów, a lekarze pierwszego kontaktu mogą z pomocą tych przyrządów przeprowadzać badania przesiewowe pod względem ryzyka zawału serca i udaru mózgowego, czego dotąd nie byli w stanie robić.

Kolejną zaletą przyrządów bezmankietowych jest możliwość wyznaczania fali tętna krwi w różnych miejscach ciała pacjenta, podczas gdy pomiary tej fali przyrządami mankielowymi są ograniczone do kończyn, gdyż tylko na nich można założyć mankiel. Na przykład miejscem pomiaru fali przyrządami mankielowymi z fonopletyzmografem jest z reguły palec ręki, a przyrządami z matrycą piezorezystorów — nadgarstek lub przedramię. W miejscach tych fala tętna jest już znacznie wytłumiona, a informacje, które można z niej uzyskać są znacząco ograniczone.

2. Cel, teza i zakres pracy

Celem pracy jest opracowanie metody bezinwazyjnego pomiaru fali tętna krwi czujnikiem pneumatycznym oraz budowa prototypu przyrządu do pomiaru, rejestracji i analizy tej fali.

Teza. Za pomocą pneumatycznego czujnika ciśnienia, opartego na wzmacniaczu typu dysza-przysłona o odpowiednio dobranych parametrach i objętego ujemnym sprzężeniem zwrotnym, można zbudować przyrząd do bezinwazyjnego pomiaru i rejestracji fali tętna krwi.

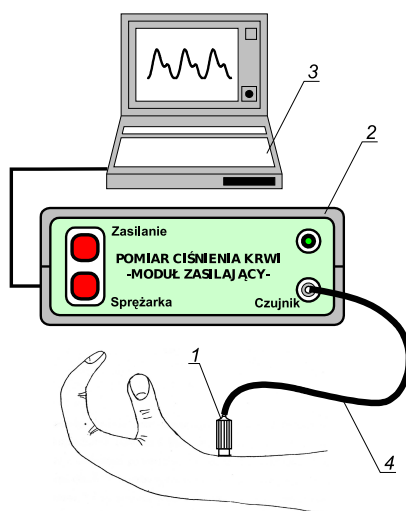
Zakres wykonanych prac:

1. Projekt i modelowanie matematyczne czujnika oraz badanie jego charakterystyk: statycznych i dynamicznych
2. Projekt, wykonanie i oprogramowanie interfejsu do współpracy czujnika z komputerem
3. Opracowanie algorytmu i programu komputerowego do analizy fali tętna krwi
4. Budowa modelu fizycznego tętnicy i otaczającej ją tkanki
5. Badania przyrządu na modelu tętnicy
6. Badanie przyrządu na pacjentach

3. Budowa przyrządu

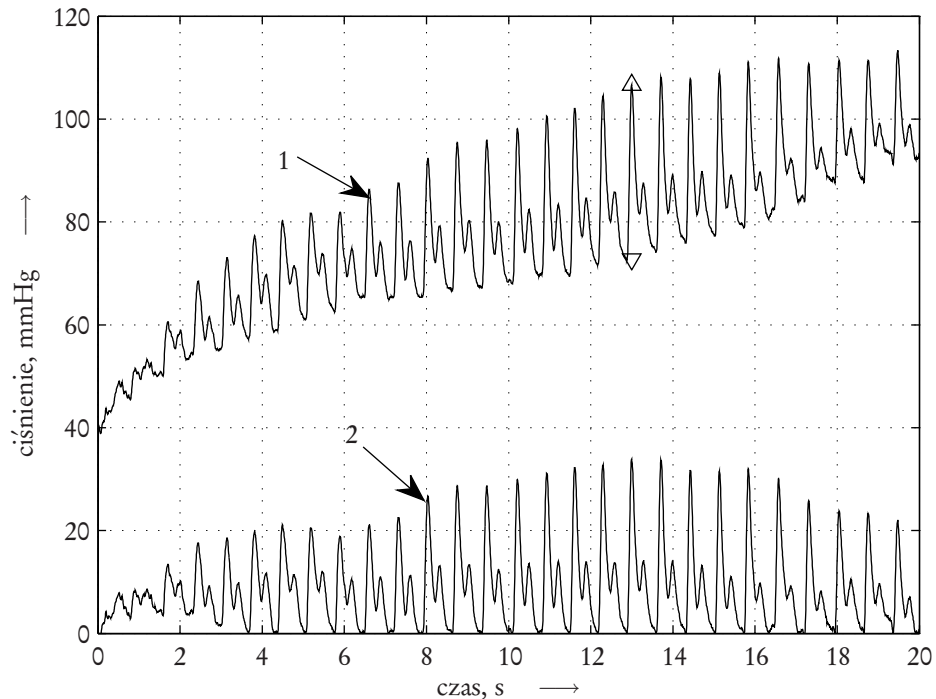
3.1. Opis ogólny przyrządu i sposobu pomiaru

Przyrząd składa się z czujnika pneumatycznego 1 (rys. 3.1), który, za pomocą elastycznego przewodu pneumatycznego 4, jest połączony z modułem zasilającym 2. Moduł ten ma do wykonania kilka zadań: dostarczyć do czujnika sprężone powietrze o stabilnym ciśnieniu, (opis tej funkcji modułu zamieszczony jest w punkcie 3.3), mierzyć i przetwarzać na sygnał cyfrowy ciśnienie wyjściowe czujnika 1, wstępnie filtrować sygnał ciśnieniowy i przesyłać go do komputera 3 (opisano to w punkcie 3.4). Komputer 3 pełni rolę rejestratora i analizatora fali tętna krwi. Program komputerowy, który realizuje te zadania opisany jest w punkcie 3.5.



Rysunek 3.1. Widok przyrządu i nadgarstka, 1 — czujnik pneumatyczny, 2 — moduł zasilający, 3 — komputer, 4 — przewód pneumatyczny.

Pomiar przeprowadza się w następujący sposób. Czujnik 1 przykładą się do ciała pacjenta, w takim miejscu, gdzie bezpośrednio pod skórą przebiega tętnica z wyczuwalnym tętnem (np. nadgarstek). Następnie, upewniwszy się, że czujnik znajduje się dokładnie nad tętnicą, stopniowo dociska się go do tej tętnicy. W czasie tego docisku na ekranie komputera pojawia się przebieg fali tętna krwi (rys. 3.2 przebieg 1),



Rysunek 3.2. Widok ekranu komputera, 1 – ciśnienie p_2 , 2 – ciśnienie p_2 po „wyprostowaniu”.

którego amplituda najpierw narasta a później spada, przy narastającej bez przerwy składowej stałej. Właściwa fala tętna krwi odpowiada przebiegowi o maksymalnej amplitudzie, przy czym zanim będzie można ją wyznaczyć należy uzyskany przebieg „wyprostować” czyli wyciąć składową stałą (przebieg 2 na rysunku 3.2). Powyższy algorytm przetwarzania jest realizowany w komputerze na bieżąco (w czasie rzeczywistym). Ostatecznie, na ekranie monitora pojawia się w powiększeniu wybrana fala tętna, która następnie podlega dalszej analizie (rys. 3.1).

Przyrząd jest przeznaczony dla lekarzy pierwszego kontaktu, którzy dzięki niemu będą mogli przeprowadzać badania przesiewowe pod kątem niedomagań układu krążenia. Już sam kształt fali tętna, zwłaszcza w tętnicy szyjnej, może wskazywać na taką dolegliwość jak stwardnienie naczyń krwionośnych, niedomykanie zastawki aorty, zastawkowe zwężenie tętnicy głównej, tętno naprzemienne lub dykrotyczne lub inne [7]. Jeżeli w komputerze byłby zainstalowany specjalistyczny program, wówczas na podstawie przebiegu fali tętna krwi w tętnicy promieniowej można by wyznaczyć falę tętna w aorcie [8] oraz szereg różnych wielkości i współczynników [9], jak współczynnik augmentacji czy iloraz żywotności Buckberga [6] oraz sztywność tętnic [3]. W szpitalu, omawiany przyrząd umożliwi ponadto szybkie i bezpieczne obserwowanie reakcji układu krążenia pacjenta na podawane leki.

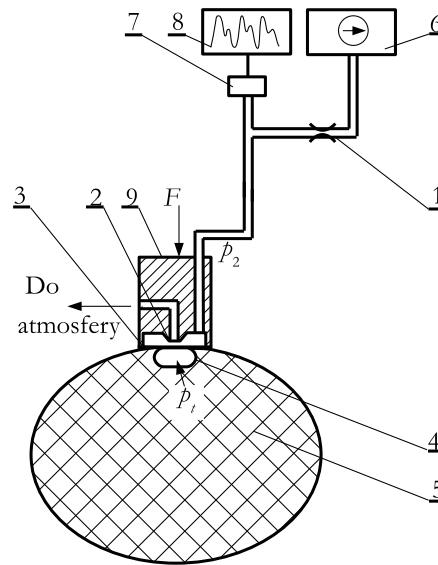
Wymienione wyżej zagadnienia wykraczają poza zakres mojej pracy i nie są przedmiotem dalszych rozważań.

3.2. Budowa i zasada działania czujnika

Konstrukcja czujnika oparta jest na pneumatycznym wzmacniaczu typu dysza-przysłona, który objęty jest ujemnym sprzężeniem zwrotnym. W skład czujnika wchodzi: kapilara 1 (rys. 3.3), dysza 2 i gumowa membrana 3 w roli przysłony.

Podstawą, na której opiera się bezinwazyjny pomiar ciśnienia krwi czujnikiem pneumatycznym, jest zasada kompensacji ciśnień [10]. Zasada ta brzmi następująco: płaska, sprężysta membrana, zamocowana na swym obrzeżu, pozostaje w nie odkształconym (neutralnym) położeniu tylko wtedy, gdy na jedną i drugą stronę działa to samo ciśnienie. Przedstawiona zasada wynika z równania opisującego stan równowagi sił, działających na membranę (rys. 3.4).

$$p_2 A_2 = p_1 A_1 - C y \quad (3.1)$$



Rysunek 3.3. Schemat czujnika i badanej tętnicy, 1 — kapilara, 2 — dysza, 3 — membrana, 4 — tętnica, 5 — nadgarstek, 6 — sprężarka, 7 — przetwornik pneumoelektryczny, 8 — mikroprocesorowy układ przetwarzający z wyświetlaczem, 9 — korpus czujnika.

gdzie:

p_1 — ciśnienie, działające na membranę 3 od dołu, nazywane dalej ciśnieniem mierzonym;

p_2 — ciśnienie, działające na membranę 3 od góry, nazywane dalej ciśnieniem kompensującym;

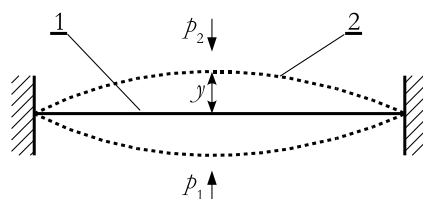
A_1 — czynna powierzchnia membrany od strony działania ciśnienia p_1 ;

A_2 — czynna powierzchnia membrany od strony działania ciśnienia p_2 ;

y — ugięcie środka membrany, liczone od jej neutralnego położenia, ugięcie w górę traktowane jest jako dodatnie;

C — sztywność membrany — siła, wywołująca jednostkowe ugięcie membrany.

Ponieważ w położeniu neutralnym ($y = 0$) czynne powierzchnie membrany są sobie równe ($A_1 = A_2$), to do utrzymania jej w tym położeniu konieczna jest równość ciśnień: $p_2 = p_1$. A więc, jeżeli ciśnienie pod membraną p_1 ma być zmierzone w sposób bezinwazyjny (bez naruszania całości membrany) to nad membranę nale-



Rysunek 3.4. Ilustracja zasady kompensacji ciśnień, 1 — membrana w położeniu neutralnym ($p_1 = p_2$), 2 — membrana odkształcona ($p_1 > p_2$).

ży doprowadzić sprężone powietrze o takim ciśnieniu p_2 aby membrana znalazła się w swoim neutralnym położeniu ($y = 0$). Wówczas ciśnienie p_2 , zmierzone manometrem zewnętrznym, jest równe ciśnieniu mierzonemu p_1 . Zasada ta dotyczy również sytuacji gdy ciśnienie p_1 zmienia się periodycznie.

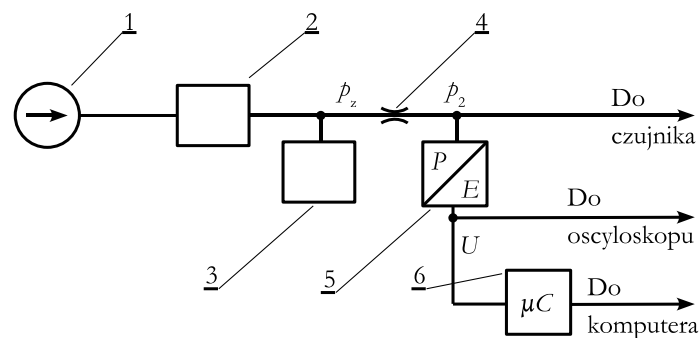
Z opisanej zasady wynika, że aby zmierzyć bezinwazyjnie ciśnienie krwi p_1 , ściana tętnicy na odcinku pomiarowym winna być najpierw spłaszczona a następnie poddana działaniu sprężonego powietrza, którego ciśnienie będzie utrzymywać spłaszczoną część tętnicy w neutralnym położeniu. Spłaszczenie tętnicy, wykonywane ręcznie za pomocą obudowy czujnika, ma na celu usunięcie naprężeń w badanej tętnicy, dzięki czemu staje się ona w tej części zupełnie wiotka. Natomiast, do utrzymania tętnicy w swoim neutralnym położeniu konieczny jest regulator nadążny, którego sygnał wyjściowy, w postaci ciśnienia kompensującego p_2 , nadąża za zmianami sygnału wiodącego, pod postacią ciśnienia mierzonego p_1 . Rolę tego regulatora w prezentowanym układzie pomiarowym spełnia pneumatyczny czujnik ciśnienia. Jego działanie jest następujące.

Gdy ciśnienie p_1 , jest równe ciśnieniu atmosferycznemu p_0 , powietrze przepływa przez kapilarę 1 (rys. 3.3) oraz komorę pomiarową i przez szczelinę (rzędu setnych milimetra) pomiędzy krawędzią dyszy 2 i membraną 3 bez przeszkód uchodzi do atmosfery. Ciśnienie jakie panuje w tej sytuacji w komorze pomiarowej i na wyjściu czujnika, jest zbliżone do ciśnienia atmosferycznego. Natomiast, gdy ciśnienie mierzone p_1 wzrasta powyżej ciśnienia atmosferycznego, wówczas następuje ugięcie

membrany 3 w kierunku dyszy 2 oraz jej przymykanie. W wyniku tego ciśnienie p_2 w komorze pomiarowej wzrasta do takiej wartości, przy której nastąpi równowaga sił działających na membranę z obydwu stron; w praktyce ciśnienie p_2 przyjmuje wartość ciśnienia mierzonego p_1 , również w przypadku zmienności ciśnienia p_1 , a membrana pozostaje w swym neutralnym położeniu. Więcej na ten temat będzie powiedziane przy omawianiu schematu strukturalnego czujnika (rozdział 4).

3.3. Moduł zasilający

Schemat blokowy modułu przedstawiony jest na rysunku 3.5. Do wytworzenia sprężonego powietrza służy sprężarka membranowa 1 zasilana napięciem stałym równym 6 V. Powietrze ze sprężarki podawane jest do reduktora ciśnienia 2 w którym za pomocą śruby można nastawiać żądane ciśnienie zasilania p_z ; dodatkowo reduktor pełni funkcję stabilizatora ciśnienia. Ciśnienie wytwarzane w sprężarce tłokowej jest

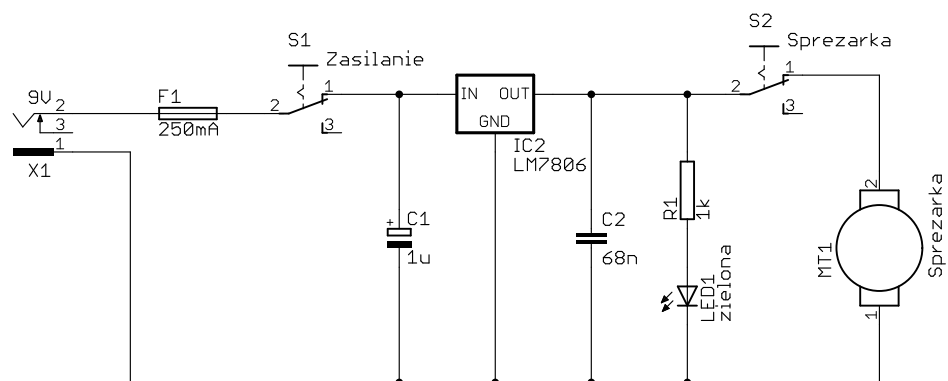


Rysunek 3.5. Schemat blokowy modułu zasilającego, 1 – sprężarka, 2 – stabilizator (reduktor), 3 – tłumik (pojemność pneumatyczna), 4 – dławik, 5 – przetwornik pneumoelektryczny, 6 – mikrokontroler.

ciśnieniem pulsującym więc do zmniejszenia wahań ciśnienia służy wspomniany stabilizator oraz tłumik pneumatyczny 3 w postaci komory. „Wygładzone” ciśnienie p_z zasila czujnik, którego dławik pneumatyczny 4 znajduje się również w module. Ciśnienie p_2 za dławikiem mierzone jest za pomocą przetwornika pneumoelektrycznego.

go 5. Napięcie wyjściowe U z przetwornika 5 (zakres 0–5 V) może być rejestrowane za pomocą oscyloskopu z pamięcią. Napięcie U jest przesyłane również do przetwornika A/C (analogowo-cyfrowy) znajdującego się w mikrokontrolerze 6 gdzie w postaci cyfrowej jest wstępnie obrabiane i wysyłane za pomocą kabla USB do komputera.

Na płycie czołowej modułu widocznej na rysunku 3.1 znajduje się króciec, służący do przyłączania czujnika oraz umieszczone są dwa przełączniki: „Zasilanie” i „Sprężarka”; pierwszy służy do włączenia obwodu zasilania a drugi sprężarki. Poprawność napięcia zasilania wskazuje zielona dioda. Schemat połączeń obwodu zasilającego sprężarkę znajduje się na rysunku 3.6. Obwód zasilany jest napięciem 9 V, które następnie jest redukowane przez stabilizator LM7806 do napięcia 6V. Dodatkowo są dwa kondensatory, tworzące filtr oraz bezpiecznik.



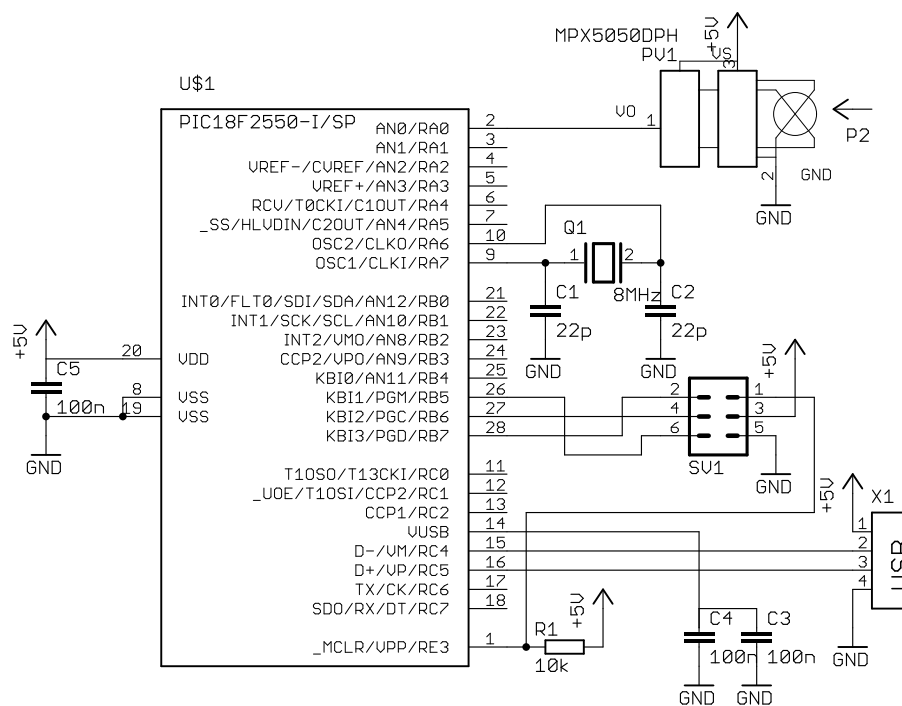
Rysunek 3.6. Schemat zasilania sprężarki.

Na płycie tylnej modułu znajduje się złącze służące do przyłączenia zasilania elektrycznego oraz złącze USB typu B do przyłączenia komputera. W module zasilającym znajduje się również układ pomiaru ciśnienia i komunikacji z komputerem. Opis znajduje się w rozdziale 3.4.

3.4. Interfejs do współpracy czujnika pneumatycznego z komputerem

3.4.1. Część elektroniczna

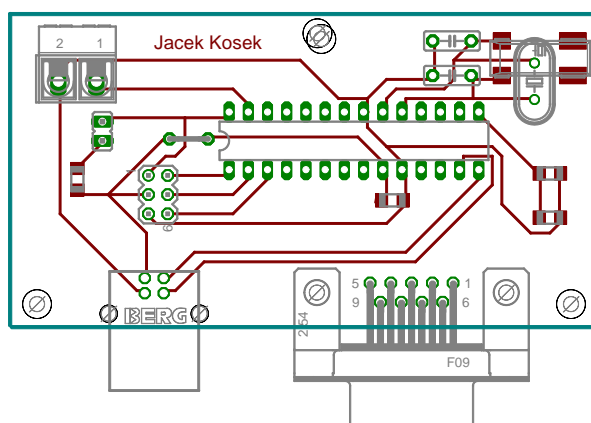
Do pracy wykorzystałem istniejący w Zakładzie Automatyki moduł zasilający, który adoptowałem do nowych potrzeb. Adaptacja ta polegała na zaprojektowaniu, wykonaniu i oprogramowaniu nowego interfejsu, opisanego poniżej.



Rysunek 3.7. Schemat modułu akwizycji i transmisji USB.

Ciśnienie wyjściowe czujnika p_2 mierzone jest za pomocą przetwornika pneumoelektrycznego PV1 (rys. 3.7) typu: Motorola MPX5050DPH. Przetwornik ten przyłączony jest do przetwornika A/C znajdującego się w mikrokontrolerze U\$1 „Microchip” PIC18F2550. Przetwornik oraz mikrokontroler zasilany jest napięciem 5V, dostarczonym za pomocą przewodu USB z komputera.

Na schemacie ideowym (rys. 3.7) oprócz przetwornika PV1 i mikrokontrolera U\$1 znajduje się układ generatora taktującego mikrokontroler, składający się z kondensatorów C1 i C2 oraz rezonatora kwarcowego Q1. Natomiast oprócz gniazda USB X1, dostępnego z zewnątrz znajduje się złącze SV1 do programowania mikrokontrolera U\$1. Płytką drukowaną wraz z rozmieszczeniem elementów znajduje się na rysunku 3.8



Rysunek 3.8. Płytką modułu akwizycji i transmisji USB.

3.4.2. Oprogramowanie mikrokontrolera

Jak już wspomniano mikrokontroler ma do spełnienia kilka zadań:

- przetwarzać napięcie analogowe z przetwornika pneumoelektrycznego 0-5V za pomocą wbudowanego przetwornika A/C,
- obliczać wartość średnią z kilkudziesięciu próbek (w zależności od trybu pracy),
- organizować dane w tak zwane ramki protokołu,
- wysyłać ramki z określoną częstotliwością,
- obsługiwać port USB.

Komunikacja przez port USB jest oparta na sterowniku CDC (Connection Device Class), umożliwia on emulację portu szeregowego w komputerze. Zdecydowano się na tę technologię ze względu na kompatybilność ze wcześniejszymi wersjami urzą-

dzenia, które były przyłączone do portu szeregowego. Natomiast protokół przesyłania danych rozbudowano w stosunku do poprzedniej wersji. Komunikacja opiera się na zasadzie Master-Slave. Komputer jest jednostką nadrzędną, która wydaje rozkazy jednostce podrzędnej, którą w tym wypadku jest mikrokontroler. Wydanie rozkazu polega na przesłaniu odpowiedniego znaku poprzez interfejs szeregowy, lista rozkazów znajduje się w tabeli 3.1.

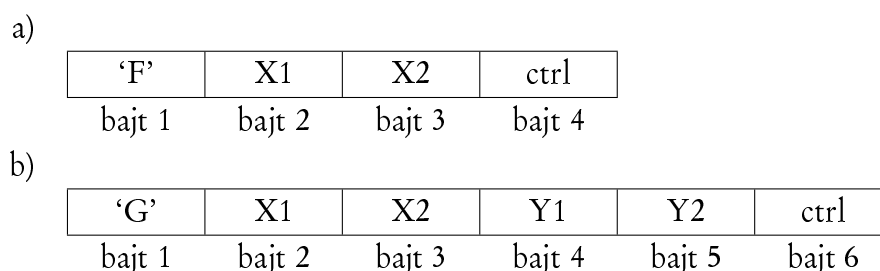
Tabela 3.1. Zestawienie rozkazów wysyłanych przez komputer do modułu.

rozkaz (ASCII)	rozkaz (dziesiętnie)	opis rozkazu
'F'	70	wysyłaj dane odczytane z jednego kanału z częstotliwością 200Hz (domyślny)
'G'	71	wysyłaj dane odczytane z jednego kanału z częstotliwością 100Hz
'H'	72	wysyłaj dane odczytane z dwóch kanałów z częstotliwością 200Hz
'I'	73	wysyłaj dane odczytane z dwóch kanałów z częstotliwością 100Hz
'E'	69	przerwij wysyłanie danych

Rozkaz 'F' jest rozkazem domyślnym, oznacza to, że w momencie zasilenia modułu mikrokontroler przełącza się w tryb wysyłania danych z pierwszego kanału z częstotliwością 200Hz (tak jakby wysyłany był rozkaz 'F').

Dane wysyłane przez mikrokontroler do komputera organizowane są w tak zwane ramki, wykorzystane są przy tym dwie długości ramek: 4 i 6 bajtowe w zależności od tego czy wysyłane są dane z jednego czy dwóch kanałów. Ramki 4 i 6 bajtowe przedstawione są na rysunku 3.9.

Pierwszy bajt rozpoczyna ramkę jeśli jest to litera 'F', oznacza to, że ramka będzie 4 bajtowa, natomiast jeśli będzie to litera 'G', oznacza to, że ramka będzie miała długość 6 bajtów. Następnie wysyłany jest starszy bajt X1 pierwszego kanału a zaraz po nim młodszy bajt X2. Jeśli ramka jest sześć bajtowa wówczas wysyłane są bajty



Rysunek 3.9. Ramki wysyłane z modułu do komputera: a) ramka cztero bajtowa (jeden kanał), b) ramka sześćo bajtowa (dwa kanały).

kanału drugiego, najpierw starszy Y1 potem młodszy Y2. Dwa bajty (X1 i X2 lub Y1 i Y2) tworzą szesnasto bitową liczbę przechowującą wartość napięcia U w miliwoltach (mV), czyli 1 V przechowywany jest jako liczba 1000 czyli $X1 = 3$ a $X2 = 232$. Więcej na ten temat znajduje się w następnym punkcie na stronie 28. Ostatnim bajtem ramki jest bajt sumy kontrolnej, w którym znajduje się suma poprzednich bajtów. Dzięki temu bajtowi jesteśmy w stanie dowiedzieć się czy ramka trafiła bez błędów.

Program realizujący powyższe zadania napisany jest w języku „C”; źródła programu znajdują się w dodatku A. Do uzyskania przebiegu o lepszej jakości na jeden wysłany punkt wykonuje się od 10 do 50 próbek sygnału a następnie wyciąga się z nich wartość średnią, liczba próbek ściśle zależy od możliwości mikroprocesora, ilości kanałów oraz częstotliwości z jaką punkty wysyłane są do komputera. W tabeli 3.2 przedstawiono zestawienie ilości próbek służących do wyznaczenia jednego punktu,

Tabela 3.2. Zestawienie trybu pracy interfejsu z ilością próbek służących do uśrednienia jednego punktu pomiarowego.

rozkaz	ilość próbek
'F'	25
'G'	50
'H'	10 (na pojedynczy kanał)
'I'	20 (na pojedynczy kanał)

w zależności od trybu pracy interfejsu wybranego przez wysłanie odpowiedniego rozkazu.

3.5. Oprogramowanie komputera

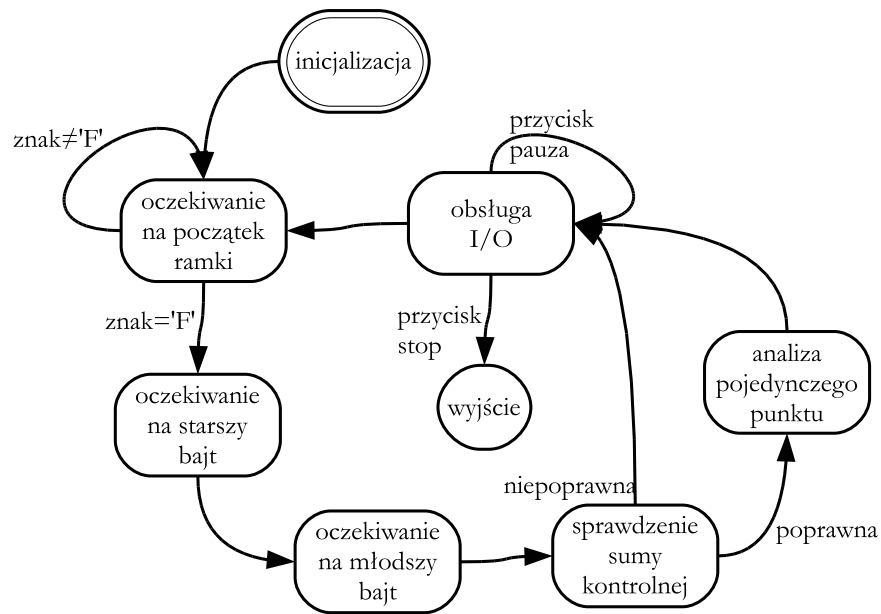
3.5.1. Zadania programu komputerowego

Interfejs, znajdujący się w module zasilającym, wysyła wartości napięcia U , zorganizowane w ramki, poprzez uniwersalną szynę szeregową (USB) do komputera; w tym celu w komputerze musi być zainstalowany sterownik CDC (connection device class). Sterownik ten emuluje port szeregowy w komputerze. W komputerach z systemem operacyjnym Windows 2000/XP/VISTA sterownik ten jest już standardowo zainstalowany; wymagana jest jedynie instalacja pliku: „Pomiar_Cisnienia.inf” (źródła zamieszczono w dodatku). Natomiast w komputerach z systemem Linux wszystko odbywa się automatycznie. Poza sterownikiem w komputerze trzeba uruchomić program rejestrujący i analizujący falę tętna krwi. Program został napisany w matlabie a jego źródła również znajdują się w dodatku.

Program wykonuje zadania w czasie rzeczywistym. Zadania te zostały podzielone na trzy niżej wymienione poziomy:

1. Zadania niskiego poziomu — główna pętla programu.
2. Analiza pojedynczej wartości — wywoływana przez pętlę główną programu po poprawnie odebrany każdym punkcie.
3. Analiza pojedynczej fali tętna krwi — wywoływana po każdorazowym wykryciu narostu ciśnienia.

Analiza przebiegu ciśnienia krwi odbywa się na bieżąco (w czasie rzeczywistym) w pętli głównej po każdorazowym otrzymaniu wartości napięcia wywoływana jest funkcja: „analiza_punktu”, która ma za zadanie wykryć ciśnienie skurczu i wywołać funkcję „analiza_cyklu”.



Rysunek 3.10. Graf przedstawiający zachowanie pętli głównej programu.

3.5.2. Pętla główna programu

Na rysunku 3.10 przedstawiono graf ilustrujący działanie pętli głównej programu. Pętla zaprojektowana jest z użyciem wzorca projektowego maszyny stanu (pętla „for” i instrukcja „switch”). Program zaczyna się od „inicjalizacji”, czyli:

- przygotowanie zmiennych,
- utworzenie i wyświetlenie głównego okna,
- ustawienie parametrów portu szeregowego.

Maszyna stanów taktowana jest strumieniem danych z portu szeregowego. Bezpośrednio po inicjalizacji maszyna wchodzi w stan „oczekiwanie na początek ramki”. W tym stanie pozostaje do momentu otrzymania pierwszego znaku ramki, którym w tym wypadku jest znak ‘F’. W następnych dwóch stanach odczytane są dwa bajty (starszy i młodszy) wartości ciśnienia. Następnym stanem jest: „sprawdzenie sumy kontrolnej”; polega ona na obliczeniu jej wartości w następujący sposób:

$$'F' + X1 + X2 + \text{ctrl} \quad (3.2)$$

gdzie:

- 'F' — pierwszy bajt ramki (początek ramki),
- X1 — drugi bajt ramki (starszy bajt wartości ciśnienia),
- X2 — trzeci bajt ramki (młodszy bajt wartości ciśnienia),
- ctrl — czwarty bajt ramki (suma kontrolna — bajtowe dopełnienie do zera).

Jeśli suma (3.2) jest równa zeru wówczas wywoływana jest funkcja „analiza_punktu”, po której następuje stan „obsługa I/O”, natomiast jeśli suma (3.2) jest różna od zera, stan „analiza_punktu” jest pomijany. Stan „obsługa I/O” wykonuje kilka zadań związanych z interfejsem użytkownika, mianowicie:

- sprawdzanie stanu przycisków,
- odświeżanie ekranu co 50 ms,
- wyczyszczanie okna wykresu w momencie osiągnięcia prawej krawędzi ekranu przez wykres.

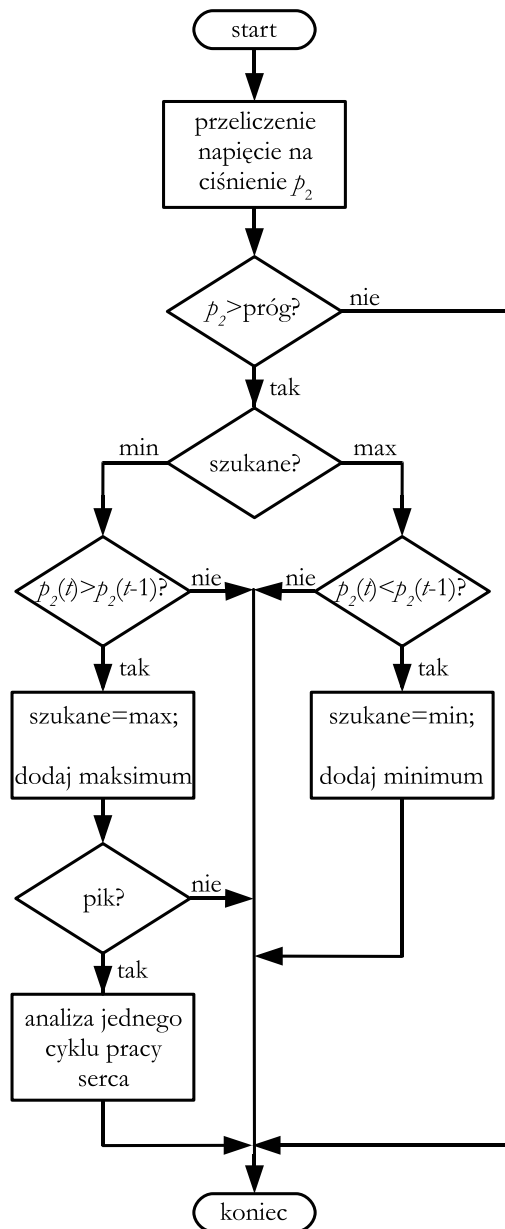
W zależności od stanu przycisków, program kończy pracę (przycisk stop), albo pozostaje w tym stanie (przycisk pauza), albo przechodzi do stanu „oczekiwanie na początek ramki”.

3.5.3. Analiza punktu pomiarowego

Algorytm analizy pojedynczego punktu przedstawiono na rysunku 3.11. Jego podstawowe zadania to:

- przeliczenie napięcia na ciśnienie,
- sprawdzenie czy czujnik jest dociśnięty do skóry pacjenta,
- wykrywanie i rejestrowanie minimów i maksimów lokalnych,
- wykrywanie ciśnienia skurczowego,
- wywoływanie funkcji „analiza_cyklu” gdy zostanie wykryte ciśnienie skurczowe.

Obliczanie ciśnienia składa się z dwóch etapów:



Rysunek 3.11. Algorytm działania funkcji: „analiza_punktu”.

Etap 1. Wyznaczenie wartości napięcia z dwóch bajtów (X1 i X2) za pomocą wzoru:

$$U = 256 \cdot X1 + X2 \quad (3.3)$$

gdzie:

U — wartość napięcia w V.

Etap 2. Przeliczenie napięcia U na wartość ciśnienia p_2 zgodnie ze wzorem:

$$p_2 = 0,0473 \cdot U + 6,352 \quad (3.4)$$

gdzie:

p_2 — ciśnienie bez uwzględnienia charakterystyki czujnika w mmHg (więcej na ten temat w rozdziale 5.3).

Jeśli ciśnienie p_2 jest mniejsze od wartości progowej równej 20 mmHg (2,7 kPa), wówczas program wychodzi z funkcji gdyż czujnik nie został dociśnięty do skóry i program nie rejestruje, nie wyświetla i nie analizuje przebiegu ciśnienia. Natomiast jeśli ciśnienie p_2 jest większe od 20 mmHg (2,7 kPa) rejestruje się je i sprawdza czy obecna wartość jest minimum lub maksimum lokalnym, jeśli tak, to również zapamiętywane jest to ekstremum. W przypadku kiedy jest to maksimum dodatkowo sprawdza się czy jest to maksimum związane z ciśnieniem skurczowym. Sprawdzenie polega na odjęciu od obecnego maksimum funkcji ciśnienia poprzedniego minimum. Jeśli wartość ta przekracza 15 mmHg (2 kPa) wywoływana jest funkcja „analiza_cykladotdotdot”.
za_cyklu”.

3.5.4. Analiza pojedynczej fali tętna

Funkcja „analiza_cyklu” jest wywoływana w momencie wykrycia ciśnienia skurczowego. Funkcja ta wykonuje szereg zadań związanych z analizą tętna krwi a jej działanie można zapisać w następujących krokach.

1. Zapisanie obecnej wartości ciśnienia skurczowego.
2. Wyszukanie minimum funkcji ciśnienia p_2 , pomiędzy obecnym a poprzednim zapisanym ciśnieniem skurczowym (wartość ta jest ciśnieniem rozkurczowym).
3. Zapisanie wartości ciśnienia rozkurczowego.

4. Na podstawie dwóch ostatnio zapisanych punktów ciśnienia rozkurczowego wyznaczane jest równanie (współczynniki: c_0 i c_1) prostej przechodzącej przez te punkty.
5. Obliczenie „wyprostowanej” fali tętna, polega na odjęciu od zarejestrowanej funkcji ciśnienia p_2 , wartości wyliczonych z równania prostej.

$$p_{2w}(t) = p_2(t) - (c_1 \cdot t + c_0) \quad (3.5)$$

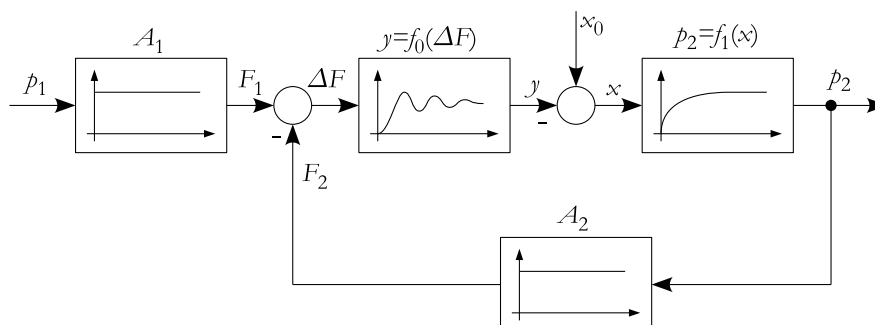
6. Wyświetlenie „wyprostowanej” p_{2w} fali tętna.
7. Sprawdzenie czy obecny przebieg posiada maksimum amplitudy, jeśli tak to odświeża wartość ciśnienia skurczowego i rozkurczowego wyświetlanego na ekranie oraz zaznacza te wartości na wykresie.

Należy zwrócić uwagę że para współczynników c_0 i c_1 jest indywidualna dla każdej pojedynczej fali tętna.

4. Model matematyczny czujnika

4.1. Cel i metoda modelowania

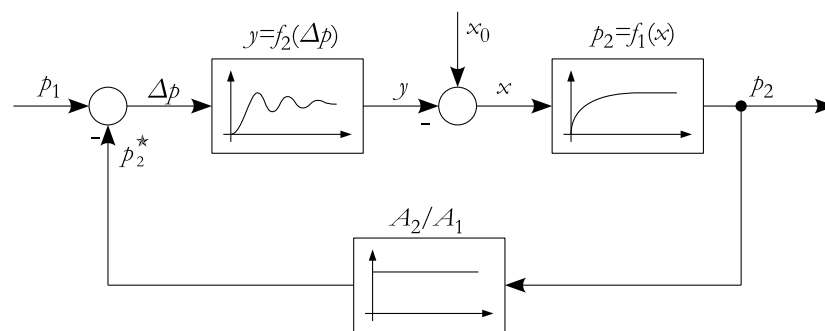
Konstrukcja czujnika była wzorowana na czujniku ciśnienia wewnątrzczaszkowego, opracowanego wcześniej w Zakładzie Automatyki [11]. Moim zadaniem było dobranie takich parametrów konstrukcyjnych i eksploatacyjnych czujnika aby można było nim mierzyć bez zniekształceń zmienne ciśnienie o ponad dziesięciokrotnie wyższych wartościach. W tym celu trzeba było zbudować model matematyczny czujnika, a następnie wyznaczyć z niego najodpowiedniejsze wartości wspomnianych parametrów.



Rysunek 4.1. Schemat strukturalny czujnika do pomiaru ciśnienia krwi ze wzmacniaczem typu dysza-przysłona, schemat wyjściowy.

Wyjściowa struktura czujnika jest przedstawiona na rysunku 4.1. Ciśnienie mierzone p_1 działa na powierzchnię czynną A_1 membrany 3 czujnika (rys 3.3), wywołując siłę F_1 skierowaną do góry. Siłę F_1 przeciwdziałają siła F_2 skierowana ku dołowi, wywołana przez działanie ciśnienia p_2 na powierzchnię czynną membrany A_2 . Różnica

tych sił $\Delta F = F_1 - F_2$ (węzeł sumujący) działając na membranę powoduje jej ugięcie y w stosunku do swojej neutralnej pozycji $y = f_0(\Delta F)$. Jeśli różnica ΔF jest dodatnia membrana wychyla się ku górze (zbliżając się do dyszy), natomiast jeśli różnica sił jest ujemna membrana ugina się ku dołowi zwiększając szczelinę między membraną a dyszą wzmacniacza typu dysza-przysłona. W pozycji neutralnej między membraną a krawędzią dyszy wzmacniacza występuje szczelina x_0 nazywana dalej odsadzeniem. Chcąc obliczyć odległość membrany (stanowiącej przysłonę wzmacniacza) od dyszy, należy odjąć od odsadzenia x_0 wartość ugięcia membrany y , czyli: $x = x_0 - y$; operacja ta jest wykonywana w drugim węźle sumującym. Wielkość x jest sygnałem wejściowym wzmacniacza typu dysza-przysłona. Natomiast sygnałem wyjściowym jest ciśnienie p_2 . Opis zależności $p_2 = f_1(x)$ przedstawiono w punkcie 4.2.



Rysunek 4.2. Schemat strukturalny czujnika do pomiaru ciśnienia krwi ze wzmacniaczem typu dysza-przysłona, schemat po przekształceniu.

Strukturę przedstawioną na rysunku 4.1, przekształcono do struktury widocznej na rysunku 4.2, przesuwając węzeł sumujący przed blok powierzchni czynnej A_1 . Możliwe to było ponieważ bloki powierzchni czynnej są członami liniowymi proporcjonalnymi. Dzięki takiej operacji unika się podczas identyfikacji pomiaru sił, powstających na membranach, co stanowi zwykle duży problem. Znacznie łatwiej mierzyć ciśnienia.

Modelowanie czujnika polegało na identyfikacji jego podstawowych podzespołów: wzmacniacza typu dysza-przysłona i membrany a następnie wyprowadzeniu,

w oparciu o zaproponowany schemat strukturalny (rys. 4.2), zależności pomiędzy sygnałem wyjściowym p_2 i wejściowym p_1 czujnika, zarówno w stanach ustalonych jak i nieustalonych.

4.2. Identyfikacja wzmacniacza typu dysza-przysłona

4.2.1. Opis wzmacniacza typu dysza-przysłona

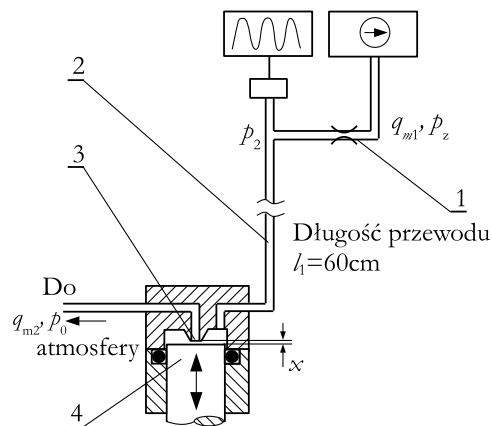
Wzmacniacz ten składa się z, połączonych szeregowo, kapilary 1 (rys 4.3) i dyszy 3, której wylot do atmosfery jest przymykany płaską przysłoną 4. Wzmacniacz jest zasilany sprężonym powietrzem, które, po przejściu kapilary a następnie szczeliny pomiędzy krawędzią dyszy i płaszczyzną przysłony, uchodzi do atmosfery. Charakterystyka statyczna wzmacniacza jest nieliniową zależnością ciśnienia p_2 za kapilarą od otwarcia dyszy x oraz takich wielkości jak ciśnienie zasilania, średnica kapilary i dyszy, długość kapilary i inne, a charakterystyka dynamiczna, dodatkowo zależy od średnicy i długości przewodu łączącego kapilarę z dyszą. Kapilarę 1 wraz z elementem dysza-przysłona 3 można traktować jako dzielnik ciśnienia, przy czym w stanach nieustalonych przewód 2 wnosi do niego niepożądaną inercyjność.

Znajomość modelu wzmacniacza typu dysza-przysłona jest niezbędna przy optymalizacji głównych parametrów konstrukcyjnych czujnika; są to:

1. średnica d_1 i długość l kapilary
2. średnica d_3 linii przesyłowej
3. średnica d_2 dyszy wzmacniacza
4. tzw. odsadzenie x_0

oraz ciśnienie zasilania p_z

Pneumatyczny wzmacniacz typu dysza-przysłona, mimo, że jest jednym z najbardziej rozpowszechnionych podzespołów pneumatyki, do dziś nie ma ogólnego (uniwersalnego) modelu matematycznego, pozwalającego na zaprojektowanie wzmacnia-



Rysunek 4.3. Schemat wzmacniacza typu dysza-przysłona, 1 — kapilara (oporność pneumatyczna), 2 — przewód (oporność i pojemność pneumatyczna), 3 — element dysza-przysłona (oporność pneumatyczna). 4 — trzcina stanowiąca przysłonę.

cza o z góry założonych charakterystykach: statycznej i dynamicznej. Powodem tego są trudności związane z opisem przepływu powietrza, zarówno przez kapilarę jak i dyszę wzmacniacza. W jednym i drugim elemencie może wystąpić, równocześnie lub na przemian, przepływ laminarny ewentualnie burzliwy, pod lub nadkrytyczny, poza tym wzmacniacz pracuje w nietypowy sposób ponieważ powietrze uchodzi do atmosfery wpływając do szczeliny pomiędzy krawędzią dyszy i membraną a nie jak w innych przyrządach — wypływa z niej.

4.2.2. Metoda opracowania modelu

W oparciu o równania mechaniki płynów i termodynamiki, opracowano oddzielnie modele przepływu powietrza przez kapilarę i element dysza-przysłona. Modele te skonfrontowano z charakterystykami doświadczalnymi wymienionych elementów, wyznaczonymi dla trzech różnych wymiarowo kapilar, trzech różnych średnic dysz oraz czterech różnych ciśnień zasilania. W oparciu o te charakterystyki skorygowano wyprowadzone wcześniej równania, które następnie wykorzystano do opisu charakterystyki statycznej całego wzmacniacza typu dysza-przysłona.

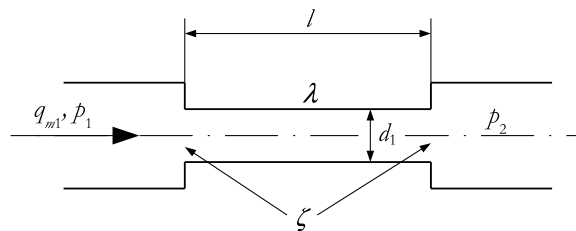
Charakterystykę dynamiczną wzmacniacza typu dysza-przysłona opisano przy

założeniu (mającym potwierdzenie w eksperymencie), że wzmacniacz ten, w rozpatrywanym zakresie częstotliwości przenoszonych sygnałów, jest członem inercyjnym pierwszego rzędu. Model dynamiki wzmacniacza typu dysza-przysłona zweryfikowano w oparciu o jego doświadczalne charakterystyki częstotliwościowe, wyznaczone na specjalnym stanowisku pomiarowym dla trzech różnych kapilar, trzech różnych dysz, czterech różnych ciśnień zasilania oraz czterech różnych średnic przewodu pneumatycznego, łączącego kapilarę z dyszą wzmacniacza.

Opracowanie modelu wzmacniacza typu dysza-przysłona podzielono na kilka etapów: modelowanie kapilary i modelowanie elementu typu dysza przysłona a następnie modelowanie własności statycznych i dynamicznych całego wzmacniacza. Każdy z tych etapów weryfikowany był eksperymentalnie.

4.2.3. Model kapilary

Pierwszym elementem wzmacniacza jest kapilara, stanowi ona oporność pneumatyczną. Mała średnica przepływowa kapilary powoduje duże straty liniowe na długości tej kapilary przez co podczas przepływu następuje spadek ciśnienia. Ponadto na końcach kapilary występują straty miejscowe. Schemat kapilary przedstawiono na rys. 4.4.



Rysunek 4.4. Schemat kapilary, ζ - straty miejscowe, λ - straty liniowe, l - długość, d_1 - średnica.

Najpierw wykonano doświadczalne charakterystyki przepływowe czterech różnych kapilar; wszystkie kapilary miały tę samą średnicę $d_1 = 0,22$ mm i następujące

długości l : 5, 10, 20, 35 mm. Warunki prowadzenia eksperymentu podyktowane były rzeczywistymi warunkami w jakich pracuje kapilara podczas pomiaru ciśnienia krwi. Eksperyment polegał na wyznaczeniu strumienia masy q_{m1} przepływającego przez kapilarę przy zadanym stałym spadku ciśnienia Δp . Otrzymane charakterystyki (rys. 4.5), zwłaszcza krótkich kapilar, wykazują pewną nieliniowość, w związku z tym odpada liniowy model Hagen-Poiseuille'a. Zamiast niego posłużono się równaniem do obliczania strumieni w rurociągach [12], uwzględniając zarówno straty liniowe λ , miejscowe ζ i straty związane ze zmianą gęstości strumienia:

$$q_{m1} = \frac{\pi d_1^2}{4} \sqrt{\frac{p_1^2 - p_2^2}{RT [\lambda(l/d_1) + \zeta + 2 \ln(p_1/p_2)]}} \quad (4.1)$$

$$\lambda = \frac{64}{Re}, \quad Re = \frac{4q_{m1}}{\pi d_1 \mu}$$

gdzie:

R — indywidualna stała gazowa,

T — temperatura absolutna,

λ — współczynnik strat liniowych,

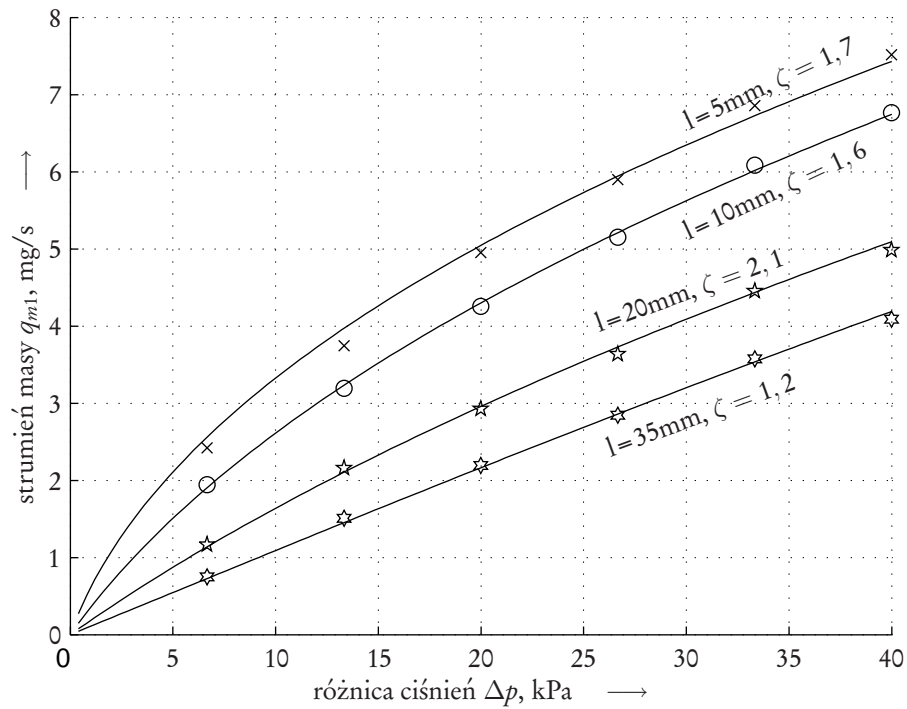
ζ — współczynnik strat miejscowych (na wylocie i wlocie),

Re — liczba Reynoldsa w kapilarze,

μ — współczynnik lepkości dynamicznej powietrza,

q_{m1} — strumień masy przepływający przez kapilarę.

Równanie to posiada uwikłaną zmienną q_{m1} oraz nieznaną wartość współczynnika strat miejscowych ζ . Współczynnik ten wyznaczony został metodą najmniejszych kwadratów opierając się na danych pomiarowych. Porównanie charakterystyk kapilar o czterech różnych długościach znajduje się na rys. 4.5. Liniami ciągłymi zaznaczono charakterystyki obliczeniowe, a punktami wyniki doświadczalne.



Rysunek 4.5. Porównanie modelu matematycznego kapilary z wynikami doświadczeń.

4.2.4. Model elementu dysza-przysłona

Kolejnym elementem wzmacniacza typu dysza-przysłona jest element dysza-przysłona (rys. 4.6), jego podstawowe parametry geometryczne to: średnica dyszy d_2 i wielkość szczeliny x między dyszą a przysłoną. Strumień masy powietrza, przepływający przez ten element dany jest równaniem:

$$q_{m2} = \alpha \pi d_2 x \sqrt{\frac{2p_2}{RT} (p_2 - p_0)} \quad (4.2)$$

gdzie:

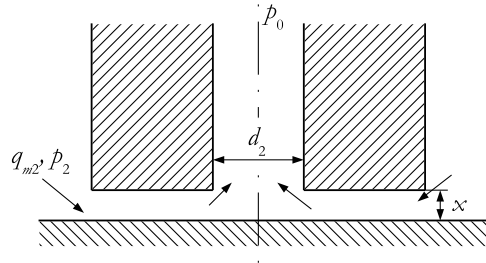
R — indywidualna stała gazowa,

T — temperatura absolutna,

α — współczynnik przepływu,

p_2 — ciśnienie na dolocie do elementu,

p_0 — ciśnienie atmosferyczne.



Rysunek 4.6. Schemat elementu dysza-przysłona z zaznaczonym kierunkiem przepływu powietrza.

Głównym problemem w tym równaniu jest dobór współczynnika przepływu α . Niestety wartości tego współczynnika występują w literaturze tylko dla wzmacniacza pracującego w warunkach znacznie odbiegających od występujących w czujniku ciśnienia krwi. W związku z tym konieczne było jego doświadczalne wyznaczenie na podstawie bilansu masy:

$$q_{m1} = q_{m2} \quad (4.3)$$

gdzie:

q_{m1} — strumień wyznaczony z równania kapilary (4.1),

q_{m2} — strumień opisywany równaniem (4.2).

Jedyną niewiadomą w równaniu (4.2) jest współczynnik α . Po wykonaniu serii obliczeń dla różnych: średnic dysz d_2 (0,6 mm, 0,8 mm), ciśnień zasilania p_z (24 kPa, 30 kPa, 40 kPa, 50 kPa), długości dławika l (5 mm, 10 mm, 20 mm), zauważono, że α zależy od dwóch kryteriów: od kryterium podobieństwa przepływu Re oraz od kryterium podobieństwa geometrycznego czyli od stosunku x/d_2 . Współczynnik α aproksymowano równaniem powierzchni:

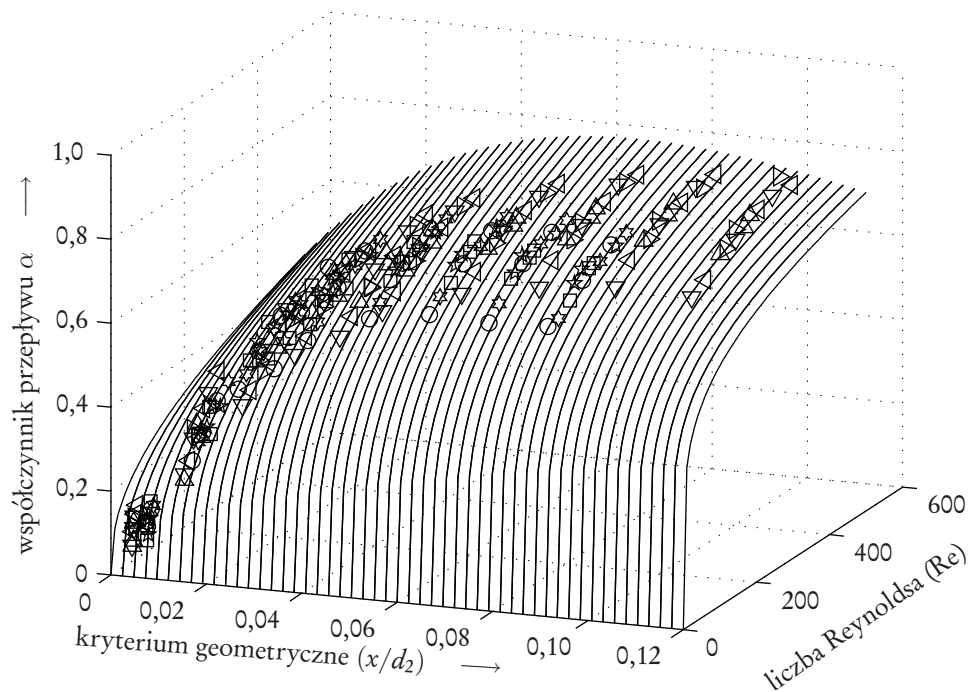
$$\hat{\alpha} = [a_0(x/d_2) + a_1] Re^{a_2(x/d_2)+a_3} \quad (4.4)$$

gdzie:

$a_{0...3}$ — stałe współczynniki

$$Re = \frac{2q_{m2}}{d_2\pi\mu}$$

Współczynniki $a_{0...3}$ dobrano metodą najmniejszych kwadratów tak aby kwadrat różnicy między powierzchnią $\hat{\alpha}$ a wynikami obliczeń tego współczynnika α był minimalny. Wykres zależności α (zaznaczony punktami) i jego aproksymacji $\hat{\alpha}$ (zaznaczony liniami) od liczby Re i x/d_2 znajduje się na rys. 4.7. Współczynniki stałe wynoszą: $a_0 = 1,75$; $a_1 = 0,044$; $a_2 = -1,66$; $a_3 = 0,38$.



Rysunek 4.7. Wykres zależności współczynnika α i jego aproksymacji $\hat{\alpha}$ od Re i x/d_2 .

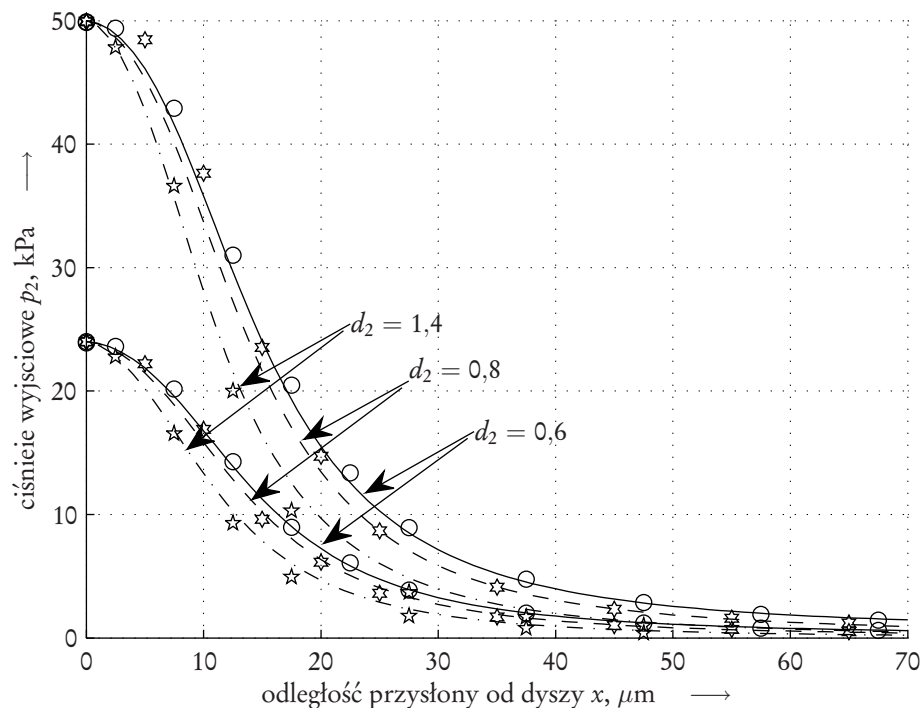
4.2.5. Model własności statycznych wzmacniacza typu dysza-przysłona

Znając model kapilary dany wzorem (4.1) oraz model elementu dysza-przysłona dany wzorem (4.2), gdzie współczynnik α aproksymowany został powierzchnią $\hat{\alpha}$,

możliwe jest uzyskanie, przy pomocy równania bilansu masy (4.3), opisu właściwości statycznych całego wzmacniacza.

Do celów projektowych najistotniejsza jest zależność ciśnienia p_2 od szczeliny między dyszą a przysłoną x (x – wielkość na wejściu, p_2 – wielkość na wyjściu) oraz możliwość zmian tych zależności za pomocą parametrów geometrycznych. Niestety, niemożliwe jest podanie analitycznej zależności p_2 od x z wielu przyczyn, jedną z nich jest to, że wzory (4.1) i (4.2) mają wiele uwikłanych parametrów np: x , Re czy p_2 . Dlatego więc model jest obliczany numerycznie; wzory (4.1) i (4.2) obliczane są w pętli iteracyjnej, natomiast p_2 jest obliczane z bilansu masy (4.3) za pomocą algorytmów optymalizacyjnych (szukanie miejsca zerowego dla różnicy $q_{m1} - q_{m2}$).

Charakterystyki doświadczalne wzmacniacza zostały wykonane przy użyciu krót-



Rysunek 4.8. Wykres zależności ciśnienia p_2 od otwarcia dyszy x dla dwóch ciśnień zasilania p_z : 24 kPa, 50 kPa i dławika o długości $l = 20$ mm

kiego przewodu łączącego element dysza-przysłona z kapilarą. Przykładowe wykresy zależności ciśnienia p_2 od x dla dwóch różnych ciśnień zasilania i trzech różnych średnic dysz d_2 przedstawione są na rys. 4.8. Różnymi rodzajami punktów zaznaczone są charakterystyki doświadczalne natomiast liniami charakterystyki modeli.

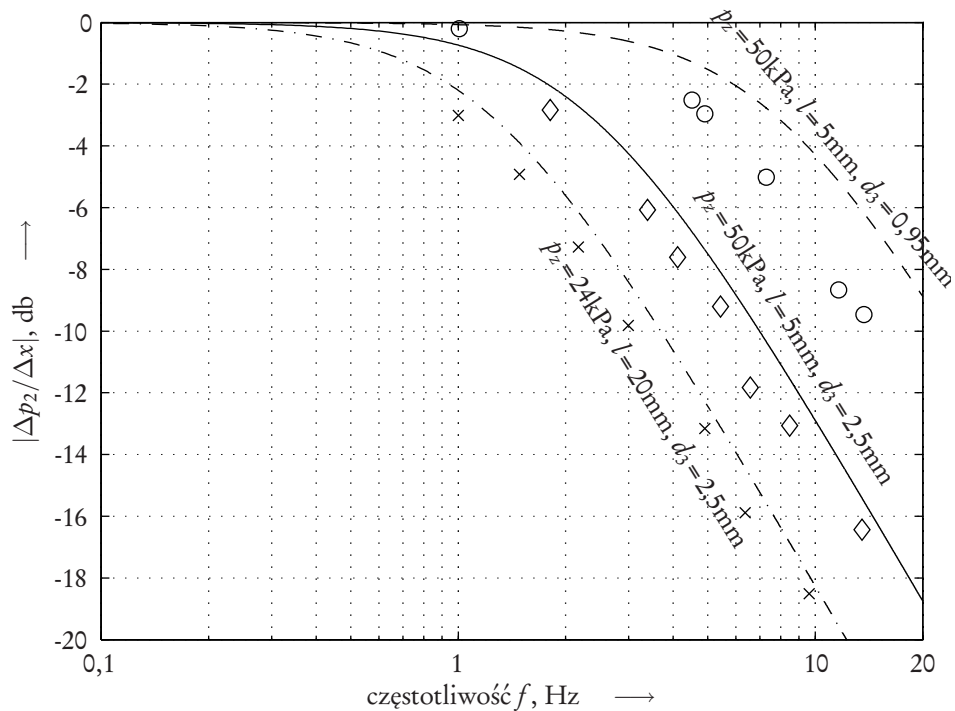
Charakterystyki dyszy o średnicy $d_2 = 1,4$ mm pełnią rolę sprawdzającą, ponieważ wzmacniacz z tą dyszą nie był wykorzystywany przy tworzeniu modelu, a użyto go jedynie w celach sprawdzających dopiero po utworzeniu modelu.

4.2.6. Model własności dynamicznych wzmacniacza typu dysza-przysłona

Kolejnym etapem jest utworzenie modelu dynamicznych właściwości wzmacniacza. Tutaj istotną rolę odgrywa dławik i przewód o długości 60 cm, łączący dławik z elementem dysza-przysłona. Wiadomo, że im krótszy przewód tym lepsze własności dynamiczne, niestety element dysza-przysłona znajduje się w czujniku natomiast kapilara wraz z przetwornikiem pneumoelektrycznym znajduje się w module zasilającym. Uznano że długość 60 cm to minimalna długość, umożliwiającą swobodne posługiwanie się czujnikiem w czasie pomiaru ciśnienia krwi.

Modelowanie rozpoczęto od wykonania doświadczalnych charakterystyk amplitudowo-częstotliwościowych wzmacniacza typu dysza-przysłona na specjalnie do tego zbudowanym stanowisku [13]. Głównym podzespołem tego stanowiska jest generator sinusoidalnych przemieszczeń liniowych, o regulowanej częstotliwości f , amplitudzie i wartości średniej. Na rys. 4.3 pokazano trzpień 4 generatora, który pełni rolę sinusoidalnie przemieszczającej się przysłony wzmacniacza typu dysza-przysłona. Charakterystyki wykonano przy czterech różnych ciśnieniach zasilania (24, 30, 40, 50 kPa) oraz dla trzech różnych kapilar o długościach l : 5, 10, 20 mm i średnicy d_1 : 0,22 mm a także dla dwóch średnic dysz d_2 : 0,6 i 0,8 mm oraz dla czterech średnic d_3 (2,5; 1,85; 1,5; 0,95 mm) przewodu zasilającego. Przykładowe charakterystyki częstotliwościowe przedstawiono na rysunku 4.9.

Przy tworzeniu opisu własności dynamicznych wzmacniacza typu dysza-przysłona



Rysunek 4.9. Przykładowe doświadczalne charakterystyki częstotliwościowe (wzmocnienie wzmacniacza sprowadzono do jedynki).

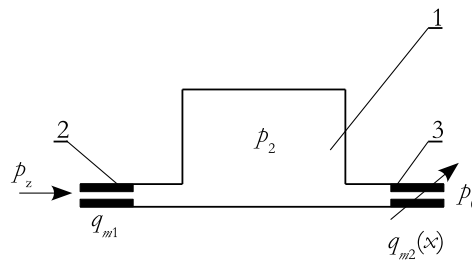
na przyjęto iż stanowi on komorę przepływową 1 (rys. 4.10) z dwoma oporami pneumatycznymi: wlotowym 2 i wylotowym 3, przy czym, opór 2 reprezentuje kapilarę, komora 1 — przewód zasilający, a opór 3 — element dysza-przysłona.

Równanie bilansu strumieni masy, przedstawionego na rysunku 4.10 członu, ma postać [14]:

$$\frac{V}{RT} \frac{dp_2}{dt} = q_1(p_2) - q_2(p_2, x) \quad (4.5)$$

gdzie:

- $q_1(p_2)$ — strumień dopływający,
- $q_2(p_2, x)$ — strumień odpływający,
- p_2 — ciśnienie za dławikiem, będące sygnałem wyjściowym wzmacniacza,
- p_z — ciśnienie zasilania,
- x — otwarcie dyszy, będące sygnałem wejściowym wzmacniacza,



Rysunek 4.10. Uproszczony schemat modelu dynamiki wzmacniacza typu dysza-przysłona.

R — uniwersalna stała gazowa,

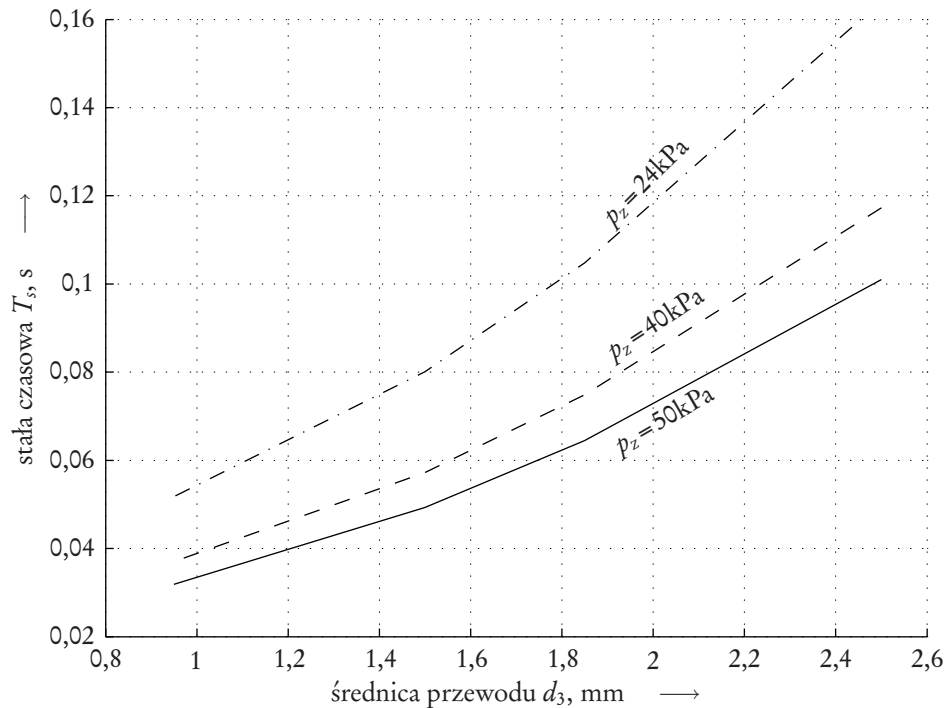
T — temperatura absolutna,

V — objętość przewodu zasilającego wraz z objętością instalacji pomiarowej.

Równanie (4.5) pozwala wyznaczyć charakterystyki amplitudowo-częstotliwościowe wzmacniacza. W tym celu na wejście modelu wprowadzany jest sygnał sinusoidalny x o wartości średniej i amplitudzie takiej jak przy wyznaczaniu charakterystyk doświadczalnych, a następnie na badaniu amplitudy sygnału wyjściowego p_2 .

Na rysunku 4.9 przedstawiono 3 przykładowe charakterystyki amplitudowo-częstotliwościowe obliczone w oparciu o równanie (4.5), a na ich tle naniesiono odpowiadające im punkty pomiarowe. Konfrontując te charakterystyki z punktami pomiarowymi, widać pewne różnice w ich przebiegu. Niższe częstotliwości przenoszenia charakterystyk doświadczalnych wynikają z dodatkowych pojemności i oporności pneumatycznych występujących w instalacji pomiarowej: przewody pomocnicze, trójniki, tulejki do łączenia przewodów o różnych średnicach, króćce przetworników i tym podobne.

Na rysunku 4.11 przedstawiono stałe czasowe wzmacniacza typu dysza-przysłona dla dławika o długości $l = 20$ mm i średnicy dyszy $d_2 = 0,6$ mm dla różnych średnic przewodu zasilającego oraz różnych ciśnień zasilania p_z . Jak widać, zmniejszając średnicę przewodu zasilającego z $d_3 = 2$ mm na $d_3 = 1$ mm stała czasowa T zmniejsza się dwukrotnie [15].



Rysunek 4.11. Zależność stałych czasowych wzmacniacza od średnicy przewodu zasilającego i ciśnienia zasilania dla dyszy $d_2 = 0,6$ mm i dławika o długości $l = 20$ mm.

4.3. Identyfikacja membrany

Model matematyczny membrany opisuje zależność ugięcia środka membrany y od różnicy ciśnień Δp . Model został utworzony w oparciu o badania eksperymentalne na stanowisku pomiarowym. Wykonane zostały dwa eksperymenty: jeden badający właściwości statyczne oraz drugi badający właściwości dynamiczne. Na podstawie tych dwóch eksperymentów zdecydowano się na opisanie membrany równaniem członu oscylacyjnego o nieliniowej charakterystyce statycznej, gdyż podczas badania charakterystyki statycznej okazało się, że ugięcie membrany w zależności od przyłożonej różnicy ciśnień jest nieliniowe, a podczas badania własności dynamicznych okazało się że membrana zachowuje się jak człon oscylacyjny.

$$T_s^2 \frac{d^2 y}{dt^2} + 2\xi T_s \frac{dy}{dt} + y = k(y) \Delta p \quad (4.6)$$

gdzie:

T_s — stała czasowa,

ξ — współczynnik tłumienia,

$k(y)$ — współczynnik wzmocnienia (sztywność ciśnieniowa membrany, zależna od jej ugięcia),

Δp — różnica ciśnień, działająca na membranę (sygnał wejściowy),

y — ugięcie membrany (sygnał wyjściowy).

Identyfikacja polegała na wyznaczeniu charakterystyki statycznej oraz parametrów: T_s i ξ .

4.3.1. Charakterystyka statyczna

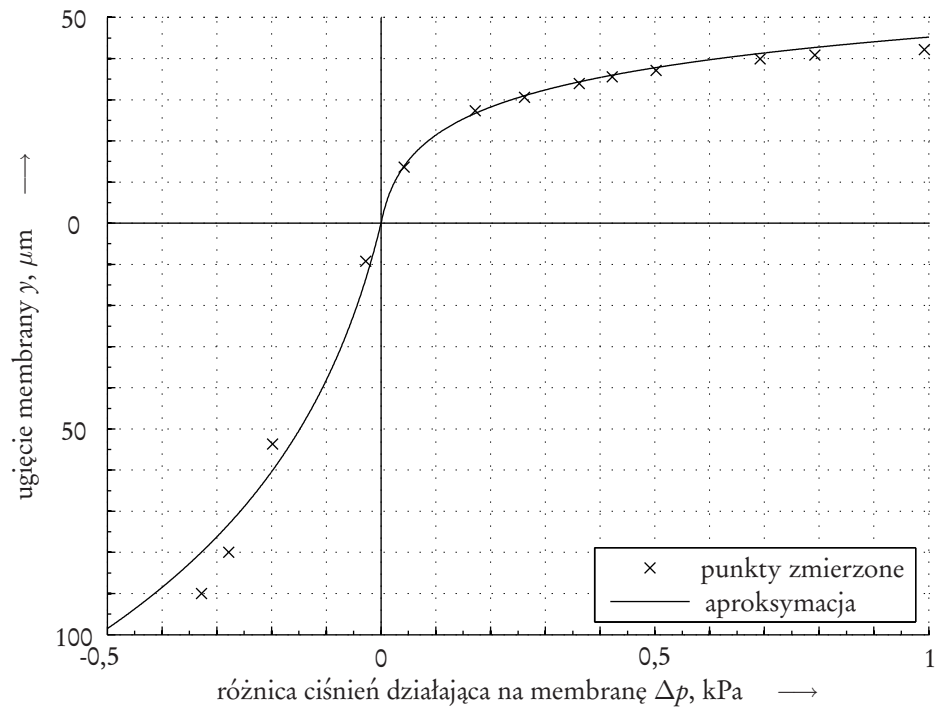
Zależność ugięcia membrany y od różnicy ciśnień Δp wyznaczono za pomocą badanego czujnika ciśnienia, pracującego w układzie otwartym: zasilanie czujnika odbywało się dyszą 2 (rys. 3.3), a przestrzeń nad membraną była połączona z atmosferą. Wcześniej w takim układzie wyznaczono charakterystykę statyczną wzmacniacza: $p_2 = f(x)$, z której w czasie eksperymentu odczytywano ugięcie membrany. Wyznaczoną w ten sposób charakterystykę statyczną (rys. 4.12) aproksymowano równaniem:

$$y = \begin{cases} 11 \cdot 10^{-6} \lg(0,06\Delta p + 1) & \text{dla } \Delta p \leq 0 \\ -55 \cdot 10^{-6} \lg(-0,01\Delta p + 1) & \text{dla } \Delta p > 0 \end{cases} \quad (4.7)$$

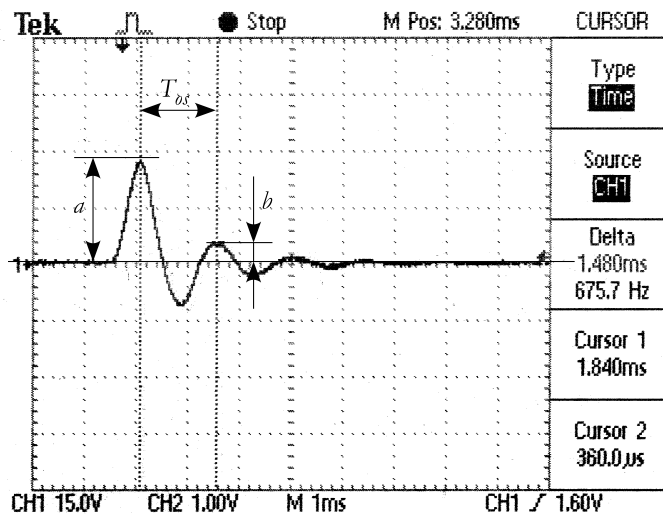
Wielkości y i Δp są wyrażone w jednostkach układu SI (m i Pa). Jak widać charakterystyka jest nieliniowa i niesymetryczna. Dla $\Delta p > 0$ membrana wychyla się ku dyszy (dodatnia wartość ugięcia y) natomiast dla $\Delta p < 0$ membrana wychyla się ku dołowi wybrzusząc się.

4.3.2. Charakterystyka impulsowa

Własności dynamiczne membrany określono za pomocą jej doświadczalnej charakterystyki impulsowej. Charakterystykę tę uzyskano metodą rzutu kulki na mem-



Rysunek 4.12. Charakterystyka statyczna membrany.



Rysunek 4.13. Charakterystyka impulsowa membrany.

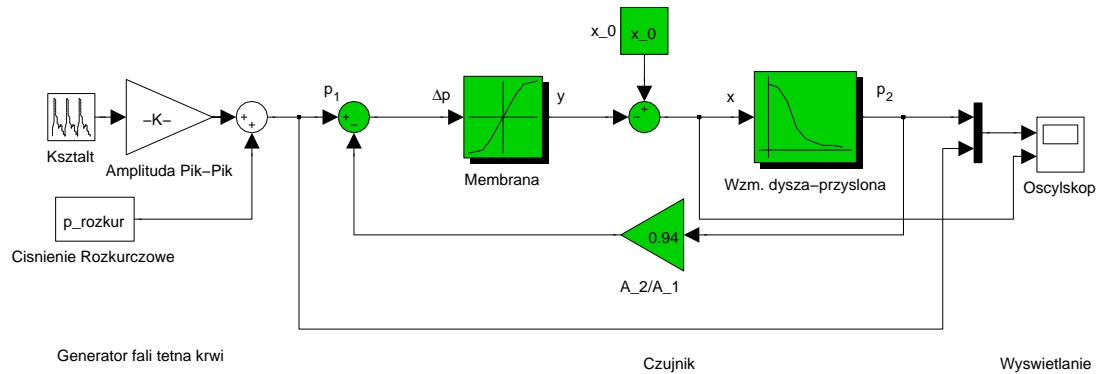
branę, zamykającą komorę pomiarową pojemnościowego przetwornika pneumoelektrycznego. Z charakterystyki tej (rys. 4.13) odczytano okres oscylacji T_{os} oraz wartości a i b . Wielkości te, pozwoliły obliczyć wartości współczynników, występujących w równaniu (4.6) [14]:

$$T_s = 0,23 \text{ ms}, \quad \xi = 0,2.$$

Wartościom T_s i ξ odpowiada częstotliwość drgań własnych nietłumionych membrany: $f = 690 \text{ Hz}$.

4.4. Model czujnika i jego weryfikacja

Model czujnika opracowano w oparciu o schemat strukturalny przedstawiony na rysunku 4.2. Pierwszy człon (od strony ciśnienia mierzonego p_1) opisuje zależność ugięcia membrany y od różnicy ciśnień Δp i dany jest równaniem (4.6). Dalej znajduje się węzeł sumujący, dzięki któremu przy pomocy odsadzenia x_0 (odległość membrany w położeniu neutralnym od dyszy) obliczana jest wartość x (otwarcie dyszy). Wartość x służy do obliczenia, za pomocą wzoru (4.5), ciśnienia wyjściowego p_2 wzmacniacza typu dysza-przystona. Trzecim członem jest człon proporcjonalny, jest to stosunek czynnych powierzchni (A_2/A_1) od strony ciśnienia p_2 i p_1 . Stosunek ten jest nieco mniejszy od jedności w tym przypadku $A_2/A_1 = 0,94$. Całość zamknięta jest ujemnym sprzężeniem zwrotnym dzięki czemu model przypomina układ regulacji nadążnej gdzie ciśnienie wyjściowe p_2 nadąża za zmianami ciśnienia mierzonego p_1 . Model numeryczny utworzony został za pomocą narzędzia „Matlab Simulink” a jego schemat przedstawiono na rysunku 4.14. Na tym schemacie można wyróżnić część generatora sygnału, modelu czujnika oraz wyświetlania. W generatorze możemy zadać kształt przebiegu oraz jego parametry takie jak amplituda czy składowa stała. Do wyświetlania służy blok oscyloskopu, na którym wyświetlane są na jednym wykresie ciśnienie z generatora p_1 oraz ciśnienie wyjściowe p_2 . Na drugim wykresie wyświetlana jest wartość x (otwarcie dyszy). Modele wzmacniacza typu dysza-przy-



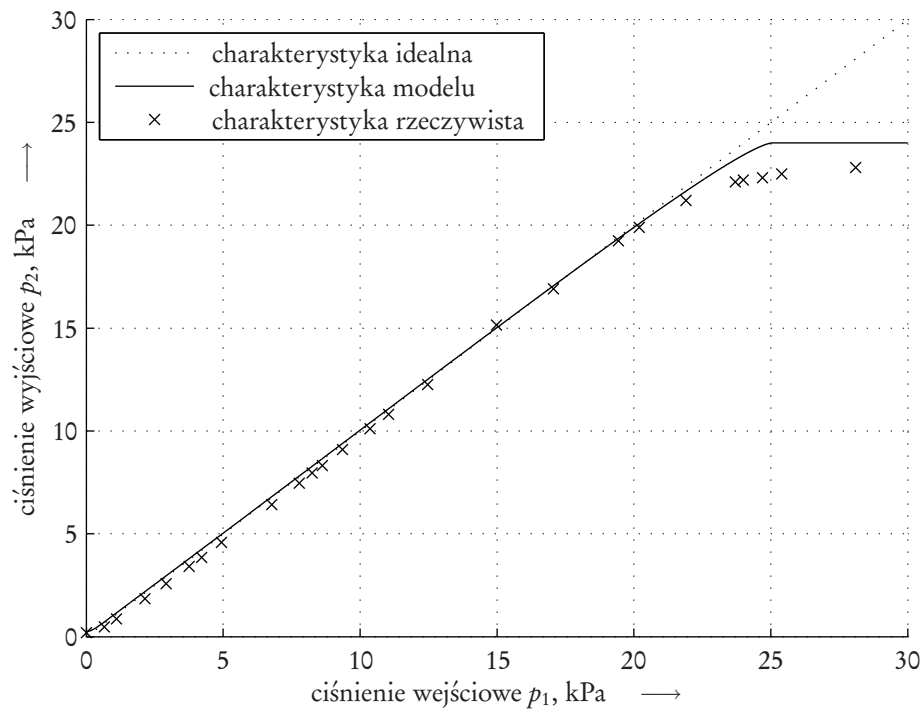
Rysunek 4.14. Schemat modelu czujnika w Simulinku.

słona i membrany zostały wprowadzone jako S-funkcje. Źródła tych plików znajdują się w dodatku C.

4.4.1. Charakterystyka statyczna

Charakterystykę statyczną określono przy założeniu zerowania się pochodnych po czasie w równaniu (4.6), $d^2y/dt^2 = 0$ i $dy/dt = 0$, natomiast w równaniu (4.5) $dp_2/dt = 0$. Na rysunku 4.15 znajduje się przykładowe porównanie charakterystyki czujnika idealnego, charakterystyki doświadczalnej oraz charakterystyki uzyskanej za pomocą modelu numerycznego. Badania doświadczalne wykonano na stanowisku pomiarowym. Membrana czujnika znajdowała się w szczelnej komorze ciśnieniowej z możliwością zadawania ciśnienia p_1 , natomiast ciśnienie wyjściowe czujnika p_2 wskazywał manometr. Natomiast badania numeryczne wykonano ustawiając długi czas symulacji (ok. 1h) w stosunku do stałej czasowej czujnika (ok. 0,1s) oraz sygnał wejściowy liniowo narastający 0,01 kPa/s co zapewniało zerowanie się pochodnych po czasie.

Niepewność modelu statycznego obliczono jako odchylenie standardowe między przebiegiem charakterystyki modelowej i doświadczalnej, wynosi ona $\sigma = 0,3$ kPa.



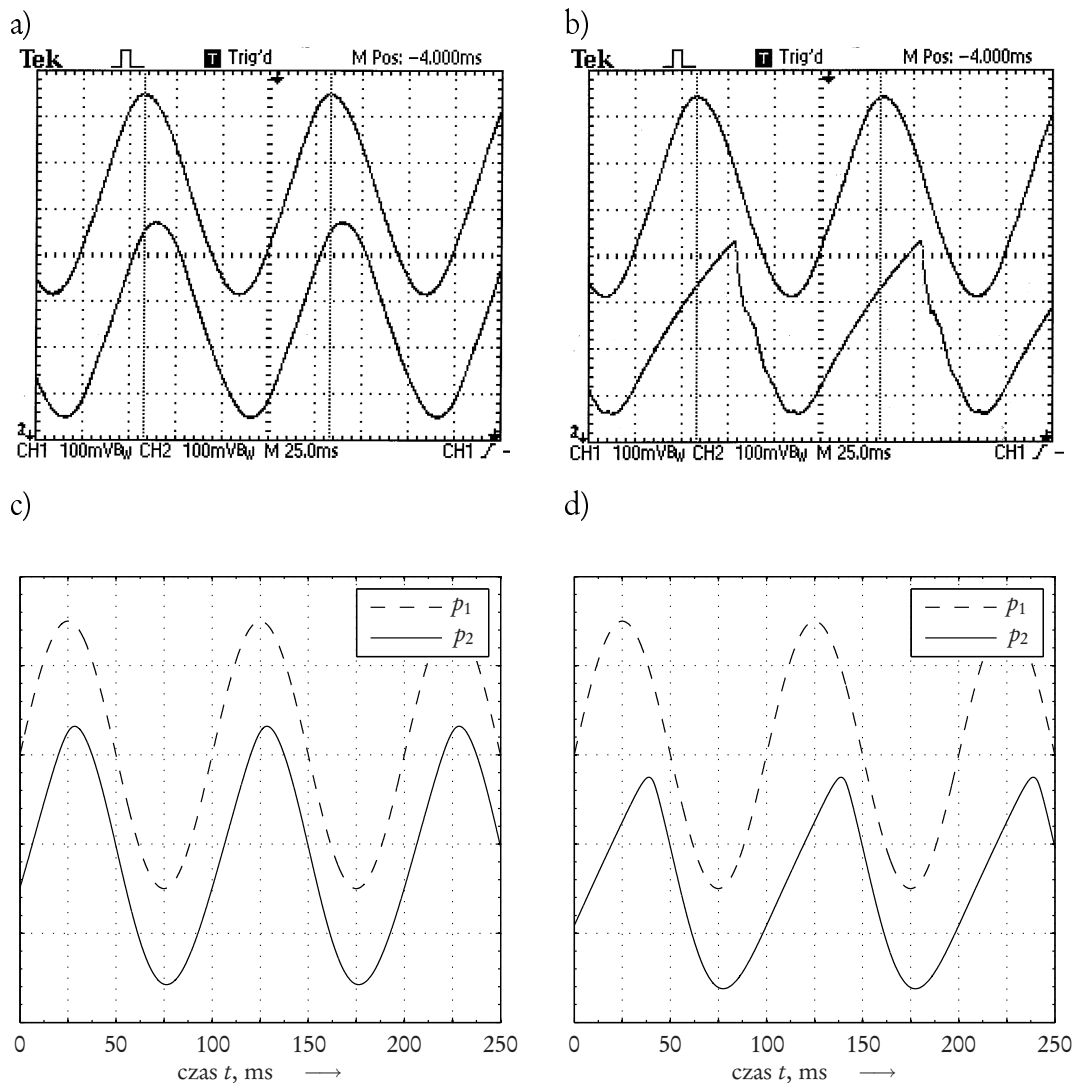
Rysunek 4.15. Charakterystyka statyczna czujnika.

4.4.2. Charakterystyka dynamiczna

Model czujnika jest nieliniowy więc badania charakterystyki dynamicznej zawężono do warunków w jakich będzie pracował czujnik. Badania doświadczalne wykonano na tym samym stanowisku pomiarowym co charakterystyki statyczne. Usunięto jedynie zbędne przewody pneumatyczne do manometru, ponieważ wprowadzały dodatkową pojemność pneumatyczną. Jako zadajnik zmiennego sygnału pneumatycznego posłużył generator pneumatyczny opisany w [16].

Przykładowe porównanie charakterystyk dynamicznych — doświadczalnych i obliczeniowych dwóch czujników o różnych parametrach znajduje się na rysunkach 4.16a, b, c i d.

Na rysunkach 4.16a i b przedstawiono doświadczalne przebiegi ciśnień p_1 i p_2 o częstotliwości 10 Hz czujników nr 1 i nr 2, a na rysunkach 4.16c i d — przebiegi tych



Rysunek 4.16. Przykładowe przebiegi sygnałów mierzonych p_1 (górne przebiegi) i wyjściowych p_2 (dolne przebiegi) czujników: a i b – wyznaczone doświadczalnie, c i d – obliczone w oparciu o model.

ciśnięć uzyskane w oparciu o modele tych samych czujników: nr 1 i nr 2. Czujnik nr 1 miał kapilarę o średnicy $d_1 = 0,22$ mm i długości $l = 5$ mm, przewód zasilający o średnicy $d_3 = 0,95$ mm i ciśnienie zasilania $p_z = 50$ kPa, natomiast czujnik nr 2 miał $d_1 = 0,22$ mm, $l = 20$ mm, $d_3 = 2,5$ mm i $p_z = 24$ kPa. Pozostałe parametry, charakteryzujące czujniki były te same.

Z porównania przebiegów na rysunkach 4.16a i c, uzyskanych przy badaniu czujnika nr 1, wynikają dwa wnioski: po pierwsze, czujnik poprawnie odtwarza mierzony przebieg ciśnienia p_1 , po drugie przebieg ciśnienia p_2 , obliczony w oparciu o model czujnika poprawnie odtwarza przebieg ciśnienia zmierzonego p_2 (z rys.4.16a).

Natomiast z porównania przebiegów uzyskanych przy badaniu czujnika nr 2 (rys. 4.16b i d) wynika, po pierwsze, czujnik niepoprawnie odtwarza mierzony przebieg ciśnienia p_1 , po drugie, przebieg ciśnienia p_2 , obliczony w oparciu o model czujnika, poprawnie odtwarza sygnał doświadczalny p_2 (powtarza zniekształcenie tego sygnału)¹.

Wniosek końcowy: opracowany model czujnika wiernie odtwarza zależność pomiędzy jego sygnałem wejściowym i wyjściowym o częstotliwości $f = 10$ Hz i może być wykorzystany do prawidłowego dobierania parametrów konstrukcyjnych czujnika [17].

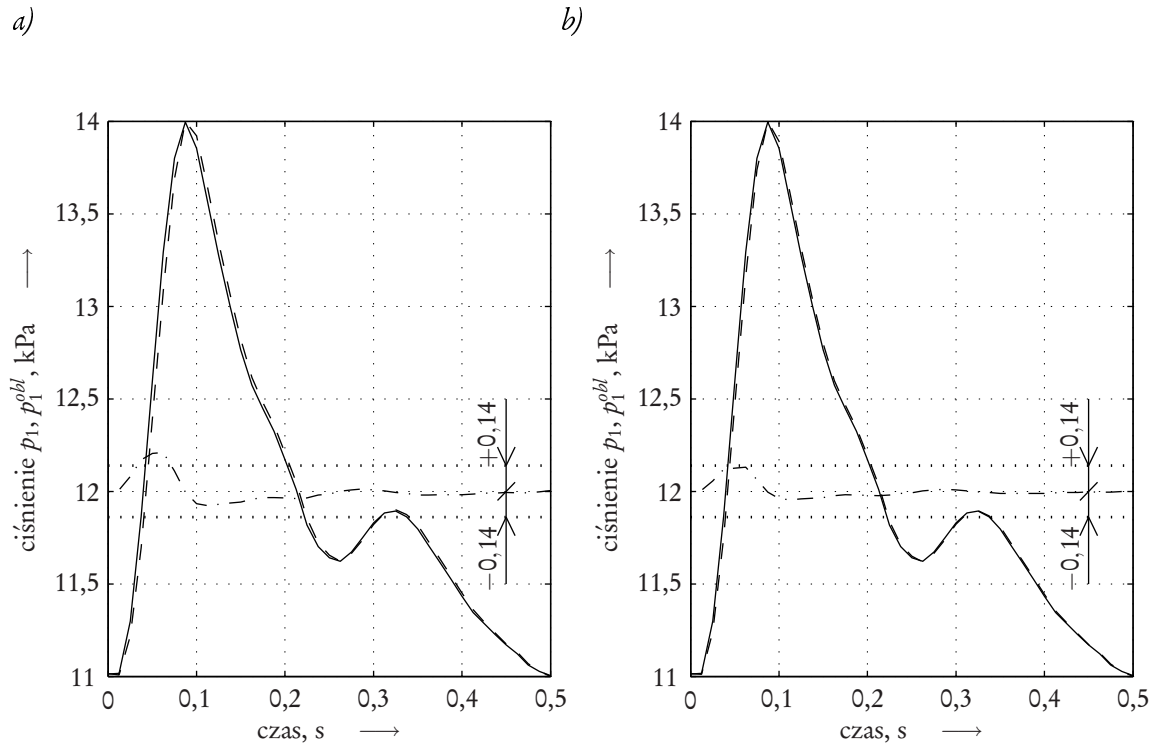
¹ Więcej na temat poprawnej pracy czujnika w rozdziale 5.1

5. Dobór parametrów czujnika

5.1. Kryterium oceny czujnika

Aby ocenić czy dany zestaw parametrów konstrukcyjnych i eksploatacyjnych zapewnia prawidłową pracę czujnika, trzeba było zdefiniować kryterium jego poprawnej pracy. Ponieważ czujnik jest przeznaczony do odtwarzania zmiennego sygnału wejściowego w postaci tętna krwi, o jego przydatności decydują głównie własności dynamiczne. Do oceny odtwarzania przyjęto typowy przebieg fali tętna krwi w tętnicy promieniowej, przedstawiony na rysunku 5.1 (linia ciągła). Jako częstotliwość podstawową przyjęto $f = 2$ Hz, co odpowiada tętnu 120 uderzeń na minutę. Natomiast ciśnienie skurczowe wynosi 14 kPa (105 mmHg) a rozkurczowe 11 kPa (82,5 mmHg). Przed wyborem kryterium poprawnej pracy czujnika, wzięto pod uwagę kilka różnych kryteriów: moduł charakterystyk częstotliwościowych [18], kryterium całkowite $\int_0^{0,5} |(p_1(t) - p_1^{obl}(t))| dt$ oraz kryterium maksimum $\max\{|p_1(t) - p_1^{obl}(t)|\}_{t=0}^{t=0,5}$, gdzie $p_1(t)$ – symulowany przebieg ciśnienia krwi, p_1^{obl} – przebieg ciśnienia krwi obliczony w oparciu o sygnał p_2 i charakterystykę czujnika. Więcej na ten temat znajduje się w punkcie 5.3.

Ze względu na swą prostotę zdecydowano się na kryterium maksimum. Przyjęto również, że działanie czujnika jest poprawne gdy wartość maksimum $p_1 - p_1^{obl}$ nie przekroczy wartości 0,14 kPa, co oznacza, że błąd względny odtwarzania przyjętego przebiegu fali tętna krwi o ciśnieniu skurczowym $p_{1s} = 14$ kPa nie przekroczy wartości $\varepsilon = 1\%$. Dla przykładu na rysunku 5.1 zestawiono przebiegi ciśnień krwi: mierzonego p_1 (linia ciągła) i obliczonego p_1^{obl} (linia przerywana) oraz różnicę tych przebiegów (linia kreska kropka). Przebiegi te uzyskano za pomocą czujników



Rysunek 5.1. Fala tętna krwi: a) parametry czujnika niespełniające kryterium, b) parametry czujnika spełniające kryterium, linia ciągła — ciśnienie wejściowe p_1 , linia przerywana — ciśnienie wyjściowe p_1^{obl} , linia kreska kropka — różnica między p_1 a p_1^{obl} , linie kropkowe — szerokość dopuszczalnej różnicy 0,14 kPa.

o parametrach: średnica dyszy $d_2 = 0,7$ mm, odsadzenie $x_0 = 0,05$ mm, średnica przewodu $d_2 = 2$ mm, długość dławika $l = 10$ mm przy czym w przypadku rysunku 5.1a ciśnienie zasilania czujnika wynosiło $p_z = 30$ kPa a w przypadku rysunku 5.1b — $p_z = 40$ kPa. Linią „kropka kreska” zaznaczono różnice między przebiegami $p_1(t) - p_1^{obl}(t)$ (przesuniętą do wartości 12 kPa) a liniami kropkowymi przedział ciśnień $\pm 0,14$ kPa. Na rysunku 5.1b widać, że spełniona jest nierówność:

$$\max\{|p_1(t) - p_1^{obl}(t)|\}_{t=0}^{t=0,5} \leq 0,14 \quad (5.1)$$

a zatem kryterium poprawnej pracy jest spełnione, natomiast na rysunku 5.1a nie.

5.2. Symulacje pracy czujnika

Do wyznaczenia zestawu parametrów konstrukcyjnych przy których czujnik pracuje poprawnie (wedle kryterium z podpunktu 5.1) posłużył model opisany w poprzednim rozdziale. Wykonano szereg symulacji dla różnych parametrów podlegających doborowi przy projektowaniu czujnika:

- ciśnienie zasilania przyrządu p_z ,
- długość dławika l (przy narzuconej stałej średnicy dławika 0,22 mm),
- średnica przewodu zasilającego d_3 (przy narzuconej stałej długości przewodu 60 cm),
- średnica dyszy d_2 wzmacniacza typu dysza-przysłona (przy narzuconej średnicy membrany i jej sztywności)
- odsadzenie dyszy x_0 .

Na wejście modelu podawano okresowy sygnał ciśnieniowy p_1 przedstawiony na rysunku 5.1a (linia ciągła), natomiast wyjściowe sygnały p_2 obliczano za pomocą modelu czujnika o zadanych parametrach konstrukcyjnych (linia przerywana). W dodatku D.1 znajduje się skrypt Matlaba, który ustawia wartości wyżej wymienionych parametrów i uruchamia symulacje modelu w Simulinku, zmieniając kolejne wartości poszczególnych parametrów. Rejestrowano nie tylko wartości ciśnienia p_2 , ale również ugięcia membrany y . Po wykonaniu wszystkich symulacji dane zapisywane są do pliku. Dopiero na podstawie wartości zapisanych w pliku wybierano zestawy parametrów odpowiadające przyjętemu kryterium poprawnej pracy czujnika

Symulacje wykonano przy czterech różnych wartościach ciśnienia zasilania p_z , długości dławika l i średnic przewodu zasilającego d_3 , oraz przy trzech wartościach średnic dyszy d_2 i dwóch wartościach odsadzenia x_0 . Wykorzystano tzw. pełny plan eksperymentu czyli wykonano symulacje dla każdego z możliwych zestawów parametrów czyli 384 eksperymentów:

- $p_z = 24, 30, 40, 50$ kPa;
- $l = 5, 10, 15, 20$ mm;

- $d_3 = 1, 1,5, 2, 2,5$ mm;
- $d_2 = 0,5, 0,7, 0,9$ mm;
- $x_0 = 0,05, 0,01$ mm.

Ze wstępnych analiz charakterystyk częstotliwościowych wynikało, że na poprawną pracę czujnika mają decydujący wpływ trzy jego parametry: l — długość kapilary, d_3 — średnica przewodu łączącego czujnik z kapilarą oraz ciśnienie zasilania p_z .

5.3. Ocena poprawnej pracy czujnika na podstawie modelu

Po zgromadzeniu wszystkich danych symulacyjnych zajęto się oceną poprawnej pracy czujnika wedle kryterium maksimum (5.1). Pierwszym etapem było przeliczenie wartości ciśnienia wyjściowego p_2 tak aby jego maksimum (ciśnienie skurczowe) i minimum (ciśnienie rozkurczowe) pokrywały się z wartościami ciśnienia skurczowego i rozkurczowego przebiegu p_1 .

$$p_1^{obl}(t) = \frac{p_2(t) - \min(p_2(t))}{\max(p_2(t)) - \min(p_2(t))} \cdot (p_{1s} - p_{1r}) + p_{1r} \quad (5.2)$$

gdzie:

$p_1^{obl}(t)$ — przebieg ciśnienia mierzonego po ujednoczeniu skali,

p_{1r} — wartość ciśnienia rozkurczowego dla przebiegu p_1 ,

p_{1s} — wartość ciśnienia skurczowego dla przebiegu p_1 .

Następnie obliczano wartość maksymalnej różnicy między przebiegami p_1^{obl} a p_1 . Wartości tych różnic dla czujnika o średnicy dyszy $d_2 = 0,7$ mm i odsadzeniu $x_0 = 0,05$ mm znajdują się w tabeli 5.1. Natomiast algorytm w postaci m-pliku Matlaba znajduje się w dodatku D.2.

Jeżeli wartość znajdująca się w tabeli 5.1 przekraczała 0,14 kPa (1 % zakresu pomiarowego) uznawano, że czujnik działa niepoprawnie. Natomiast, zestaw zadanych parametrów czujnika uznawano za poprawny gdy maksymalna różnica między ciśnieniem wejściowym p_1 i przeliczonym wyjściowym p_1^{obl} nie przekraczała $\varepsilon = 0,14$ kPa.

Tabela 5.1. Wartości maksymalnych różnic $p_1 - p_1^{obl}$ w kPa dla czujnika o parametrach: $d_2 = 0,7$ mm, $x_0 = 0,05$ mm

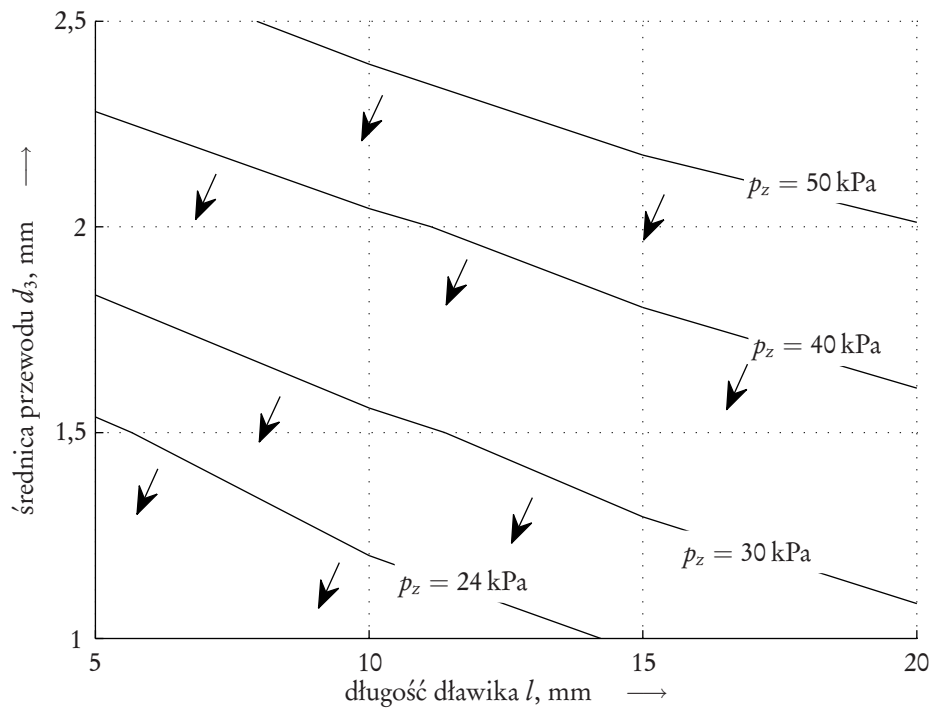
	$p_z = 24$ kPa				$p_z = 30$ kPa			
	$l = 5$	$l = 10$	$l = 15$	$l = 20$	$l = 5$	$l = 10$	$l = 15$	$l = 20$
$d_3 = 1$ mm	0,0840	0,1124	0,1450	0,1812	0,0633	0,0819	0,1031	0,1265
$d_3 = 1,5$ mm	0,1336	0,1810	0,2380	0,3030	0,1000	0,1302	0,1655	0,2061
$d_3 = 2$ mm	0,2175	0,3015	0,4101	0,5391	0,1598	0,2122	0,2759	0,3475
$d_3 = 2,5$ mm	0,3538	0,5127	0,6629	0,7783	0,2591	0,3478	0,4664	0,5920
	$p_z = 40$ kPa				$p_z = 50$ kPa			
$d_3 = 1$ mm	0,0554	0,0538	0,0648	0,0776	0,0594	0,1393	0,0704	0,0552
$d_3 = 1,5$ mm	0,0673	0,0837	0,1024	0,1232	0,0575	0,0616	0,0735	0,0868
$d_3 = 2$ mm	0,1061	0,1329	0,1643	0,2006	0,0798	0,0969	0,1163	0,1381
$d_3 = 2,5$ mm	0,1665	0,2126	0,2683	0,3315	0,1236	0,1514	0,1841	0,2223

5.4. Dobór parametrów konstrukcyjnych czujnika

Na podstawie tabeli 5.1 powstał wykres przedstawiony na rysunku 5.2. Na osi poziomej są zaznaczone długości dławika l natomiast na osi pionowej średnice przewodu zasilającego d_3 , natomiast liniami zaznaczono granice obszarów poprawnej pracy czujnika dla czterech różnych ciśnień zasilania. Obszar poniżej danej linii (zaznaczony strzałkami) oznacza poprawną pracę czujnika o wartości kryterialnej poniżej 0,14 kPa natomiast obszar powyżej danej linii oznacza pracę niepoprawną według tego kryterium.

Dla przykładu, czujnik z dławikiem o długości $l = 10$ mm i średnicy przewodu $d_3 = 2$ mm działa poprawnie przy ciśnieniach zasilania $p_z = 40$ i 50 kPa. Natomiast przy ciśnieniu zasilania o wartości $p_z = 30$ kPa, do poprawnej pracy czujnika, przy przyjętej długości dławika $l = 10$ mm (przykład przebiegu z rys. 5.1a), wymagane jest zmniejszenie średnicy przewodu zasilającego d_3 do wartości około 1,5 mm (przykład przebiegu z rys. 5.1b).

Analiza otrzymanych wyników wykazała, że odsadzenie x_0 i średnica dyszy d_2



Rysunek 5.2. Obszary poprawnej pracy czujnika

mają mały wpływ na właściwości dynamiczne czujnika. Natomiast głównymi czynnikami wpływającymi na te właściwości są: długość dławika l (przy zadanej jego średnicy), średnica przewodu zasilającego d_3 (przy zadanej jego długości) oraz ciśnienie zasilania p_z . Wpływ ciśnienia zasilania na poprawną pracę czujnika okazał się być o wiele większy niż zakładano to w początkowej fazie badań, podobnie było w przypadku średnicy przewodu zasilającego d_3 . Dlatego też, wykres na rysunku 5.2, może być stosowany przy dobieraniu wielkości l , d_3 , i p_z czujników o odsadzeniach w zakresie $x_0 = 0,05 - 0,12$ mm i średnic dyszy w zakresie $d_2 = 0,5 - 0,9$ mm.

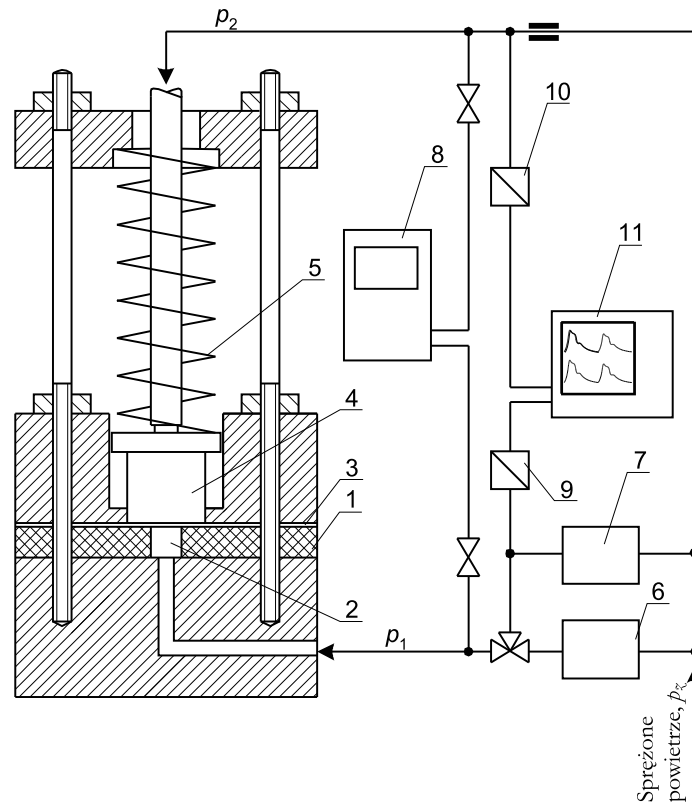
6. Ocena przydatności przyrządu do pomiaru fali tętna krwi

6.1. Badanie na modelu fizycznym tętnicy

Pierwszym etapem oceny przydatności przyrządu do pomiaru fali tętna krwi było przetestowanie go na stanowisku pomiarowym zawierającym model fizyczny tętnicy. Stanowisko pomiarowe składało się z płaskiego kawałka miękkiej gumy 1 (rys. 6.1), w którym wykonano rowek 2, przykryty od góry szczelnie cienką powłoką z lateksu 3. Rowek 2 wraz z powłoką 3 stanowią model spłaszczonej tętnicy, a guma 1 — tkankę otaczającą tę tętnicę. Badany czujnik 4 umieszczony jest nad tętnicą i dociskany do niej za pośrednictwem sprężyny 5. Ciśnienie sprężonego powietrza p_1 , reprezentujące ciśnienie krwi, jest nastawiane ręcznie zadajnikiem 6 (badania statyczne) lub generowane, w postaci fali tętna krwi, za pomocą specjalnego generatora 7 [16] (badania dynamiczne). Ciśnienie p_1 oraz ciśnienie wyjściowe p_2 czujnika 4, podczas badań statycznych, mierzono manometrem 8, a podczas badań dynamicznych — rejestrowano za pośrednictwem przetworników pneumoelektrycznych 9, 10 i oscyloskopu 11.

Przed wyznaczeniem charakterystyk czujnika na modelu tętnicy, zmierzono siłę, niezbędną do właściwego spłaszczenia tętnicy na żywym organizmie. Pomiar przeprowadzono na tętnicy promieniowej jedenastu osób: sześciu pań i pięciu panów w wieku od 25 do 76 lat.

Siłomierzem w tych pomiarach była sprężyna śrubowa, za pośrednictwem której czujnik dociskano do tętnicy, natomiast, wskaźnikiem właściwego jej spłaszczenia było pojawienie się na ekranie oscyloskopu fali tętna krwi o maksymalnej amplitu-



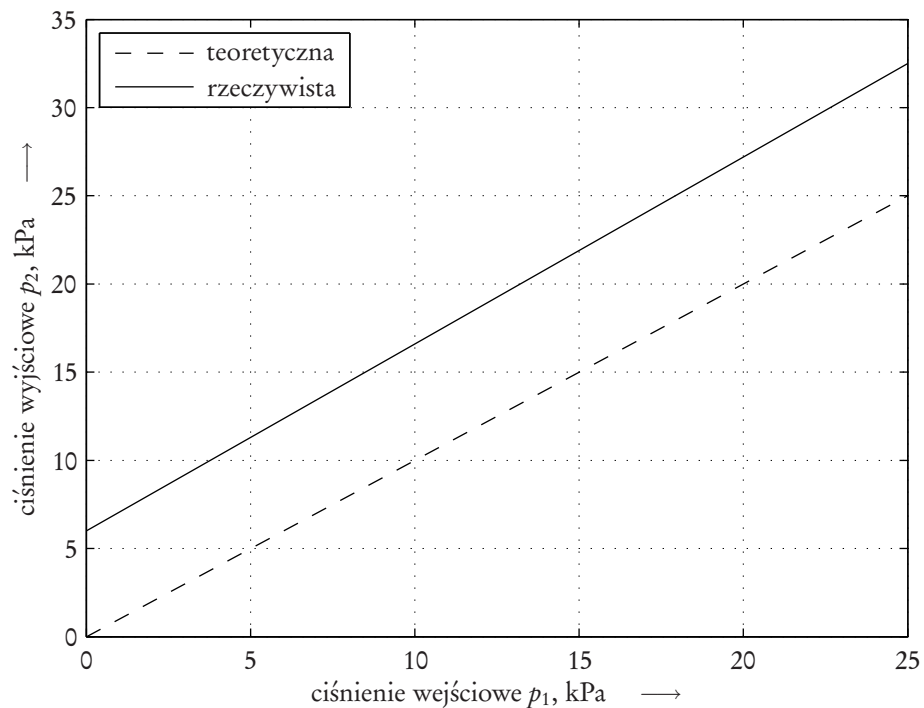
Rysunek 6.1. Schemat stanowiska do badania czujnika na modelu fizycznym tętnicy

dzie. Średnia wartość siły docisku wyniosła $F = 1,50 \text{ N}$, a odchylenie standardowe osiemdziesięciu ośmiu odczytów – $0,22 \text{ N}$. Siła ta była wywierana na czujnik podczas wyznaczania jego charakterystyk, zarówno statycznych jak i dynamicznych, na modelu tętnicy.

Na rysunku 6.2 przedstawiono charakterystykę statyczną czujnika, wyznaczoną na modelu fizycznym tętnicy.

Charakterystykę tę opisuje równanie:

$$p_2 = a + b \cdot p_1 \quad (6.1)$$



Rysunek 6.2. Charakterystyka statyczna czujnika przy badaniu na sztucznej tętnicy

lub

$$p_1 = \frac{p_2 - a}{b} \quad (6.2)$$

$$r = 0,99; \quad P < 0,001$$

gdzie:

p_1 — ciśnienie w modelu tętnicy w kPa,

p_2 — ciśnienie wyjściowe czujnika w kPa,

r — współczynnik korelacji,

P — poziom istotności współczynnika korelacji.

$a = 5,9$ kPa; $b = 1,08$ — stałe współczynniki wyznaczone metodą najmniejszych kwadratów

Charakterystyka ta w zakresie pomiarowym jest liniowa i jak można było się

spodziewać, przebiega powyżej charakterystyki teoretycznej i jest praktycznie do niej równoległa.

Natomiast, wynikiem badań dynamicznych były przebiegi fali tętna p_1 , zmierzone wewnątrz modelu tętnicy i przebiegi ciśnienia p_2 , zmierzone na wyjściu czujnika. Przykład takich przebiegów, o częstotliwości 2 Hz i o tej samej podziałce osi pionowej, pokazano na oscylogramie — rysunek 6.3a.

Jak widać wskazania czujnika p_2 są zawyżone w stosunku do rzeczywistego przebiegu p_1 i powinny być skorygowane. Do korekcji tej użyto zależności (6.2), zapisanej w postaci:

$$p_1^{obl} = \frac{p_2 - a}{b} \quad (6.3)$$

$$a = 5,9 \text{ kPa}, \quad b = 1,08$$

gdzie:

p_1^{obl} — ciśnienie wewnątrz tętnicy, obliczone w oparciu o wskazanie czujnika p_2 w kPa.
 a i b — współczynniki wyznaczone metodą najmniejszych kwadratów.

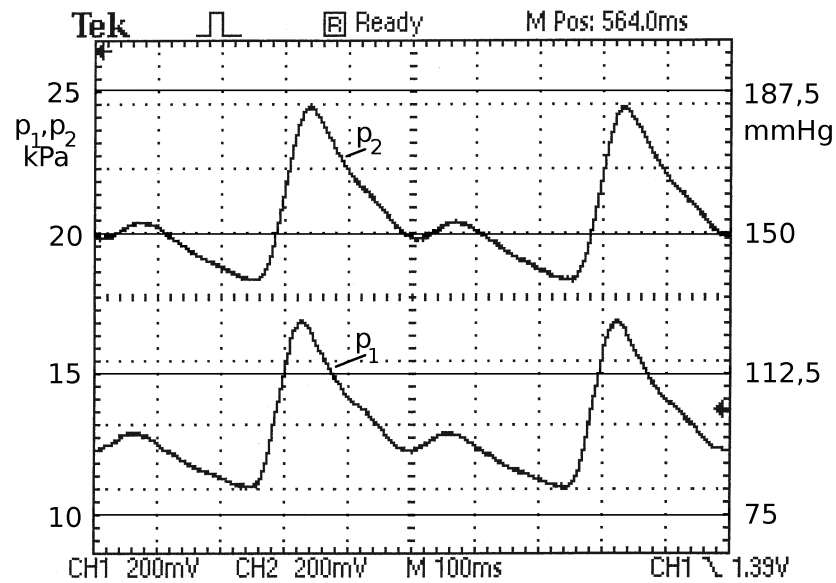
Na rysunku 6.3b pokazano rzeczywisty przebieg ciśnienia p_1 wewnątrz tętnicy, na którym gwiazdkami zaznaczono punkty przebiegu ciśnienia p_1^{obl} , obliczone w oparciu o równanie (6.3)¹. Średnia rozbieżność między przebiegami, wyznaczona w 21 punktach, wynosi niecałe 1,5 % w odniesieniu do wartości ciśnienia skurczowego.

Z przeprowadzonego eksperymentu wynikają następujące wnioski:

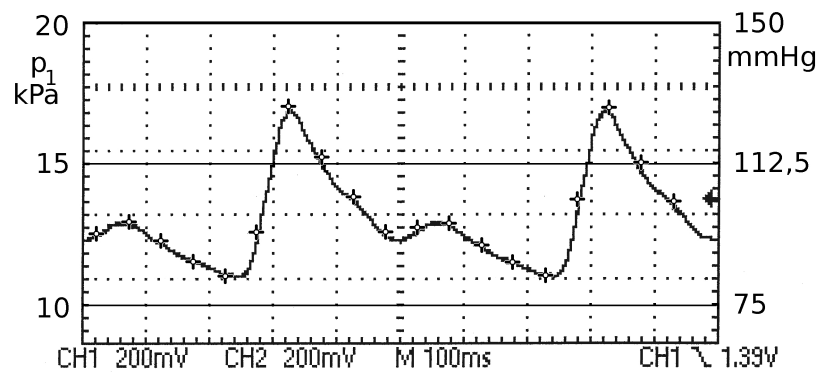
- czujnik ma bardzo dobre własności dynamiczne: amplitudy obydwu przebiegów są praktycznie sobie równe,
- charakterystyka statyczna czujnika jest liniowa: kształty obydwu przebiegów fali tętna pokrywają się,
- korekcja przebiegu ciśnienia wyjściowego czujnika p_2 jest przeprowadzona poprawnie.

¹ wcześniej zlikwidowano przesunięcie fazowe pomiędzy przebiegami ciśnień p_1 i p_2

a)



b)



Rysunek 6.3. Przebiegi fali tętna podczas badań na sztucznej tętnicy

6.2. Badania laboratoryjne na pacjentach

Ponieważ parametry modelu niezbyt dokładnie odzwierciedlają rzeczywiste własności tętnicy i otaczającej ją tkanki, wyznaczona charakterystyka statyczna czujnika (rys. 6.2) nie może być użyta do korekty zmierzonej na pacjencie fali tętna krwi.

W tym celu konieczna jest charakterystyka czujnika, wyznaczona na pacjentach. Wyznaczanie tej charakterystyki polegało na porównaniu odczytów ciśnienia skurczowego i rozkurczowego uzyskanych za pomocą przyrządu badanego i przyrządu wzorcowego. Zgodnie z „Międzynarodowymi wytycznymi...” [19] do wzorcowania badanego przyrządu używa się tradycyjnego przyrządu mankietowego i odpowiednio zmodyfikowanej metody tonów Koratkowa. Podczas pomiaru używano stetoskopu z dwoma parami słuchawek a w odczycie uczestniczyły dwie osoby. Jeśli wyniki ich odczytów różniły się o więcej niż 4 mmHg pomiar powtarzano. W pierwszym etapie pomiarów zgodnie z [19] mierzono ciśnienia piętnastu pacjentom, wykonując na każdym z nich siedem pomiarów, cztery pomiary przyrządem wzorcowym i trzy pomiary przyrządem badanym, przeprowadzając badania na przemian raz wzorcowym raz badanym, zaczynając i kończąc serię na pomiarze przyrządem wzorcowym. Literą „B” oznaczmy pomiar przyrządem badanym natomiast literą „W” pomiar przyrządem wzorcowym. Wówczas przebieg wzorcowania dla każdego pacjenta wygląda następująco:

$$W_1 \quad B_1 \quad W_2 \quad B_2 \quad W_3 \quad B_3 \quad W_4$$

Następnie pomiary organizowano w trzy pary:

$$(W_1 B_1 \quad B_1 W_2)_1 \quad (W_2 B_2 \quad B_2 W_3)_2 \quad (W_3 B_3 \quad B_3 W_4)_3$$

Z każdej pary wybierano tę, która posiada mniejsze różnice odpowiadających sobie ciśnień skurczowych i rozkurczowych.

W wyniku takiego wzorcowania, w którym wzięło udział łącznie 15 pań i panów w wieku od 25 do 76 lat otrzymano 45 punktów pomiarowych, w oparciu o które wyznaczono uniwersalną charakterystykę przyrządu, opisaną równaniem

$$p_1^{obl} = \frac{p_2 - a}{b} \quad (6.4)$$

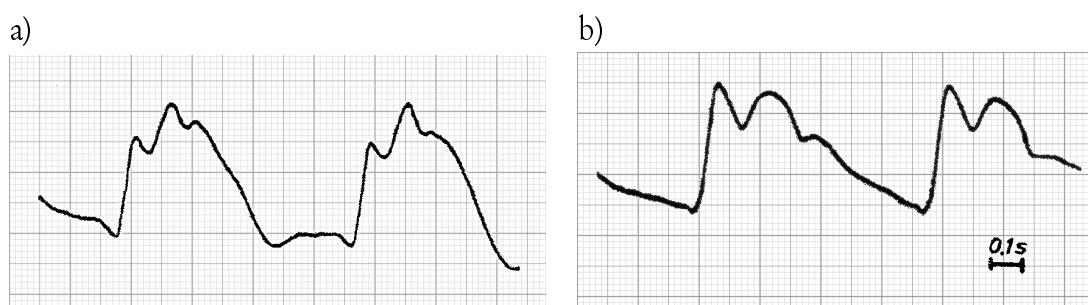
gdzie:

$$a = 43 \pm 13 \text{ mmHg}, \quad b = 0,83 \pm 0,1$$

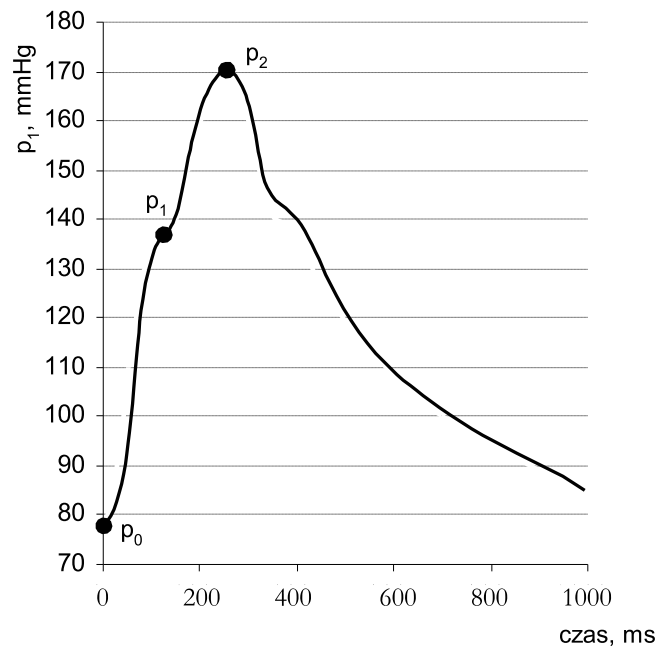
Współczynnik korelacji powyższego wzorcowania wynosi: $r = 0,79$ a poziom istotności jest mniejszy od $0,05$. Dla porównania przyrząd z matrycą piezorezystorów opisany we wstępie posiada współczynnik korelacji dla ciśnienia skurczowego $r_s = 0,80$ natomiast dla ciśnienia rozkurczowego $r_r = 0,77$ przy poziomie istotności $P < 0,001$ [20].

Ze względu na stosunkowo duże odchylenie standardowe współczynników $a(\pm 13 \text{ mmHg})$ i $b(\pm 0,1)$ odczytanie z wykresu wartości ciśnień (w tym również ciśnienia skurczowego i rozkurczowego) mogą znacznie różnić się od wartości rzeczywistych (do $\pm 15 \text{ mmHg}$)

W tym miejscu należy przypomnieć, że głównym przeznaczeniem omawianego przyrządu jest wyznaczanie kształtu fali tętna krwi, a nie pomiar ciśnienia skurczowego i rozkurczowego, z czym obecnie nie ma większych problemów. Jak wiadomo, już po samym kształcie fali tętna krwi można rozpoznać szereg schorzeń układu krążenia, a niektóre współczynniki liczbowe charakteryzujące stan tego układu można wyznaczać bez dokładnej znajomości podziałki ciśnienia. Jako przykład, na rysunku 6.4 pokazano przebiegi fali tętna w tętnicy szyjnej: przy niedomykalności zastawki aorty (rys. 6.4a) oraz podzastawkowym zwężeniu drogi odpływu lewej komory serca (rys. 6.4b) [7].



Rysunek 6.4. Przebiegi fali tętna w tętnicy szyjnej [7].



Rysunek 6.5. Fala tętna krwi w aortalnej [6].

Ponadto, przebieg fali tętna w tętnicy promieniowej można obecnie przetransponować na przebieg fali tętna w aortalnej [6]. Z przebiegu tego wyznacza się szereg ważnych współczynników, w tym między innymi współczynnik augmentacji W_a i współczynnik Buckberga (współczynnik żywotności) W_b :

$$W_a = \frac{p_2 - p_1}{p_2 - p_0} \cdot 100 \% \quad (6.5)$$

$$W_b = \frac{A_r}{A_s} \cdot 100 \% \quad (6.6)$$

gdzie:

p_0 — wartość ciśnienia rozkurczowego

p_1, p_2 — wartości ciśnień pierwszego i drugiego wzniesienia na linii skurczu (rys. 6.5)

A_r — pole powierzchni pod linią rozkurczu

A_s — pole powierzchni pod linią skurczu

Jeżeli jednak, na zarejestrowanych przebiegach konieczna jest dokładna podziałka ciśnienia, wówczas przeprowadza się wzorcowanie przyrządu na konkretnym pacjencie. Zmierzone wartości ciśnień (w mmHg) skurczowych i rozkurczowych, wskazywanych przez przyrząd mankietowy (p_{1s} , p_{1r}) oraz przyrząd do pomiaru fali tętna krwi (p_{2s} , p_{2r}), wyznaczają dwa punkty charakterystyki statycznej. Linia prosta, przechodząca przez te punkty jest przyjmowana jako charakterystyka przyrządu; opisuje ją równanie:

$$p_2 = b \cdot p_1 + a \quad \text{lub} \quad p_1^{obl} = \frac{p_2 - a}{b} \quad (6.7)$$

$$b = \frac{p_{2s} - p_{2r}}{p_{1s} - p_{1r}}; \quad a = p_{2s} - b \cdot p_{1s}$$

gdzie:

p_1^{obl} i p_2 — ciśnienie obliczone i wskazywane przez przyrząd, w mmHg.

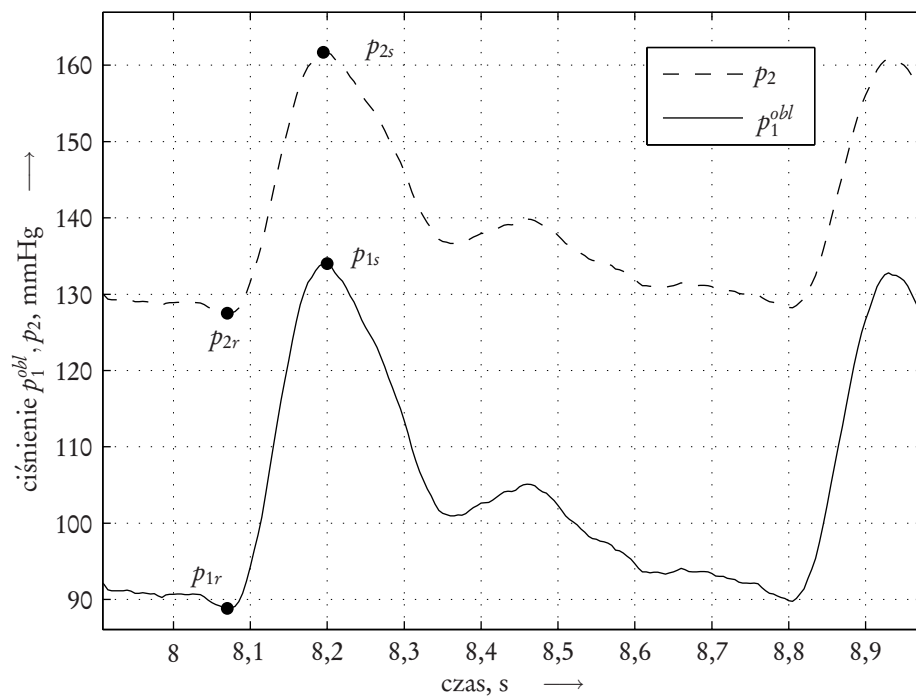
W należącym do przyrządu komputerze 3 (rys. 3.1) znajduje się program, który, w parciu o wprowadzone wartości współczynników a i b , rejestrowaną bezpośrednio falę tętna krwi przetwarza na falę rzeczywistą.

Dla przykładu, na rysunku 6.6 pokazano zmierzony omawianym przyrządem przebieg fali tętna krwi p_2 oraz przebieg skorygowany p_1^{obl} , wg równania (6.7).

Wartości ciśnienia skurczowego i rozkurczowego wskazywane przez badany przyrząd i zmierzonych przyrządem mankietowym wynoszą: $p_{2s} = 162$ mmHg, $p_{2r} = 128$ mmHg, $p_{1s} = 134$ mmHg, $p_{1r} = 89$ mmHg. Wyliczone z tych wartości współczynniki charakterystyki wynoszą: $a = 60$ mmHg, $b = 0,76$.

6.3. Badania kliniczne

Badania kliniczne przeprowadzono w Klinice Nefrologii i Transplantacji Akademii Medycznej we Wrocławiu. Badania polegały na równoczesnym pomiarze fali tętna krwi pacjenta dwoma przyrządami: przyrządem badanym na prawym nadgarstku

Rysunek 6.6. Przykład zmierzonej (p_2) i skorygowanej (p_1^{obl}) fali tętna krwi.

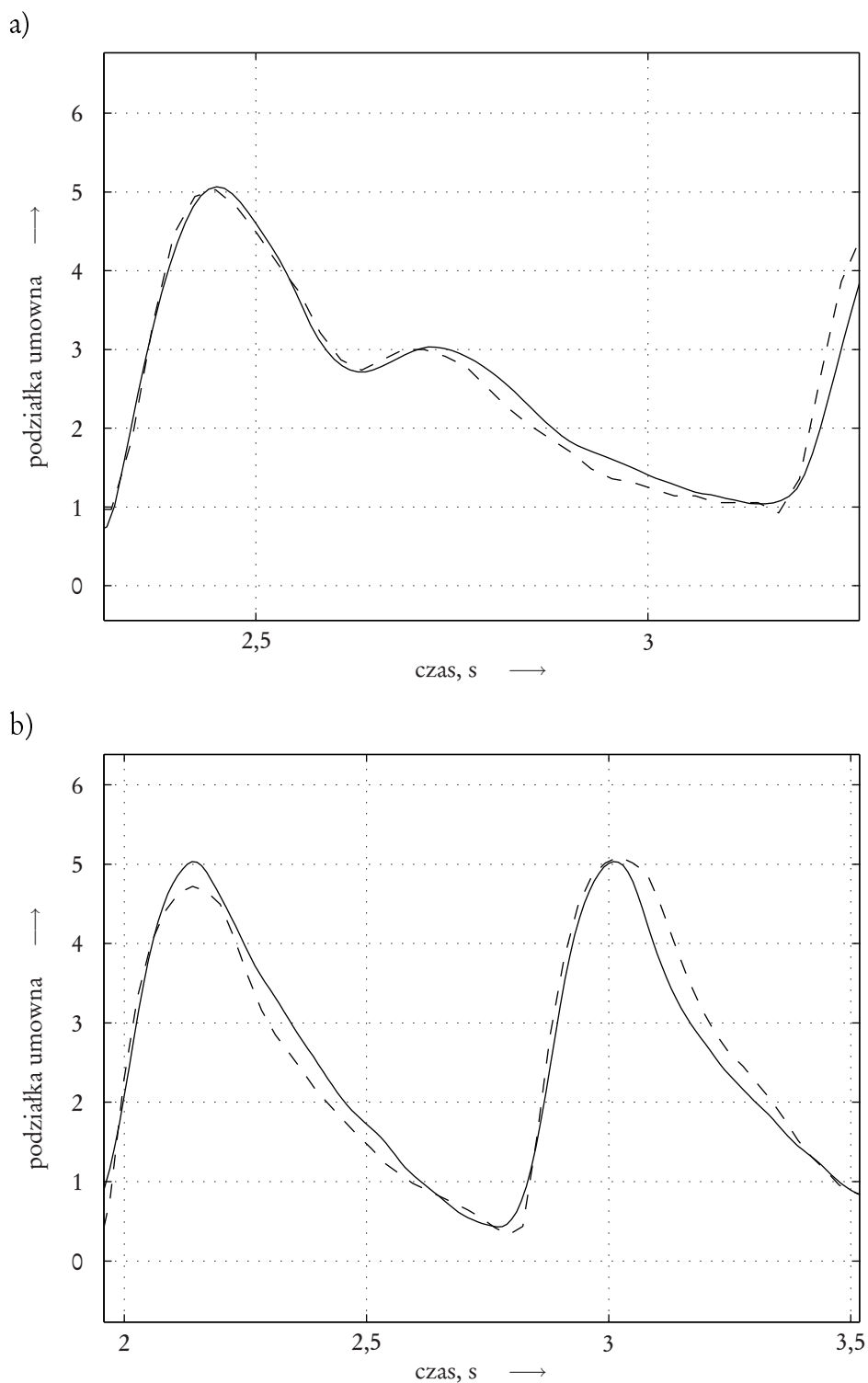
Rysunek 6.7. Pletyzmograf firmy „BioMedix inc.”

oraz pletyzmografem firmy „BioMedix inc.” typu „PVL” (rys. 6.7) na lewym nadgarstku.

Ponieważ (z powodu mankietu pneumatycznego) częstotliwość przenoszenia pletyzmografu jest niższa niż badanego przyrządu, przed porównaniem obydwu wyników, sygnał wyjściowy badanego przyrządu przechodził przez filtr dolnoprzepustowy w postaci członu inercyjnego I-rzędu.

Na rysunkach 6.8a i b zestawiono ze sobą przebiegi sygnałów: z pletyzmografu — linia przerywana oraz przefiltrowanego sygnału z badanego przyrządu. Przebiegi na rysunku 6.8a dotyczą pacjenta w wieku 30 lat, a na rysunku 6.8b — w wieku 60 lat.

Przeprowadzone badania traktowane są jako wstęp do badań z zastosowaniem pomiarów fali tętna krwi metodą krwawą. Mimo to już na obecnym etapie widać, że charakterystyczne cech przebiegu fali tętna krwi (np. obecność zagłębienia dykrotycznego) są odtwarzane opracowanym przyrządem z wystarczającą dla praktyki lekarskiej dokładnością.



Rysunek 6.8. Zestawienie przebiegu sygnału z pletyzmografu (linia przerywana) oraz przefiltrowanego sygnału z badanego przyrządu (linia ciągła), a) pacjent w wieku 30 lat, b) pacjent w wieku 60 lat

7. Podsumowanie i wnioski

Za najważniejsze osiągnięcia mojej pracy uważam następujące zagadnienia. Po pierwsze, opracowanie i zweryfikowanie modelu matematycznego pneumatycznego wzmacniacza typu dysza-przysłona oraz, opartego na nim, czujnika niskich ciśnień. Mimo iż wzmacniacz typu dysza-przysłona jest powszechnie stosowany w automatyce i robotyce, do dziś nie posiada on ogólnego modelu matematycznego. Opracowany przeze mnie model, ma co prawda ograniczony do niskich ciśnień zakres stosowania, ale jego zaletą jest to, że pozwala głębiej niż dotychczas wnikać w podstawy fizyczne działania wzmacniacza typu dysza-przysłona. Przykładem tego może być wyjaśnienie za pomocą tego modelu pewnego paradoksu. Mianowicie, z doświadczalnych charakterystyk częstotliwościowych wynika, że stała czasowa samego wzmacniacza jest **większa** niż stała czasowa czujnika, który oprócz wzmacniacza posiada jeszcze dodatkowo membranę i ujemne sprzężenie zwrotne¹.

Drugim osiągnięciem jest opracowanie wytycznych do projektowania pneumatycznych czujników ciśnienia. Wyróżniono między innymi parametry, decydujące o właściwościach czujnika, zwłaszcza dynamicznych: oporność dławika, pojemność przewodu zasilającego, ciśnienie zasilania i podano ich wartości liczbowe, gwarantujące poprawną pracę czujnika (rys. 5.2).

Trzecie zagadnienie to budowa modelu fizycznego tętnicy i otaczającej ją tkanki. Badania z użyciem tego modelu pozwoliły lepiej zrozumieć mechanizm przesuwania się w czasie pomiaru charakterystyki statycznej czujnika względem charakterysty-

¹ Publikacja na ten temat jest w przygotowaniu

ki teoretycznej i potwierdziły słuszność zastosowanej korekty sygnały wyjściowego czujnika.

Do czwartego osiągnięcia zaliczam opracowanie, budowę i oprogramowanie interfejsu do współpracy czujnika z komputerem przy jednoczesnej filtracji sygnału. Wcześniejsze wersje, tego interfejsu (wykonane na zamówienie), poprzez zastosowaną w nich cyfrową filtrację wprowadzały zniekształcenia sygnału.

Piąte zagadnienie to program komputerowy do analizy i przetwarzania sygnału wyjściowego przetwornika piezoelektrycznego.

WNIOSKI

Opracowany prototyp przyrządu do bezinwazyjnego pomiaru i rejestracji fali tętna krwi już w obecnym stanie może być wykorzystany do prac badawczych w Klinikach¹.

W przyszłości przewidywana jest miniaturyzacja modułu zasilającego i udoskonalenie programu komputerowego do analizy fali tętna krwi. Tak udoskonalony przyrząd mógłby służyć lekarzom pierwszego kontaktu do badań przesiewowych pod kątem ryzyka zawału serca i udaru mózgowego, a pacjentom z chorobami kardiologicznymi lub cukrzycą umożliwiłby obserwowanie we własnym zakresie reakcji układu krwionośnego na przyjmowane leki i ustrzec się przed ich przedawkowaniem. Natomiast w szpitalach opracowany przyrząd może być użyteczny, gdy konieczny jest natychmiastowy pomiar fali tętna krwi lub gdy pomiar tej fali metodą krwawą nie jest możliwy.

¹ Nawiązano, na razie nieformalną, współpracę pomiędzy Zakładem Automatyki i Kriogeniki Politechniki Wrocławskiej a Kliniką Neurochirurgii oraz Kliniką Nefrologii i Transplantacji Akademii Medycznej we Wrocławiu w sprawie wykozystania opracowanego przyrządu do badań naukowych prowadzonych w tych Klinikach.

A. Źródła programu zainstalowanego w mikrokontrolerze

Program: „USBmodul_V2.6” oparty jest na sterowniku Microchip „USBcdc”.

A.1. Lista zmian w pliku: „autofiles/usbcfg.h”

W stosunku do sterownika firmy Microchip zmieniono plik: „USBmodul_V2.6/autofiles/usbcfg.h”, lista zmian wygenerowana za pomocą programu „diff”:

```
1 diff USBcdc/fw/Cdc/autofiles/usbcfg.h USBmodul_V2.6/autofiles/usbcfg.h
2 43c43
3 > #define UCFG_VAL                _PUEN|_TRINT|_LS|MODE_PP //to use USB 1.1
4 ---
5 < #define UCFG_VAL                _PUEN|_TRINT|_FS|MODE_PP
6 45,46c45,46
7 > //#define USE_SELF_POWER_SENSE_IO //JKcommented v1.24
8 > //#define USE_USB_BUS_SENSE_IO //no USB and BUS sense
9 ---
10 < #define USE_SELF_POWER_SENSE_IO
11 < #define USE_USB_BUS_SENSE_IO
```

A.2. Plik nagłówkowy: „user/user.h”

W stosunku do sterownika firmy Microchip zmieniono w całości plik: „USBmodul_V2.6/user/user.h”, źródła znajdują się poniżej:

```
1 //Jacek Kosek (c) 20-7-2006 1-8-2006
2
3 #ifndef USER_H
4 #define USER_H
```



```

5
6 /** P U B L I C P R O T O T Y P E S *****/
7 void UserInit(void);
8 void ProcessIO(void);
9
10 #endif //USER_H

```

A.3. Implementacja, plik: „user/user.c”

W stosunku do sterownika firmy Microchip zmieniono w całości plik: „USBmodul_V2.6/user/user.c”, źródła znajdują się poniżej:

```

1 //Jacek Kosek (c) 20-7-2006 1-8-2006
2 //All user stuff is placed here, initialisation and base loop
3
4 /** I N C L U D E S *****/
5 //choosing right pic (microcontroller type), this include autodetect hardware.
6 #include <p18cxxx.h>
7 //Analog to Digital and Timers are also used.
8 #include <adc.h>
9 #include <timers.h>
10
11 //Some nice definitions of data types types
12 #include "system\typedefs.h"
13
14 //all USB includes can be add by this file
15 #include "system\usb\usb.h"
16
17 //I shut get off this file because it is for original commercial evaluation board
18 //but it has good definitions witch I use.
19 #include "io_cfg.h" // I/O pin mapping
20
21 //My user file definitions there is nothing special
22 #include "user\user.h"
23
24 //RealTime functions
25 #include "realtime.h"
26
27 /** V A R I A B L E S *****/
28 #pragma udata
29 //two states off two switches (it indicates old state of this buttons)
30 byte old_sw2,old_sw3;
31
32 //timer_value = sampling_time * cpu_freq / 4 / internal_loops
33 // 3000 = 0.01 * 24M / 4 / 20
34 word timer_value = 1200;

```

```

35
36 //How many probes for mean
37 byte probes_for_mean=25;
38
39 //How many channels
40 byte ADCchannels=1;
41
42
43 typedef enum {
44     waiting_for_command,
45     waiting_to_start_sending,
46     sending,
47     configuring_usb,
48     command_recived
49 } StatusDEF;
50 StatusDEF status=configuring_usb;
51
52     static dword wait_count =0;
53
54     static unsigned char input_buffer[] = "S";
55
56
57 /** P R I V A T E   P R O T O T Y P E S   *****/
58 //initializations functions
59 void InitializeTIMER0(void);
60 //void InitializeADC(void);
61 void InitializeADC_1CH(void);
62 void InitializeADC_2CH(void);
63 //blinking leds shows USB status
64 void BlinkUSBStatus(void);
65 //configuring while module after reciving command
66 void configure_status(void);
67 //is switches changing status?
68 BOOL Switch2IsPressed(void);
69 BOOL Switch3IsPressed(void);
70
71 /** D E C L A R A T I O N S   *****/
72 #pragma code
73
74 //*****
75 //Base Initialization function called by main function
76 void UserInit(void)
77 {
78     mInitAllLEDs();
79     mInitAllSwitches();
80     old_sw2 = sw2;
81     old_sw3 = sw3;
82     InitializeTIMER0();
83 //     InitializeADC();
84 }//end UserInit

```

```
85
86 //this timer will starts the A/D conversions every 3000 cycles
87 void InitializeTIMER0(void)
88 {
89     OpenTimer0( TIMER_INT_OFF & T0_16BIT & T0_SOURCE_INT & T0_PS_1_1 );
90 }
91
92 /*void InitializeADC(void)
93 {
94     OpenADC( ADC_FOSC_8 & ADC_RIGHT_JUST & ADC_2_TAD,
95             ADC_CH0 & ADC_INT_OFF & ADC_VREFPLUS_VDD &
96             ADC_VREFMINUS_VSS, 0);
97     ConvertADC();          // Start conversion
98 }
99 */
100 void InitializeADC_1CH(void)
101 {
102     ADCchannels=1;
103     OpenADC( ADC_FOSC_8 & ADC_RIGHT_JUST & ADC_2_TAD,
104             ADC_CH0 & ADC_INT_OFF & ADC_VREFPLUS_VDD &
105             ADC_VREFMINUS_VSS, 0);
106     ConvertADC();          // Start conversion
107 }
108
109 void InitializeADC_2CH(void)
110 {
111     ADCchannels=2;
112     OpenADC( ADC_FOSC_8 & ADC_RIGHT_JUST & ADC_2_TAD,
113             ADC_CH0 & ADC_INT_OFF & ADC_VREFPLUS_VDD &
114             ADC_VREFMINUS_VSS, 0);
115     ConvertADC();          // Start conversion
116 }
117
118
119 //*****
120 //Base loop function
121 void ProcessIO(void)
122 {
123     //blink leds to see what is happening
124     BlinkUSBStatus();
125
126
127     // User Application USB tasks if connection is established
128     if ((usb_device_state < CONFIGURED_STATE) || (UCONbits.SUSPND == 1)) {
129         status = configuring_usb;
130         return;
131     }
132
133     // Read from USB
134     if (getsUSBUSART(input_buffer,1))
```

```
135     {
136         status = command_recived;
137     }
138
139     //switching between status
140     switch (status){
141         case configuring_usb :
142             //usb was not configured till now
143             //now we should wait for command
144             status = waiting_for_command;
145         break;
146         case waiting_to_start_sending :
147             if (wait_count == 0){
148                 status = command_recived;
149             }
150             else wait_count--;
151         break;
152         case sending :
153             if (INTCONbits.TMR0IF) { //when timer is overloaded do the
154                 //conversion and rest of stuff
155                 everyProbe ();
156             } //end if (INTCONbits.TMR0IF) — time to read from ADC
157         break;
158         case waiting_for_command :
159             //generally do nothing
160             //but:
161             if (input_buffer[0] == 'S') {
162                 wait_count = 100000U;
163                 status = waiting_to_start_sending;
164             }
165         break;
166         case command_recived :
167             configure_status ();
168         break;
169     }
170 } //end ProcessIO
171
172
173
174 void configure_status(void) {
175     switch (input_buffer[0]){
176         case 'S': //default configuration
177             InitializeADC_1CH ();
178
179             //timer_value = sampling_time * cpu_freq / 4 / internal_loops
180             //      3000 = 0.01 * 24M / 4 / 20
181             timer_value = 1200;
182
183             //How many probes for mean
184             probes_for_mean = 25;
```

```

185         status = sending;
186     break;
187     case 'F': //same as default configuration
188         InitializeADC_1CH ();
189         //timer_value = sampling_time * cpu_freq / 4 / internal_loops
190         //      3000 = 0.01 * 24M / 4 / 20
191         timer_value = 1200;
192         //How many probes for mean
193         probes_for_mean = 25;
194         status = sending;
195     break;
196     case 'G': //same as default configuration
197         InitializeADC_2CH ();
198         //timer_value = sampling_time * cpu_freq / 4 / internal_loops
199         //      3000 = 0.01 * 24M / 4 / 20
200         timer_value = 1500;
201         //How many probes for mean
202         probes_for_mean = 10;
203         status = sending;
204     break;
205     case 'E' :
206         //basically start do nothing
207         status = waiting_for_command;
208     break;
209     default :
210         //continue sending or start last configured sending
211         status = waiting_for_command;
212 }
213 }
214
215
216
217 //*****
218 void BlinkUSBStatus(void)
219 {
220     static word led_count=0;
221
222     if(led_count == 0)led_count = 10000U;
223     led_count--;
224
225     #define mLED_Both_Off()      {mLED_1_Off ();mLED_2_Off ();}
226     #define mLED_Both_On()      {mLED_1_On ();mLED_2_On ();}
227     #define mLED_Only_1_On()    {mLED_1_On ();mLED_2_Off ();}
228     #define mLED_Only_2_On()    {mLED_1_Off ();mLED_2_On ();}
229
230     if (UCONbits.SUSPND == 1)
231     {
232         if (led_count == 0)
233         {
234             mLED_1_Toggle ();

```

```

235         mLED_2 = mLED_1;           // Both blink at the same time
236     } //end if
237 }
238 else
239 {
240     if (usb_device_state == DETACHED_STATE)
241     {
242         mLED_Both_Off ();
243     }
244     else if (usb_device_state == ATTACHED_STATE)
245     {
246         mLED_Both_On ();
247     }
248     else if (usb_device_state == POWERED_STATE)
249     {
250         mLED_Only_1_On ();
251     }
252     else if (usb_device_state == DEFAULT_STATE)
253     {
254         mLED_Only_2_On ();
255     }
256     else if (usb_device_state == ADDRESS_STATE)
257     {
258         if (led_count == 0)
259         {
260             mLED_1_Toggle ();
261             mLED_2_Off ();
262         } //end if
263     }
264     else if (usb_device_state == CONFIGURED_STATE)
265     {
266         if (led_count == 0)
267         {
268             mLED_1_Toggle ();
269             mLED_2 = !mLED_1;           // Alternate blink
270         } //end if
271     } //end if (...)
272 } //end if (UCONbits.SUSPND...)
273
274 } //end BlinkUSBStatus
275
276 BOOL Switch2IsPressed (void)
277 {
278     if (sw2 != old_sw2)
279     {
280         old_sw2 = sw2;                 // Save new value
281         if (sw2 == 0)                  // If pressed
282             return TRUE;              // Was pressed
283     } //end if
284     return FALSE;                     // Was not pressed

```

```

285 } //end Switch2IsPressed
286
287 BOOL Switch3IsPressed (void)
288 {
289     if (sw3 != old_sw3)
290     {
291         old_sw3 = sw3;           // Save new value
292         if (sw3 == 0)           // If pressed
293             return TRUE;       // Was pressed
294     } //end if
295     return FALSE;              // Was not pressed
296 } //end Switch3IsPressed
297
298 /** EOF user.c *****/

```

A.4. Plik nagłówkowy: „realtime.h”

W stosunku do sterownika firmy Microchip dodano plik:

„USBmodul_V2.6/realtime.h”, źródła znajdują się poniżej:

```

1 //Jacek Kosek (c) 20-7-2006 1-8-2006
2 void everyProbe ();
3 void everySend ();

```

A.5. Implementacja, plik: „realtime.c”

W stosunku do sterownika firmy Microchip dodano plik:

„USBmodul_V2.6/realtime.c”, źródła znajdują się poniżej:

```

1 //Jacek Kosek (c) 20-7-2006 1-8-2006
2
3 /** I N C L U D E S *****/
4 //choosing right pic (microcontroller type), this include autodetect hardware.
5 #include <p18cxxx.h>
6 //Analog to Digital and Timers are also used.
7 #include <adc.h>
8 #include <timers.h>
9 //Some nice definitions of data types types
10 #include "system\typedefs.h"
11
12 //all USB includes can be add by this file
13 #include "system\usb\usb.h"
14

```

```
15 //I shut get off this file because it is for original commercial evaluation board
16 //but it has good definitions witch I use.
17 #include "io_cfg.h"           // I/O pin mapping
18
19 //My user file definitions , there is nothing special
20 #include "user\user.h"
21
22 //RealTime functions
23 #include "realtime.h"
24
25 #pragma udata
26
27 //result of ADC conversion
28 DWCRD voltage[4]={0,0,0,0};
29
30 //number of actual conversions
31 unsigned char probka=0;
32
33 //numer of actual chanel
34 unsigned char ADCchannel=1;
35
36 //this is what I send to PC:
37 //first is char 'F'
38 //second is High byte of conversion
39 //third is Low byte of conversion
40 //and last is supplement (addition) to zero
41 unsigned char bufor[]="F11223344C";
42
43 //in this variable is stored number of cycles between timer overload and called "if"
44 //this variable is used to right set up next period of time in timer
45 word strataT;
46
47 //Configuration of A/D realtime conversions and usb sending.
48 //timer
49 extern word timer_value;
50
51 //How many probes for mean
52 extern byte probes_for_mean;
53
54 //How many channels
55 extern byte ADCchannels;
56
57 //int led3count=0;
58
59 // F U N C T I O N S *****
60 void everyProbe () {
61     //4 next lines should be separated function ex: TimerReload()
62     INTCONbits.TMR0IF = 0;           //clear timer overload status bit
63     strataT = ReadTimer0 ();         //how many time expire when the timer
64                                     //was overloaded
```



```
65     WriteTimer0(70+strataT-timer_value);    //set up new value of timer with
66         //correction
67
68     //if(strataT > 1120) {++led3count; if (led3count==5)
69     //{mLED_3_Toggle(); led3count=0;}}
70     if (strataT > 1120) {mLED_3_Toggle()};
71
72     //add result of conversion and increment 'probka' variable
73
74     switch (ADCchannels) {
75         case 1 :
76             voltage[0]._dword = voltage[0]._dword + ReadADC();
77             probka++;
78         break;
79         case 2 :
80             //two channels
81             switch (ADCchannel) {
82                 case 1:
83                     voltage[0]._dword = voltage[0]._dword
84                     + ReadADC();
85                     SetChanADC(ADC_CH1);
86                     ADCchannel = 2;
87                 break;
88                 case 2:
89                     voltage[1]._dword = voltage[1]._dword
90                     + ReadADC();
91                     SetChanADC(ADC_CH0);
92                     ADCchannel = 1;
93                     probka++;
94
95                 break;
96             }
97         break;
98     }
99
100     //start conversion
101     ConvertADC();
102
103     //every 20 conversions send over the USB results
104     if (probka == probes_for_mean) {
105         everySend();
106     } //if (probka == 20) — time to send to USB
107 } // void everyProbe()
108
109 void everySend() {
110     switch (ADCchannels){
111         case 1:
112             //calculate what I should to send
113             voltage[0]._dword = voltage[0]._dword /
114             ( probes_for_mean / 5 );
```

```
115
116         //create string to send
117         bufor[1]= voltage[0].byte1;    //high byte
118         bufor[2]= voltage[0].byte0;    //low byte
119         bufor[3]= 0-bufor[0]-bufor[1]-bufor[2];
120             //calculate supplement to zero ERROR 1
121
122         //reset values of conversion numbers and result
123         probka = 0;
124         voltage[0]._dword = 0;
125
126
127
128         //if microcontroller have some free time send the string
129         if (mUSBUSARTIsTxTrfReady ())
130         {
131             mUSBUSARTTxRam((byte*) bufor ,4);
132         }
133     break;
134     case 2:
135         //calculate what I should to send
136         voltage[0]._dword = voltage[0]._dword /
137             ( probes_for_mean / 5 );
138         voltage[0]._dword = voltage[0]._dword /
139             ( probes_for_mean / 5 );
140
141         //create string to send
142         bufor[1]= voltage[0].byte1; //high byte
143         bufor[2]= voltage[0].byte0; //low byte
144         bufor[3]= voltage[1].byte1; //high byte
145         bufor[4]= voltage[1].byte0; //low byte
146         bufor[5]= 0-bufor[0]-bufor[1]-bufor[2]-bufor[3]-bufor[4];
147             //calculate supplement to zero
148
149         //reset values of conversion numbers and result
150         probka = 0;
151         voltage[0]._dword = 0;
152         voltage[1]._dword = 0;
153
154
155
156         //if microcontroller have some free time send the string
157         if (mUSBUSARTIsTxTrfReady ())
158         {
159             mUSBUSARTTxRam((byte*) bufor ,6);
160         }
161     break;
162 } //sitch (ADCchannels)
163 } // void everySend()
```

B. Źródła programu zainstalowanego w komputerze

Program zainstalowany w komputerze ma postać skryptu „Matlaba„ w systemie „Linux„, z nieznacznymi zmianami można go również użyć w systemie Windows. Źródła „modulmatlab.m” znajdują się poniżej:

```
1 function pomiar = modulmatlab ()
2 %Jacek Kosek 13.11.2007–21.11.2007
3 %Program czyta z modulu czujnika do pomiaru cisnienia krwi
4 %STALE AUTORYTARNE:
5 global MINPIKPIK MAXIMUMDIFF;
6 MINPIKPIK = 14.5; %Minimalna roznica miedzy maksimum a minimum tak zeby zostala
7 %uznana za
8 %amplitude pik-pik
9 MAXIMUMDIFF = 7; %Maksymalna roznica miedzy poprzednim odczytem
10 %
11 %
12 %-----
13 %wlasny algorytm wszystkich maximow i minimow
14 global szukane maxima minima
15 szukane = 0; % (0–minimum 1–maximum) Zaczynamy szukac od minimum
16 maxima = []; % w tym wektorze przechowywane beda indeksy wszystkich maximow z wektora:
17 %pomiar. wartosci
18 minima = []; % w tym wektorze przechowywane beda indeksy wszystkich minimow z wektora:
19 %pomiar. wartosci
20
21
22 MINIMALVALUE = 100;
23
24 %Przymiarka do czujnika
25 %ustawienie transmisji
26 !stty -F /dev/ttyACM0 ignbrk ignpar -icrnl -ixon -ixoff -opost -isig -icanon -iexten -echo
27
28 %a teraz troche zmiennych
29 global button pomiar
30 button = 0;
31 state = 0;
32 exit = 0;
```

```
33  wyswietl=0;
34  pomiar.wartosci= nan(1,6000);
35  pomiar.index = 0;
36
37          pomiar.pikMin = [];
38          pomiar.pikMax = [];
39          pomiar.wyprostowana=nan(1,6001);
40          pomiar.systolic=0;
41          pomiar.diastolic=0;
42
43  %Przygotowanie Plotu z przyciskami
44  global Dfig Dsurowe Dobliczone Dmax Dlegend;
45  Dfig      =figure;
46  hold on;
47  Dmax      =plot(0,0,'r^');
48  Dsurowe   =plot(pomiar.wartosci,'b-');
49  Dobliczone=plot(pomiar.wartosci,'k-');
50  Dlegend   =legend('000_00','Location','NorthWest');
51  grid on;
52  hold off;
53
54  axis([0 6000 0 250]);
55  stop_button   = uicontrol('Style','Pushbutton','string','Stop','Position',...
56                        [80 20 180 20], 'Enable', 'on', 'Callback', @stop_action);
57  start_button  = uicontrol('Style','Pushbutton','string','Start','Position',...
58                        [280 20 180 20], 'Enable', 'on', 'Callback', @start_action);
59  analize_button = uicontrol('Style','Pushbutton','string','Koniec','Position',...
60                        [480 20 180 20], 'Enable', 'on', 'Callback', @koniec_action);
61  set(Dfig,'ToolBar','figure');
62
63  %Otwarcie plku modulu
64  czujnik = fopen('/dev/ttyACM0');
65
66
67  drawnow;
68  while exit==0
69      c = fread(czujnik, [1 1]);
70          switch state;
71              case 0
72                  if (c=='F')
73                      state = 1;
74                  end
75                  %wyswietlanie c
76                  wyswietl = wyswietl + 1;
77                  if wyswietl==20
78                      %Reakcja na przycisk koniec
79                      if button == 3
80                          exit = 1;
81                      end
82                      wyswietl=0;
```

```
83         drawnow;
84     end
85     case 1
86         firstChar = c;
87         state = 2;
88     case 2
89         lastChar = c;
90         state = 3;
91     case 3
92         if ( (firstChar + lastChar + 'F' + c) ~= 0 )
93             punkt = firstChar * 256 + lastChar;
94             if (punkt > MINIMALVALUE) && (button ~= 2)
95                 dodajPoint(punkt);
96         end
97     end
98     state = 0;
99 end
100 end
101 fclose(czujnik);
102
103
104 function start_action(src, evt)
105     button = 1;
106 end
107
108 function stop_action(src, evt)
109     button = 2;
110 end
111
112 function koniec_action(src, evt)
113     button = 3;
114 end
115
116 function dodajPoint(punkt)
117     pomiar.index = pomiar.index + 1;
118     if (pomiar.index >= 6000) || (button==1)
119         button=0;
120         pomiar.index=1;
121         pomiar.wartosci= nan(1,6000);
122         szukane=0; % (0—minimum 1—maximum) Zaczynamy szukac od minimum
123         maxima=[]; % w tym wektorze przechowywane beda indeksy wszystkich
124             %maksimow z wektora: pomiar.wartosci
125         minima=[]; % w tym wektorze przechowywane beda indeksy wszystkich
126             %minimow z wektora: pomiar.wartosci
127         pomiar.pikMin=[];
128         pomiar.pikMax=[];
129         pomiar.wyprostowana=nan(1,6000);
130         pomiar.systolic=0;
131         pomiar.diastolic=0;
132         set(Dobliczone, 'YData', pomiar.wyprostowana);
```

```

133         set (Dmax, 'YData', 0);
134         set (Dmax, 'XData', 0);
135         set (Dmax, 'DisplayName', '000_00');
136     drawnow;
137
138     end
139     %nowa charakterystyka
140     pomiar.wartosci(pomiar.index)=punkt*0.0473+6.352-30;
141     %stara charakterystyka
142     %pomiar.wartosci(pomiar.index)=punkt*0.03630298544288+7.500616827042-35;
143     set (Dsurowe, 'YData', pomiar.wartosci);
144     i=pomiar.index;
145     if i>2
146         if (szukane==1) %jesli szukamy maximum
147             if pomiar.wartosci(i)<pomiar.wartosci(i-1)
148                 %poprzedni punkt to bylo maximum
149                 szukane=0; %od teraz bedziemy szukac minimum
150                 maxima=[maxima i-1]; %dodaj wartosc poprzedniego indeksu do
151                     %macierzy maksima
152
153                 %jesli amplituda miedzy obecnym maximum i ostatnim minimum jest
154                 %wieksza od MINPIKPIK
155                 if (pomiar.wartosci(i-1)...
156                     -pomiar.wartosci(minima(numel(minima)))) >= MINPIKPIK
157                     %amplituda miedzy obecnym maximum i ostatnim minimum
158                     %jest wieksza od MINPIKPIK
159                     %Zajmij sie analiza poprzedniego pik
160                     analiza_jednego_piku(i);
161                 end%if znaleziono pik-pik
162             end %koniec szukania maksimum lokalnego
163         else %jesli szukamy minimum
164             if pomiar.wartosci(i)>pomiar.wartosci(i-1)
165                 %poprzedni punkt to bylo minimum
166                 szukane=1; %od teraz bedziemy szukac maksimum
167                 minima=[maxima i-1]; %dodaj wartosc poprzedniego indeksu do
168                     %macierzy minima
169             end
170         end %koniec szukania minimum lokalnego
171     end
172 end
173
174
175 %-----
176 function analiza_jednego_piku(i)
177 %global MINPIKPIK MAXIMUMDIFF;
178
179 pomiar.pikMax = [pomiar.pikMax i-1]; %dodaj maksimum pik-pik do macierzy wyjsciowej
180 ilosc = numel(pomiar.pikMax); %dla wygody w tej zmiennej przechowuje rozmiar
181 %macierzy maksimum
182 %Teraz poszukujemy minimum znajdujacego sie po miedzy obecnym a poprzednim maksimum

```

```

183 if ilosc == 1
184     poprzednie_max = 1;           %jesli nie bylo poprzedniego max
185 else
186     poprzednie_max = pomiar.pikMax( ilosc - 1 ); %to szukaj od pocztku
187 end
188
189 szukane_min=pomiar.wartosci(porzecznie_max);
190 for j=[ poprzednie_max : i-1 ]
191     if pomiar.wartosci(j)<=szukane_min %czy nowe minimum
192         szukane_min=pomiar.wartosci(j); %nowe minimum
193         pomiar.pikMin( ilosc ) =j;%zapisanie nowego minimum
194     end %if nowe minimum
195 end %for j
196 %W tym momencie znamy pomiar.pikMin i pomiar.pikMax
197
198 %Teraz prostujemy nasz wykres
199 %obliczamy wspolczynnik nachylenia
200 if ilosc > 2
201     %pomiar.wyprostowana=zeros(pomiar.pikMin(1),1);
202
203     min_x      = pomiar.pikMin(ilosc);
204     min_x_pop  = pomiar.pikMin(ilosc - 1 );
205     min_y      = pomiar.wartosci(min_x);
206     min_y_pop  = pomiar.wartosci(min_x_pop);
207
208
209     wsp = ( min_y - min_y_pop ) / ( min_x -min_x_pop);
210     for j=[min_x_pop:min_x-1]
211         pomiar.wyprostowana(j) = pomiar.wartosci(j) - ((j-min_x_pop) * wsp ...
212             + min_y_pop);
213     end%od minimum do minimum
214     set(Dobliczone, 'YData', pomiar.wyprostowana + 10);
215
216     %Sprawdzenie czy to maksymalna amplituda
217     if pomiar.systolic-pomiar.diastolic < ...
218         pomiar.wyprostowana( pomiar.pikMax(ilosc-1) );
219         %Tak jest to maksymalna amplituda ale sprawdzic trzeba czy nie ma za
220         %duzych
221         %roznic w stosunku do ostatniego pomiaru
222         if (abs(min_y - min_y_pop) < MAXIMUMDIFF) && ...
223             (abs(pomiar.wartosci(pomiar.pikMax(ilosc)) ...
224                 -pomiar.wartosci(pomiar.pikMax(ilosc-1)))<MAXIMUMDIFF)
225             pomiar.systolic = pomiar.wartosci( pomiar.pikMax(ilosc-1) );
226             pomiar.diastolic = pomiar.systolic ...
227                 - pomiar.wyprostowana( pomiar.pikMax(ilosc-1) );
228             pomiar.indexMax = pomiar.pikMax(ilosc-1);
229             set(Dmax, 'YData', pomiar.systolic);
230                 set(Dmax, 'XData', pomiar.indexMax);
231             set(Dmax, 'DisplayName', [num2str(round(pomiar.systolic)) ...
232                 '\ ' num2str(round(pomiar.diastolic))]);

```

```
233     end
234   end
235 end
236 %
237
238 end
239 end
```


C. Źródła S-funkcji wykorzystanych w modelu czujnika

C.1. S-funkcja modelu membrany: „membrana_s.m”

S-funkcja modelu membrany składa się z jednego pliku:

```
1 function [sys,x0,str,ts] = sfuntmpl(t,x,u,flag)
2 %Jacek Kosek 22-03-2005 23-03-2005 (c)
3 %Funkcja do uzycia jako bloczek s-funkcji w matlabie
4 %uruchomic ja mozna przy pomocy biblioteki: komp.mdl
5 switch flag,
6
7     case 0,
8         [sys,x0,str,ts]= mdlInitializeSizes;
9
10    case 1,
11        sys=mdlDerivatives(t,x,u);
12
13    case 2,
14        sys=mdlUpdate(t,x,u);
15
16    case 3,
17        sys=mdlOutputs(t,x,u);
18
19    case 4,
20        sys=mdlGetTimeOfNextVarHit(t,x,u);
21
22    case 9,
23        sys=mdlTerminate(t,x,u);
24
25    otherwise
26        error(['Unhandled_flag_=',num2str(flag)]);
27
28 end
29
30 %=====
```

```

31
32 function [sys,x0,str,ts]=mdlInitializeSizes
33
34 sizes = simsizes;
35
36 sizes.NumContStates = 2;
37 sizes.NumDiscStates = 0;
38 sizes.NumOutputs = 1;
39 sizes.NumInputs = 1;
40 sizes.DirFeedthrough = 1;
41 sizes.NumSampleTimes = 1; % at least one sample time is needed
42
43 sys = simsizes(sizes);
44
45 x0 = [0 0];
46
47 str = [];
48
49 ts = [0 0];
50
51 %=====
52 function sys=mdlDerivatives(t,x,u)
53 zeta = 0.2;
54 T = 230*1e-6;
55 if u>=0
56     k=11e-6*log(1+u*.06);
57 else
58     k=-55e-6*log(1-u*.01);
59 end
60 dx1=(k-2*zeta*T*x(1)-x(2))/T^2;
61 dx2=x(1);
62 sys = [dx1; dx2];
63 %=====
64 function sys=mdlUpdate(t,x,u)
65
66 sys = [];
67
68
69 %=====
70 function sys=mdlOutputs(t,x,u)
71
72 sys = x(2);
73
74 %=====
75 function sys=mdlGetTimeOfNextVarHit(t,x,u)
76
77 sampleTime = 1; % Example, set the next hit to be one second later.
78 sys = t + sampleTime;
79
80 %=====

```

```

81 function sys=mdlTerminate(t,x,u)
82
83 sys = [];

```

C.2. S-funkcja modelu wzmacniacza: „wzmacniacz_s.m”

S-funkcja modelu wzmacniacza wywołuje kilka funkcji:

„C = cfun(sred, T)” funkcja zwracająca pojemność pneumatyczną a pobierając średnicę przewodu i Temperaturę,

„kapil = kapilfun(p1, p2, T, typ)” funkcja zwracająca strumień masy przepływający przez kapilarę pobierając ciśnienie przed i za kapilarą, temperaturę oraz długość dławika,

„przew = dyszprzyfun(x,p2,p0,dysza,T)” funkcja zwracająca strumień masy przepływający przez element dysza-przysłona pobierając uchylenie dyszy, ciśnienie przed i za dyszą, średnicę dyszy oraz temperaturę.

Natomiast funkcja „przew = dyszprzyfun(x,p2,p0,dysza,T)” wywołuje:

„fun_aprok_alfa3.m”

```

1 function [sys,x0,str,ts] = sfuntmpl(t,x,u,flag,p0,p1,typ,dysza,T,sred,p00s)
2 %Jacek Kosek 18-11-2005 19-11-2005 (c)
3 %Funkcja do uzycia jako bloczek s-funkcji w matlabie
4 %uruchomic ja mozna przy pomocy biblioteki: komp.mdl
5 switch flag,
6
7 case 0,
8     [sys,x0,str,ts]=mdlInitializeSizes(p00s);
9
10 case 1,
11     sys=mdlDerivatives(t,x,u,p0,p1,typ,dysza,T,sred);
12
13 case 2,
14     sys=mdlUpdate(t,x,u);
15
16 case 3,
17     sys=mdlOutputs(t,x,u,p0);
18
19 case 4,
20     sys=mdlGetTimeOfNextVarHit(t,x,u);
21

```

```

22 case 9,
23     sys=mdlTerminate(t,x,u);
24
25 otherwise
26     error(['Unhandled_flag_', num2str(flag)]);
27
28 end
29
30 %=====
31
32 function [sys,x0,str,ts]=mdlInitializeSizes(p00s)
33
34 sizes = simsizes;
35
36 sizes.NumContStates = 1;
37 sizes.NumDiscStates = 0;
38 sizes.NumOutputs = 1;
39 sizes.NumInputs = 1;
40 sizes.DirFeedthrough = 1;
41 sizes.NumSampleTimes = 1; % at least one sample time is needed
42
43 sys = simsizes(sizes);
44
45 x0 = [p00s];
46
47 str = [];
48
49 ts = [0 0];
50
51 %=====
52 function sys=mdlDerivatives(t,p2,x,p0,p1,typ,dysza,T,sred)
53
54 C = cfun(sred,T);
55 if x<0 x=0; end;
56 kapil=kapilfun(p1,p2,T,typ);
57 %przew=przewdpfun(x,p2,p0,dysza,T,sred)
58 przew=dyszprzfun(x,p2,p0,dysza,T);
59 dp2dt=(kapil-przew)/C;
60
61 sys = dp2dt;
62
63 %=====
64 function sys=mdlUpdate(t,x,u)
65
66 sys = [];
67
68
69 %=====
70 function sys=mdlOutputs(t,x,u,p0)
71

```

```

72 sys = x-p0;
73
74 %=====
75 function sys=mdlGetTimeOfNextVarHit(t,x,u)
76
77 sampleTime = 1; % Example, set the next hit to be one second later.
78 sys = t + sampleTime;
79
80 %=====
81 function sys=mdlTerminate(t,x,u)
82
83 sys = [];

```

C.3. Funkcja licząca pojemność pneumatyczną: „cfun.m”

```

1 function C = cfun(d3,T)
2 %Jacek Kosek 24-11-2005 24-11-2005 (c)
3 %Funkcja zwraca pojemnosc wzmacniacza
4 %
5 % C = cfun(d3)
6 %
7 %dane zwracane:
8 %C - Pojemnosc wzmacniacza
9 %dane wejsciowe:
10 %d3 - srednica przewodu zasilajacego
11
12 Vdod = pi * 2.5e-3^2 * 150e-3 /4;
13
14 C= ( Vdod + pi * d3^2 * 600e-3 / 4 )/ (287 * T);

```

C.4. Funkcja licząca strumień kapilary: „kapilfun.m”

```

1 function qm = kapilfun( p1, p2, T, typ);
2 %Jacek Kosek 28-9-2005 19-11-2005(c)
3 %Funkcja zwraca strumien masowy przeplywajacy przez kapilare.
4 % strumien wyznaczany jest na podstawie cisnien za i przed kapilara
5 % oraz temperatury, zaszyte sa 4 kapilary.
6 %
7 % qm = kapilfun( p1, p2, T, typ)
8 %
9 %dane zwracane:
10 % qm - strumien masowy w kg/s
11 %dane wejsciowe:
12 % p1 - cisnienie przed kapilara w Pa

```

```

13 % p2 - cisnienie za kapilara w Pa
14 % T - srednia temperatura gazu w K
15 % typ - typ kapilary:
16 %          1 - l = 5mm i zeta = 1.7
17 %          2 - l = 10mm i zeta = 1.6
18 %          3 - l = 20mm i zeta = 2.1
19 %          4 - l = 35mm i zeta = 1.2
20
21 %DANE: =====
22 switch typ;
23     case 1;
24         l = 5e-3;
25         zeta = 1.7;
26     case 2;
27         l = 10e-3;
28         zeta = 1.6;
29     case 3;
30         l = 20e-3;
31         zeta = 2.1;
32     case 4;
33         l = 35e-3;
34         zeta = 1.2;
35     otherwise;
36         disp('Nieznany_Dlawik. ');
37         qm = -1;
38         return;
39 end
40
41 %Indywidualna stala gazowa dla powietrza
42 R = 287; %Nm/(kgK)
43
44 %srednica kapilary: zmierzona za pomoca przeplywu cieczy
45 d = .22e-3; %m
46
47 %A to obliczenie (ze wzoru przyblizonego) lepkości dynamicznej
48 mu = 17.1667e-6+(T-273)*5e-8;% Ns/m^2
49
50 Re_new = 10; %zakladamy do iteracji Re
51 Re=0;
52
53 %Warunki przeplywu
54 if p1<0    p1=0;end
55 if p2<0    p2=0;end
56 if p1<=p2    qm=0;return;end
57
58
59 while abs(Re_new - Re)/Re_new >= 1e-10 %Obliczanie iteracyjne
60     Re = Re_new;
61
62

```

```

63 %dla laminarnego straty liniowe wynosza:
64 lamb = 64/Re;
65 %Obliczenie qm z ogolnego wzoru:
66 qmm = pi*d^2/4*sqrt( (p1^2-p2^2) /R/T/(lamb*l/d+zeta+2*log(p1/p2)) );
67 Re_new = 4*qmm/pi/d/mu; %nowa liczba Re
68 end; %while Re
69 qm = qmm;

```

C.5. Funkcja licząca strumień elementu dysza-przysłona: „dyszprzyfun.m”

```

1 function qm = dyszprzyfun(x, p2, p0, dysza, T);
2 %Jacek Kosek 27-10-2005 19-11-2005(c)
3 %Funkcja zwraca strumien masowy przeplywajacy przez czlon dysza-przyslona.
4 % strumien wyznaczany jest na podstawie cisnien za i przed dysza-przyslona
5 % oraz srednicy dyszy, odleglosci przyslony od dyszy i temperatury.
6 %
7 % qm = dyszprzyfun(x, p2, p0, dysza, T)
8 %
9 %dane zwracane:
10 % qm - strumien masowy w kg/s
11 %dane wejsciowe:
12 % x - odleglosc przyslony od dyszy w m
13 % p2 - cisnienie przed dysza-przyslona w Pa
14 % p0 - cisnienie za dysza-przyslona w Pa
15 % dysza- srednica dyszy w m
16 % T - srednia temperatura gazu w K
17
18 %DANE: =====
19
20
21 %Indywidualna stala gazowa dla powietrza
22 R = 287; %Nm/(kgK)
23
24 %A to obliczenie (ze wzoru przyblizonego) lepkości dynamicznej
25 mu = 17.1667e-6+(T-273)*5e-8;% Ns/m^2
26
27 %obliczenia iteracyjne gdyz alfa zalezy od Re a Re od qm
28 Re_new = 10;
29 Re = 0;
30
31 %Warunki przeplywu
32 if p2<=p0 qm=0;return ;end
33 if x<=0 qm=0;return ;end
34 if dysza <= 0 qm=0;return ;end
35 if T <= 0 qm=0;return ;end

```

```

36 if p2 <= 0 qm=0;return;end
37 if p0 <= 0 qm=0;return;end
38
39
40 %Rekurencyjne liczenie Re
41 while abs(Re_new-Re)/Re_new >= 1e-10;
42     Re=Re_new;
43     %aproxymacja alfy (wspolczynnika przeplywu
44
45     wsp=[1.7435    0.0437    -1.6642    0.3765];
46     alfa = fun_aprok_alfa3(wsp, x/dysza, Re);
47     %obliczenie qm kapilary
48
49     qm= alfa*x*pi*dysza*sqrt(2*p2*(p2-p0)/R/T);
50
51     %nowe Re (do alfy)
52     Re_new = 2 * qm / pi / dysza / mu;
53 end;

```

C.6. Funkcja aproksymująca współczynnik α : „fun_aprok_alfa3.m”

```

1 function alfa = fun_aprok_alfa3(wsp_aprok_alfa, x_do_d, Re);
2 %Jacek Kosek 27-10-2005 19-11-2005(c)
3 %Funkcja dwoch zmiennych aproksymujaca alfe
4 %
5 % alfa = fun_aprok_alfa3(wsp_aprok_alfa, x_do_d, Re)
6 %
7 %dane zwracane:
8 % alfa          - wspolczynniki strat miejscowych
9 %dane wejsciowe:
10 % wsp_aprok_alfa - wspolczynniki aproksymacji alfy
11 % x_do_d         - pierwsza zmienna x/d
12 % Re            - druga zmienna liczba Re
13
14 a = wsp_aprok_alfa(1);
15 b = wsp_aprok_alfa(2);
16 c = wsp_aprok_alfa(3);
17 d = wsp_aprok_alfa(4);
18
19 alfa = (a.*x_do_d+b).*Re.^(c.*x_do_d+d);
20 if alfa == inf alfa = 0;end

```


D. Skrypty wykorzystane przy doborze parametrów czujnika

D.1. Skrypt uruchamiający symulacje: „eksperymenty.m”

```
1 %Jacek Kosek (c)05.02.2007–06.02.2007
2 %Ten skrypt uruchamia eksperymenty na modelu
3
4 cisnienie_p1=[11013 11017 11298 11866 12597 13293 13800 13995 13854 13569 13275 ...
5              13006 12764 12574 12447 12324 12173 12012 11818 11702 11641 11622 ...
6              11673 11749 11829 11883 11894 11863 11797 11707 11616 11525 11434 ...
7              11346 11286 11225 11170 11122 11061 11027 11004];
8
9 %Wgrywam sygnał tetna
10 load tetno_simulink %dwie zmienne: tet i czas
11
12 %ustawiam parametry czujnika
13 p_rozkur=11000;%pa
14 p_pikpik= 3000;%pa
15 x_0_vector=[50e-6 10e-6]; %m
16 p_ot=1e5;%pa
17 p_z_vector=[25000 30000 40000 50000];%pa
18 l_dl_vector=[5e-3 10e-3 15e-3 20e-3];%m
19 sr_dyszy_vector=[.7]*1e-3;%m
20 sr_przew_vector=[1 1.5 2 2.5]*1e-3;%m
21
22 for m=1:size(x_0_vector,2);
23     for l=1:size(sr_dyszy_vector,2);
24         for k=1:size(p_z_vector,2);
25             for j=1:size(l_dl_vector,2);
26                 for i=1:size(sr_przew_vector,2);
27                     %wybieranie danych do konkretnego eksperymentu
28                     x_0=x_0_vector(m);
29                     sr_dyszy=sr_dyszy_vector(l);
30                     p_z=p_z_vector(k);
31                     l_dl=l_dl_vector(j);
32                     sr_przew=sr_przew_vector(i);
33
34                     %symulacja
35                     sim czujnik_tetno;
```

```

36
37     %porzadkowanie danych wyjsciowych
38     Wykresy(i,j,k,l,m)=ScopeData;
39     for ii=1:41; cisnienie_p2(ii,i,j,k,l,m)=simout(ii+39);end
40     [size(sr_przew_vector,2) size(l_dl_vector,2) size(p_z_vector,2) ...
41      size(sr_dyszy_vector,2) size(x_0_vector,2);i j k l m]
42     end; %for i
43     end; %for j
44     end; %for k
45     end;%for l
46 end;%for m
47 save eksperymenty_dane Wykresy cisnienie_p2;
48
49 %porzadkowanie danych wyjsciowych
50 %ScopeData
51 %for i = 1:41; cisnienie_p2(i)=simout(i+39);end
52 %plot(cisnienie_p1 , cisnienie_p2);
53 %plot([0:1:40], abs(fft(cisnienie_p1-mean(cisnienie_p1))),[0:1:40], ...
54 %      abs(fft(cisnienie_p2-mean(cisnienie_p2))))
55 %angle

```

D.2. Skrypt obliczający kryterium: „bledy_2.m”

```

1 % Jacek Kosek 12.02.2007–12.02.2007
2 cisnienie_p1=[11013 11017 11298 11866 12597 13293 ...
3              13800 13995 13854 13569 13275 13006 12764 12574 ...
4              12447 12324 12173 12012 11818 11702 11641 ...
5              11622 11673 11749 11829 11883 11894 11863 ...
6              11797 11707 11616 11525 11434 11346 11286 ...
7              11225 11170 11122 11061 11027 11004];
8 wyzerowany_p1_nieunormowany = ( cisnienie_p1-mean(cisnienie_p1) );
9 maximum_wyzerowanego=max(wyzerowany_p1_nieunormowany);
10 wyzerowany_p1 = wyzerowany_p1_nieunormowany / max(wyzerowany_p1_nieunormowany);
11
12 %ustawiam parametry czujnika
13 p_rozkur=11000;%pa
14 p_pikpik= 3000;%pa
15 x_0_vector=[50e-6 10e-6]; %m
16 p_ot=1e5;%pa
17 p_z_vector=[25000 30000 40000 5000];%pa
18 l_dl_vector=[5e-3 10e-3 15e-3 20e-3];%m
19 sr_dyszy_vector=[.7]*1e-3;%m
20 sr_przew_vector=[1 1.5 2 2.5]*1e-3;%m
21
22
23 for m=1:size(x_0_vector,2);
24     for l=1:size(sr_dyszy_vector,2);

```

```
25     for k=1:size(p_z_vector,2);
26         for j=1:size(l_dl_vector,2);
27             for i=1:size(sr_przew_vector,2);
28
29                 wyzerowany_p2_nieunormowany=cisnienie_p2(:,i,j,k,l,m)-...
30                                     mean(cisnienie_p2(:,i,j,k,l,m));
31                 wyzerowany_p2=wyzerowany_p2_nieunormowany/max(wyzerowany_p2_nieunormowany);
32
33                 odchylka=wyzerowany_p1'-wyzerowany_p2;
34             for n=1:size(odchylka,2)
35                 odchylka_n(n)=abs(odchylka(n));
36             end
37
38             %Kryterium Maximum
39             blad_max(i,j,k,l,m)=max(odchylka);
40
41             %Kryterium calkowe
42             calka=0;
43             for n=1:size(odchylka,2);
44                 calka=calka+abs(odchylka(n));
45             end;
46             blad_calka(i,j,k,l,m)=calka*maximum_wyzerowanego/1000/size(odchylka,2);
47             end; %for i
48         end; %for j
49     end; %for k
50 end;%for l
51 end;%for m
52 blad_max=blad_max*maximum_wyzerowanego/1000
```

Bibliografia

- [1] Kanjilal P.P., i inni. A non invasive approach to continous measurement of blood pressure. *Measurement in Clinical Medicine, 3rd International Conference*, str. 67–71, Edynburg, 1986.
- [2] karta katalogowa f-my COLIN, Japonia. *Non-invasive Real-time Arterial Tonometry*.
- [3] Terry S., Low T., Ablow C.M. Silicon pressure transducer arrays for blood-pressure measurement. Raport instytutowy, ICSensors – Custom Product Papers and Briefs, str. 4.37-4.46., 2002.
- [4] Werszko M., Kosek J. Problemy bezinwazyjnych pomiarów fali tętniczej krwi / w: Modelowanie i pomiary w medycynie. *Materiały VII sympozjum Modelowanie i Pomiary w Medycynie*, str. 101–106, Krynica 2005.
- [5] Ng K.-G., i inni. Progress on the development of the mediwatch ambulatory blood pressure monitor and related devices. *Blood Pressure Monitoring*, (9):str. 149–165, 2004.
- [6] O'Rourke M., i inni. Pulse wave analysis. *Blackwell Science Ltd J Clin Pharmacol*, (51):str. 507–522, 2001.
- [7] Dąbrowski A. *Inne graficzne metody badania układu krążenia*, wolumen Nauka o chorobach wewnętrznych pod redakcją W. Orłowskiego, rozdział 10. Państwowy Zakład Wydawnictw Lekarskich, Warszawa, 1988.
- [8] Chen-Huan Chen, Erez Nevo, Barry Fetics, Peter H. Pak, Frank C.P. Yin, W. Lowell Maughan, David A. Kass. Estimation of central aortic pressure waveform by mathematical transformation of radial tonometry pressure. *Circulation*, (95):1827–1836, 1997.
- [9] James D. Cameron, Barry P. McGrath, Anthony M. Dart. Use of radial artery application tonometry and a generalized function to determine aortic pressure augmentation in subjects with treated hypertension. *Journal of the American College of Cardiology*, (32):1214–1220, 1998.
- [10] Werszko M., Werszko R. Zasada kompensacji ciśnień w pomiarach medycznych. *IV Sympozjum: Modelowanie i pomiary w medycynie*, str. 227–234, Krynica, 2002.
- [11] Werszko M., Werszko R. Wrocławskie czujniki ciśnienia wewnątrzczaszkowego. *Pomiary Automatyka Kontrola*, (7):str. 5–7, 1999.

-
- [12] Kościelny W., Woźniak C. Modele charakterystyk przepływowych oporów pneumatycznych. *IX Ogólnopolska Konferencja Pneuma*, str. 73–82, Sklarska Poręba, 1995.
- [13] Werszko M., Werszko R. Generator sinusoidalnych sygnałów przesunięciowych i ciśnieniowych. *Pneumatyka*, (4):46–47, 2000.
- [14] Chorowski B., Werszko M. *Mechaniczne urządzenia automatyki*. Wydawnictwa Naukowo-Techniczne, 1990.
- [15] Kosek J. Model matematyczny wzmacniacza typu dysza-przysłona pneumatycznego czujnika ciśnienia krwi. *Pomiary Automatyka Kontrola*, (5bis):67–71, 2006.
- [16] Jędrusyna A., Kubowicz Z. Pneumatyczny generator dowolnych sygnałów ciśnieniowych. *Pneumatyka*, (2):33–35, 2005.
- [17] Kosek J. Model matematyczny pneumatycznego czujnika niskich ciśnień. *Pneumatyka*, (1):31–33, 2007.
- [18] Kosek J. Dobór parametrów konstrukcyjnych pneumatycznego czujnika ciśnienia krwi. *Pomiary Automatyka Kontrola*, (9bis):609–612, 2007.
- [19] O'Brien E., i inni. Working group on blood pressure monitoring of the european society of hypertension international protocol for validation of blood pressure measuring devices in adults. *Blood Pressure Monitoring*, 7(1), 2002.
- [20] Weiss B. M., Spahn D. R., Rahmig H., Rohling R., Pasch T. Radial artery tonometry: moderately accurate but unpredictable technique of continuous non-invasive arterial pressure measurement. *British Journal of Anaesthesia*, (76):405–411, 1996.

Spis rysunków

1.1.	Przykładowa fala tętna krwi w tętnicy promieniowej pacjenta z ciśnieniem skurczowym i rozkurczowym 120/80.	8
1.2.	Podział przyrządów do pomiarów fali tętna krwi.	8
1.3.	Uproszczony schemat przyrządu z matrycą piezorezystorów.	9
1.4.	Uproszczony schemat przyrządu z fotopletyzmoграфem.	10
3.1.	Widok przyrządu i nadgarstka.	14
3.2.	Widok ekranu komputera.	15
3.3.	Schemat czujnika i badanej tętnicy.	17
3.4.	Ilustracja zasady kompensacji ciśnień.	18
3.5.	Schemat blokowy modułu zasilającego.	19
3.6.	Schemat zasilania sprężarki.	20
3.7.	Schemat modułu akwizycji i transmisji USB.	21
3.8.	Płytką modułu akwizycji i transmisji USB.	22
3.9.	Ramki wysyłane z modułu do komputera.	24
3.10.	Graf przedstawiający zachowanie pętli głównej programu.	26
3.11.	Algorytm działania funkcji: „analiza_punktu”.	28
4.1.	Schemat strukturalny czujnika do pomiaru ciśnienia krwi ze wzmacniaczem typu dysza-przysłona, schemat wyjściowy.	31
4.2.	Schemat strukturalny czujnika do pomiaru ciśnienia krwi ze wzmacniaczem typu dysza-przysłona, schemat po przekształceniu.	32
4.3.	Schemat wzmacniacza typu dysza-przysłona.	34
4.4.	Schemat kapilary.	35
4.5.	Porównanie modelu matematycznego kapilary z wynikami doświadczeń.	37
4.6.	Schemat elementu dysza-przysłona z zaznaczonym kierunkiem przepływu powietrza.	38
4.7.	Wykres zależności współczynnika α i jego aproksymacji $\hat{\alpha}$ od Re i x/d_2	39

4.8.	Wykres zależności ciśnienia p_2 od otwarcia dyszy x dla dwóch ciśnień zasilania p_z : 24 kPa, 50 kPa i dławika o długości $l = 20$ mm	40
4.9.	Przykładowe doświadczalne charakterystyki częstotliwościowe (wzmocnienie wzmacniacza sprowadzono do jedynki).	42
4.10.	Uproszczony schemat modelu dynamiki wzmacniacza typu dysza-przysłona.	43
4.11.	Zależność stałych czasowych wzmacniacza od średnicy przewodu zasilającego i ciśnienia zasilania dla dyszy $d_2 = 0,6$ mm i dławika o długości $l = 20$ mm.	44
4.12.	Charakterystyka statyczna membrany.	46
4.13.	Charakterystyka impulsowa membrany.	46
4.14.	Schemat modelu czujnika w Simulinku.	48
4.15.	Charakterystyka statyczna czujnika.	49
4.16.	Przykładowe przebiegi sygnałów mierzonych p_1 i wyjściowych p_2 czujnika.	50
5.1.	Fala tętna krwi	53
5.2.	Obszary poprawnej pracy czujnika	57
6.1.	Schemat stanowiska do badania czujnika na modelu fizycznym tętnicy	59
6.2.	Charakterystyka statyczna czujnika przy badaniu na sztucznej tętnicy	60
6.3.	Przebiegi fali tętna podczas badań na sztucznej tętnicy	62
6.4.	Przebiegi fali tętna w tętnicy szyjnej [7].	64
6.5.	Fala tętna krwi w aorcie [6].	65
6.6.	Przykład zmierzonej (p_2) i skorygowanej (p_1^{obl}) fali tętna krwi.	67
6.7.	Pletyzmograf firmy „BioMedix inc.”	67
6.8.	Zestawienie przebiegu sygnału z pletyzmografu (linia przerywana) oraz przefiltrowanego sygnału z badanego przyrządu (linia ciągła), a) pacjent w wieku 30 lat, b) pacjent w wieku 60 lat	69

Spis tabel

3.1.	Zestawienie rozkazów wysyłanych przez komputer do modułu.	23
3.2.	Zestawienie trybu pracy interfejsu z ilością próbek służących do uśrednienia jednego punktu pomiarowego.	24
5.1.	Wartości maksymalnych różnic $p_1 - p_1^{obl}$ w kPa dla czujnika o parametrach: $d_2 = 0,7$ mm, $x_0 = 0,05$ mm	56