Metody numeryczne w elektrotechnice

Ćwiczenie 7 — Zastosowania wartości i wektorów własnych macierzy

Dariusz Borkowski

Wstęp

Na zajęciach rozwiążecie zadania, pokazujące trzy typowe (spośród wielu) zastosowania wartości i wektorów własnych. Zadania za 1, 2 i 3 punkty związane ze stopniem trudności.

1 Rozsprzęganie układu sprzężonych równań różniczkowych – 2 pkt

Filtr aktywny 2–rzędu o strukturze pokazanej na rysunku 1 może być opisany opisany transmitancją

$$G(s) = \frac{Y(s)}{U(s)} = \frac{1}{R_1 R_2 C_1 C_2 s^2 + C_2(R_1 + R_2)s + 1} = \frac{1}{q_2 s^2 + q_1 s + 1} \tag{1}$$

gdzie R_1, R_2, C_1, C_2 to wartości elementów na schemacie, $Y(s)$ jest transformatą Laplace'a napięcia wyjściowego, $X(s)$ jest transformatą Laplace'a napięcia wejściowego. Równanie to można przekształcić do postaci

$$Y(s)(q_2 s^2 + q_1 s + 1) = U(s) \tag{2}$$

a następnie przejść do dziedziny czasu, przenieść wszystko oprócz drugiej pochodnej $y(t)$ na prawą stronę, otrzymując równanie różniczkowe drugiego rzędu

$$\ddot{y} = \frac{1}{q_2} u - \frac{q_1}{q_2}\dot{y} - \frac{1}{q_2} y \tag{3}$$

Po podstawieniu za zmienne stanu

$$x_1 = y \qquad x_2 = \dot{y} \tag{4}$$

równanie to można zapisać w postaci macierzowego układu dwóch równań pierwszego rzędu, zwanych równaniem stanu

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \tag{5}$$

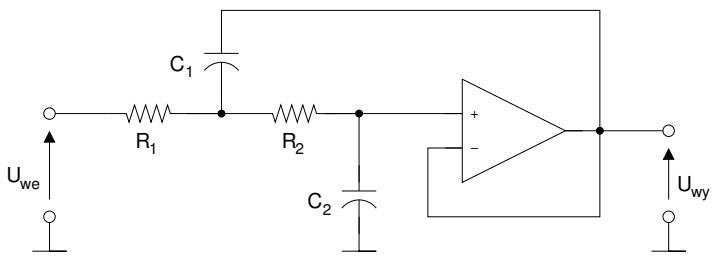
w których $\boldsymbol{x} = \begin{bmatrix} x_2 & x_1 \end{bmatrix}^T$ jest wektorem zmiennych stanu.

Taki układ równań można łatwo rozwiązać gdy macierz stanu $\boldsymbol{A}$ jest diagonalna. Jeśli zaś równania są ze sobą sprzężone, czyli pochodna jednej zmiennej stanu zależy również od wartości innych zmiennych stanu to macierz $\boldsymbol{A}$ nie jest diagonalna. Jeśli nie jest diagonalna to w większości przypadków można to zadanie rozwiązać przez diagonalizację macierzy $\boldsymbol{A}$ za pomoca jej wektorów własnych. Macierz $\boldsymbol{A}$ jest związana z macierzą diagonalną $\boldsymbol{D}$ zaleznościami

$$\boldsymbol{D} = \boldsymbol{U}^{-1}\boldsymbol{A}\boldsymbol{U} \qquad \boldsymbol{A} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^{-1} \tag{6}$$

gdzie $\boldsymbol{U}$ jest macierzą, której kolumny zawierają kolejne wektory własne macierzy $\boldsymbol{A}$. Mając zatem jednorodny układ równań sprzężonych

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} \tag{7}$$



Rysunek 1: Schemat filtra aktywnego 2 rzędu.

można go zapisać jako

$$\dot{\boldsymbol{x}} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^{-1}\boldsymbol{x} \tag{8}$$

i po podstawieniu za nowe zmienne stanu $\boldsymbol{w} = \boldsymbol{U}^{-1}\boldsymbol{x}$ i otrzymać układ równań rozsprzężonych

$$\dot{\boldsymbol{w}} = \boldsymbol{D}\boldsymbol{w} \tag{9}$$

Aby rozwiązać sprzężone, jednorodne równanie stanu z macierzą $\boldsymbol{A}$ o wymiarach $N \times N$

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} \tag{10}$$

(czyli takie, w którym nie ma iloczynu $\boldsymbol{B}\boldsymbol{u}$), należy wykonać następujące czynności.

1. Wyznaczyć wartości własne $\boldsymbol{\lambda} = [\ \lambda_1 \quad \lambda_2 \quad \dots \quad \lambda_N\ ]^T$ macierzy $\boldsymbol{A}$ i związane z nimi wektory własne w postaci macierzy $\boldsymbol{U}$ oraz obliczyć wyznacznik $\det(\boldsymbol{U})$. Jeśli wyznacznik jest zerem to znaczy, że macierz $\boldsymbol{A}$ nie jest diagonalizowalna i tą metodą nie da się tego zrobić. STOP.

2. Wyznaczyć wartości początkowe $\boldsymbol{w}(0)$ na podstawie wartości początkowych oryginalnych zmiennych stanu jako
$$\boldsymbol{w}(0) = \boldsymbol{U}^{-1}\boldsymbol{x}(0) \tag{11}$$

3. Następnie zdefiniować wektor nowych zmiennych stanu (tak jak $\boldsymbol{x}$ jest to wektor funkcji czasu o znanej postaci)
$$\boldsymbol{w}(t) = \begin{bmatrix} w_1(0)e^{\lambda_1 t} \\ w_2(0)e^{\lambda_2 t} \\ \vdots \\ w_N(0)e^{\lambda_N t} \end{bmatrix} \tag{12}$$

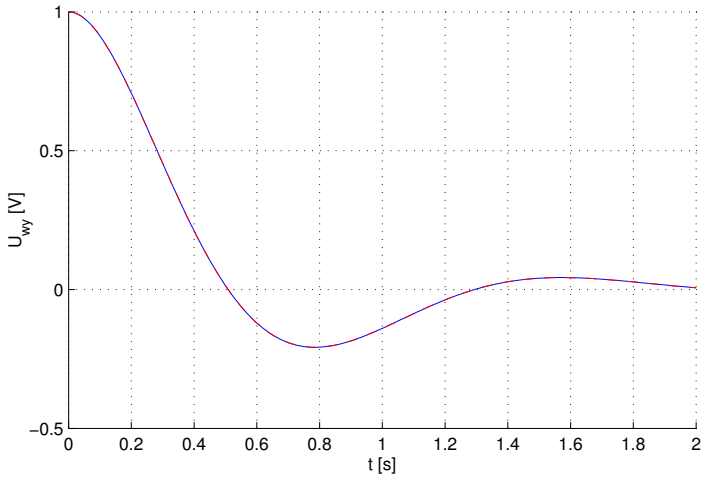
4. Ostatecznie przebieg zmiennych stanu $\boldsymbol{x}(t)$ od wartości początkowych $\boldsymbol{x}(0)$ obliczyć jako
$$\boldsymbol{x}(t) = \boldsymbol{U}\boldsymbol{w}(t) \tag{13}$$

Przyjmij wartości elementów schematu $R_1 = 200\ \mathrm{k\Omega}$, $R_2 = 100\ \mathrm{k\Omega}$, $C_1 = \frac{30}{8}\ \mu\mathrm{F}$, $C_2 = \frac{2}{3}\ \mu\mathrm{F}$. Wyprowadź na papierze[1] równania stanu dla transmitancji (1). Zauważ, że macierz $\boldsymbol{A}$ nie jest diagonalna. Wyznacz na papierze wartości własne macierzy $\boldsymbol{A}$ przez rozwiązanie równania charakterystycznego $\det(\boldsymbol{A} - \lambda\boldsymbol{I}) = 0$. Wyznaczone wartości własne są liczbami zespolonymi równym biegunom transmitancji (1). W przypadku zespolonych wartości własnych wyznaczenie wektorów własnych lepiej zostawić Matlabowi. Zarówno wektory i wartości własne wyznacza funkcja `eig`.

Korzystając z opisanej wyżej metody rozwiązywania układu równań różniczkowych sprzężonych rozwiąż wyprowadzone wcześniej równanie stanu. Ponieważ dysponujesz jedynie rozwiązaniem równania jednorodnego, możesz przyjąć, że na wyjściu filtra opisanego równaniem stanu istniał stan ustalony i napięcie równe $1\ \mathrm{V}$, a w chwili $t = 0$ zwarto wejście filtra, czyli podano zerowe napięcie. Oznacza to, że za warunki początkowe przyjmiesz $\dot{y}(0) = 0$ i $y(0) = 1$ czyli $\boldsymbol{x}(0) = [\ 0 \quad 1\ ]^T$. Wyrysuj przebieg napięcia wyjściowego $y(t)$, dla zadanych warunków początkowych i porównaj z wykresem uzyskanym funkcją `lsim`. W nowszych wersjach Matlaba funkcja `lsim` wymaga podania równań stanu w postaci obiektu `sys` tworzonego funkcją `ss`. Z kolei funkcja `ss` wymaga podania pełnego opisu równaniami stanu złożonego z czerech macierzy $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{D}$ (patrz `lab02.pdf`), przy czym $\boldsymbol{D}$ nie jest wspomnianą wyżej macierzą diagonalną. Konieczne jest więc wyprowadzenie macierzy $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{D}$, które są trywialne. $\boldsymbol{B}$ łączy wartość wejścia $u(t)$ z którymś z równań, $\boldsymbol{C}$ łączy wartość wyjścia $y(t)$ z konkretną zmienną stanu, a $\boldsymbol{D}$ jest zerem (patrz Miller vs. Ziobro). W starszych wersjach macierze $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{D}$ podaje się bezpośrednio jako argumenty funkcji `lsim`. Funkcja `lsim` przyjmuje również wektor czasu `t` jak i warunki początkowe `x0`.

Jeśli wszystko wykonałeś poprawnie, wykresy powinny nałożyć się na siebie jak na rysunku 2

[1] Przejście od opisu w postaci transmitancji do równań stanu można w Matlabie wykonać za pomocą funkcji `ss`, `tf` należących do Control System Toolbox



Rysunek 2: Przebieg napięcia na wyjściu filtra po zwarciu wejścia podczas stanu ustalonego.

2 Kompresja sygnału przez transformację Karhunena–Loevego – 2 pkt

Przyjmijmy, że budujemy system akwizycji danych. System ma pracować długo i archiwizować dane na dysku twardym. Rejestrowaną wielkością będzie wartość skuteczna napięcia sieci energetycznej niskiego napięcia, wyznaczaną droga filtracji dolnoprzepustowej. Można zatem rozważyć kompresję stratną tego sygnału, tak aby oszczędzić miejsce na dysku.

Dyskretnoczasowy sygnał $\boldsymbol{x}$, poddawany kompresji można przedstawić jako N–wymiarowy wektor $\boldsymbol{x} = [\ x_1 \quad x_2 \quad \ldots \quad x_N\ ]^T$. Jeśli dynamikę sygnału można opisać modelem autoregresji M–tego rzędu

$$x_n = a_1 x_{n-1} + a_2 x_{n-2} + \ldots + a_M x_{n-M} = \sum_{i=1}^{M} a_i x_{n-i} \tag{14}$$

wtedy wartość aktualnej próbki sygnału zależy jedynie od M poprzednich próbek tego sygnału. Możemy wtedy sygnał podzielić na M elementowe realizacje tego sygnału i na ich podstawie estymować macierz kowariancji sygnału

$$\boldsymbol{R}_x = \mathrm{E}\left[(\boldsymbol{x} - \bar{\boldsymbol{x}})(\boldsymbol{x} - \bar{\boldsymbol{x}})^{\mathrm{T}}\right] \tag{15}$$

gdzie $\bar{\boldsymbol{x}}$ jest estymatą wartości oczekiwanej całego sygnału $\boldsymbol{x}$, a $\mathrm{E}(\bullet)$ jest wartością oczekiwaną z $\bullet$.

Macierz $\boldsymbol{R}_x$ (rozmiaru $M \times M$), która jest symetryczna, dodatnio określona, posiada M ortogonalnych wektorów własnych[2] $\boldsymbol{v}_m$ skojarzonych z wartościami własnymi $\boldsymbol{\lambda_m}$, przy czym wartości własne są uporządkowane malejąco czyli

$$\lambda_1 \geqslant \lambda_2 \geqslant \ldots \geqslant \lambda_m \geqslant \ldots \geqslant \lambda_M \tag{16}$$

a wektory własne tworzą bazę przekształcenia ortogonalnego i są zapisywane jako kolumny macierzy przekształcenia

$$\boldsymbol{V} = [\ \boldsymbol{v}_1 \quad \boldsymbol{v}_2 \quad \ldots \quad \boldsymbol{v}_M\ ] \tag{17}$$

którą można wykorzystać do przekształcenia[3] sygnału $\boldsymbol{x}$ na nowy sygnał $\boldsymbol{y}$

$$\boldsymbol{y} = \boldsymbol{V}^T(\boldsymbol{x} - \bar{\boldsymbol{x}}) \tag{18}$$

znanego jako transformacja Karhunena–Loevego[4].

Aby zgodziły się wymiary macierzy przekształceniu poddawane są M–elementowe realizacje sygnału $\boldsymbol{x}$, a w wyniku otrzymuje się M–elementowe realizacje sygnału $\boldsymbol{y}$. Mimo, że sygnał $\boldsymbol{x}$ został przekształcony w inny sygnał, to ilość danych do zapisania pozostała niezmieniona. Powinieneś również zauważyć, że jest

[2] Analogia do wielomianów Grama, ortogonalnych na zbiorze węzłów.

[3] Ponieważ $\boldsymbol{V}$ jest ortogonalna to $\boldsymbol{V}^{-1} = \boldsymbol{V}^T$.

[4] Najczęściej spotykamy ten wzór w postaci bez wartości oczekiwanej czyli $\boldsymbol{y} = \boldsymbol{V}^T\boldsymbol{x}$. Wymaga to oczywiście założenia o zerowej wartości oczekiwanej kompresowanego sygnału dotyczącą całości rozważań. Wtedy we wzorach od (15) do (22) można pominąć $\bar{\boldsymbol{x}}$.

to aproksymacja, a w tym przypadku nawet interpolacja, oczywiście z bazą KL a nie jednomianów. Zmniejszyć ilość danych potrzebnych do opisu sygnału można usuwając z macierzy $\boldsymbol{V}$ wektory własne (kolumny) związane z najmniejszymi wartościami własnymi. Można tak zrobić, gdyż wektor własny związany z największą wartością własną określa składową (kierunek) o największej wariancji, a pozostałe przy podanym wyżej uporządkowaniu pokazują składowe o coraz mniejszej wariancji. Można przyjąć, że usunięcie składowych o małej wariancji nie wpłynie znacząco na kształt skompresowanego a następnie zrekonstruowanego sygnału. Zapiszmy pozostałe K kolumn macierzy $\boldsymbol{V}$ jako macierz prostokątną

$$\boldsymbol{U} = [\ \boldsymbol{v}_1 \quad \ldots \quad \boldsymbol{v}_K\ ] \qquad K \leqslant M \tag{19}$$

i otrzyamy nadokreślony układ równań, taki sam jak w aproksymacji (bo to zresztą jest aproksymacja). Układ taki rozwiązujemy

$$\boldsymbol{y} = (\boldsymbol{U}^T\boldsymbol{U})^{-1}\boldsymbol{U}^T(\boldsymbol{x} - \bar{\boldsymbol{x}}) \tag{20}$$

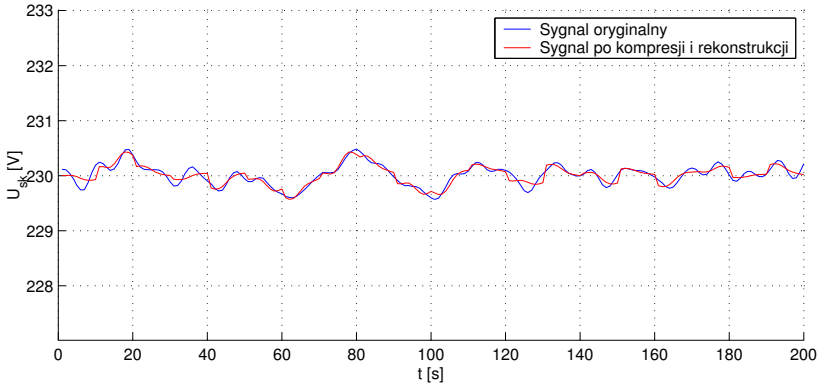
lub jeśli dokonaliśmy wcześniej normalizacji wektorów własnych rozwiązanie upraszcza się do

$$\boldsymbol{y} = \boldsymbol{U}^T(\boldsymbol{x} - \bar{\boldsymbol{x}}) \tag{21}$$

Nowy sygnał $\boldsymbol{y}$ jest skompresowaną, ze stratą jakości, postacią sygnału $\boldsymbol{x}$. Jeśli chcemy sygnał oryginalny zrekonstruować (rozpakować) na podstawie zapisanych, skompresowanych danych musimy dokonać odwrotnego przekształcenia

$$\boldsymbol{x}' = \boldsymbol{U}\boldsymbol{y} + \bar{\boldsymbol{x}} \tag{22}$$

Widać, że do zapisania M próbek sygnału $\boldsymbol{x}$ można użyć K próbek sygnału $\boldsymbol{y}$. Współczynnik kompresji wynosi zatem $M : K$ gdzie $K \leqslant M$. Na rysunku 3 pokazano sygnał oryginalny oraz sygnał poddany kompresji, a nastepnie rekonstrukcji. Współczynnik kompresji był równy 5, gdyż z 10 wektorów własnych pozostawiono 2 związane z największymi wartościami własnymi, co spowodowało widoczną stratę jakości.



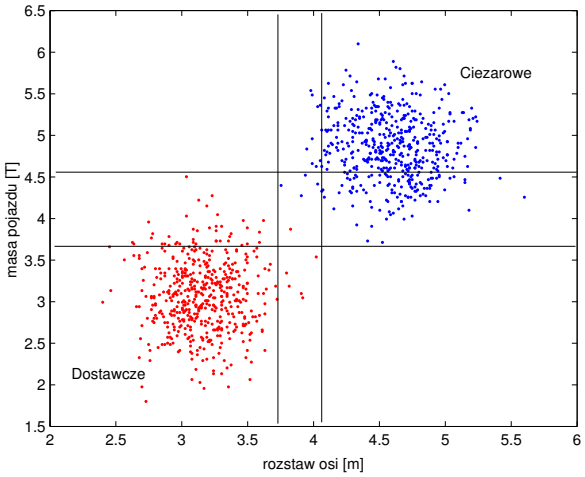
Rysunek 3: Porównanie sygnału oryginalnego z sygnałem oryginalnym poddanym kompresji a następnie rekonstrukcji.

Napisz w Matlabie program kompresujący, według powyższego algorytmu, sygnał zawarty w pliku `lab07z02.dat`, a nastepnie rekonstruujący sygnał oryginalny. Przyjmij rozmiar macierzy kowariancji (15) 10×10 i sprawdź wpływ współczynnika kompresji (od 1 do 10) na kształt zrekonstruowanego sygnału. Pokaż na wykresie sygnały oryginalny i zrekonstruowany. Wyjaśnij skąd, przy wysokim współczynniku kompresji, biorą się widoczne co kilka próbek załamania w sygnale zrekonstruowanym, znajdź analogię do kompresji JPEG i MPEG2.

3 Zmiana układu współrzędnych w zadaniu klasyfikacji – 3 pkt

W punkcie pomiarowym należącym do Zarządu Dróg Krajowych w celach statystycznych zbierane są dane dotyczące liczności i rodzajów pojazdów przejeżdżających przez dany punkt pomiarowy. Ze względu na koszt urządzeń pomiarowych klasyfikacja ta często odbywa się jedynie na podstawie liczby i odstępów między osiami pojazdu. Coraz częściej mierzona jest również masa pojadu w ruchu. Jednym z problemów w tej klasyfikacji jest rozróżnienie mniejszych, dwuosiowych samochodów ciężarowych od większych samochodów dostawczych, gdyż zarówno przedziały mas, w których mieszczą się obie klasy, jak i przedziały

odległości między osiami zachodzą na siebie. Z tego powodu zarejestrowano reprezentatywną próbkę (po 1000 pojazdów z każdej klasy), tzw. ciąg uczący, który pozwalał na dokonanie klasyfikacji. Rejestrowanie ciągu uczącego polegało na zaznaczaniu przez obserwatora do której klasy pojazdów należy każdy pojazd przejeżdżający przez punkt poiarowy. Wyniki rejestracji ciągu uczącego pokazano na rysunku 4.



Rysunek 4: Ciąg uczący z zaznaczonymi obszarami, w których nie da się zaliczyć pojazdów do danej klasy.

Na rysunku tym widać przedziały, w których klasyfikacja po samej masie lub samym rozstawie osi nie dałaby dobrych rezultatów. Widać również, że rozkład wyników każdej klasy przypomina dwuwymiarowy rozkład normalny. Dlatego można zdecydować się klasyfikować pojazdy powyżej pokazanej na rysunku ukośnej linii do jednej klasy, a poniżej do drugiej, czyli inaczej mówiąc wystarczy obrócić układ współrzędych tak aby nowa oś x była zgodna z kierunkiem połączenia maksimów gęstości rozkładów obu klas. Oczywiście przy nowym zestawie danych, mie można mieć stuprocentowej pewności, że klasyfikacja została dokonana poprawnie w każdym przypadku, gdyż opiera się ona na rachunku prawdopodobieństwa[5].

Dane uczące, którymi dysponujemy mają postać macierzy

$$\boldsymbol{A}_{2000\times 2} = \begin{bmatrix} \boldsymbol{mc} & \boldsymbol{rc} \\ \boldsymbol{md} & \boldsymbol{rd} \end{bmatrix} \tag{23}$$

gdzie lewa kolumna zawiera masy pojazdów w tonach, a prawa zawiera rozstawy osi w metrach. Długość kolumn wynika, ze sklejenia wektorów mas $\boldsymbol{mc}$ ciężarówek i $\boldsymbol{md}$ dostawczych oraz rozstawów $\boldsymbol{rc}$, $\boldsymbol{rd}$ (w każdej klasie 1000 pojazdów). Kierunki osi nowego układu współrzędnych będą wyznaczone przez wektory własne macierzy kowariancji $\boldsymbol{C} = \mathrm{cov}(\boldsymbol{A})$. Wektor własny związany z największą wartością własną macierzy kowariancji $\boldsymbol{C}$ będzie osią wzdłuż której dokonamy klasyfikacji. Do obrócenia układu współrzędnych macierzy o wymiarach $M \times N$ potrzebna jest macierz przekształcenia o wymiarach $N \times N$. W tym przypadku będzie to macierz $\boldsymbol{V} = [\boldsymbol{v_1 v_2}]$ złożona z wszystkich (dwóch) wektorów własnych macierzy $\boldsymbol{C}$. Transformacja układu współrzędnych jest iloczynem macierzy

$$\boldsymbol{B} = \boldsymbol{AV} \tag{24}$$

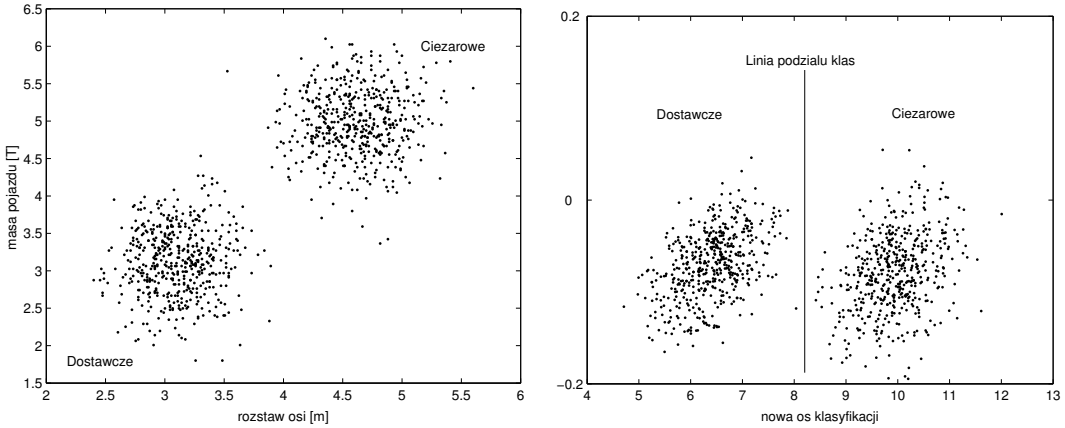
gdzie $\boldsymbol{B}$ jest nową macierzą, w nowym układzie współrzędnych.

Po dokonaniu transformacji otrzymamy wynik jak na rysunku 5. Musimy jeszcze wybrać punkt na osi poziomej, który dzieli wyniki na dwie klasy. W tak prostym przypadku można ten punkt wybrać „na oko“, choć w literaturze dotyczącej klasyfikacji podawane są wzory określające położenie linii rozdzielającej klasy[6].

Mając wyznaczoną macierz przekształcenia $\boldsymbol{V}$ możemy przekształceniu (24) poddać nowe (nie weryfikowane przez obserwatora) macierze danych pomiarowych zawierające parametry pojazdów ciężarowych i dostawczych. Po wykonaniu przekształcenia (24) możemy klasyfikować pojazdy z dużym prawdopodobieństwem poprawnej klasyfikacji.

[5] Prawdopodobieństwo poprawnej klasyfikacji można ocenić znając dokładnie typ rozkładu, wyznaczając parametry rozkładu (np. wartości średnie oraz wariancje mas i rozstawów osi każdej klasy w przypadku rozkładu normalnego).

[6] Można tą linię stosunkowo łatwo znaleźć jako zbiór punktów, w których wartości funkcji gęstości prawdopodobieństwa obu rozkładów dwuwymiarowych są jednakowe i znajdują się na prostej łączącej maksima obu rozkładów.



Rysunek 5: Próba przed (po lewej) i po transformcaji układu współrzędych (po prawej).

A teraz napisz w C program, który na podstawie ciągu uczącego zawartego w pliku `lab07z03ucz.dat` wyznaczy macierz przekształcenia $\boldsymbol{V}$ i zastosuje to przekształcenie do innego ciągu danych podanego jako argument wejściowy programu (np. pliku `lab07z03proba.dat`), a następnie zapisze przetransformowane dane do pliku wyjsciowego `out.dat`. Ostatecznie wczytaj wynikowy plik do Matlaba i przedstaw na wykresie.

Szkielet programu znajdziesz w pliku `lab07z03.c`. Są w nim wskazówki dotyczące kolejności operacji wykonywanych przez program, jednak nie jest to jedyna słuszna droga. Jeśli umiesz ten program napisać w inny sposób to napisz go inaczej (np. korzystając z widoków macierzy i wektorów). Ważne jest żeby program miał te same funkcje i poprawnie działał. Program ten będzie nieco bardziej rozbudowany niż inne programy pisane na tych zajęciach w języku C. Wymaga on zastosowania w jednym programie kilku mechanizmów, które znasz z poprzednich zajęć oraz kilka nowych opisanych dalej. Są przyjmowanie argumentu wywołania, odczyt danych z pliku, obliczanie kowariancji, pobieranie danych ze standardowego wejścia, wektory i macierze, wyznaczanie wartości i wektorów własnych, mnożenie macierzy, zapis wyników do pliku.

W przypadku problemów z uruchamianiem programu testuj jego poszczególne fragmenty (od początku) z podglądem zmiennych (wwatch w trybie debug) lub z wypisywaniem wyników na ekran. Wyniki porównuj z wyliczonymi w Matlabie.

Poniżej znajduje się opis funkcji, które po raz pierwszy pojawiają się w tym zadaniu.

`double gsl_stats_covariance (const double data1[], const size_t stride1, const double data2[], const size_t stride2, const size_t n)` wyznacza kowariancję dwóch zestawów danych zawartych w tablicach `data1` i `data2` o długości `n` według wzoru $\mathrm{cov}(\boldsymbol{xy}) = \sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$. Argumenty `stride1`, `stride2` oznaczają skok i przyjmij je jako równe 1.

`gsl_eigen_symmv_workspace * gsl_eigen_symmv_alloc (const size_t n)` allokuje pamięć potrzebną do wyznaczania wartości i wektorów własnych. `n` jest rozmiarem macierzy kwadratowej, której wektory własne są wyznaczane. Zwraca wskaźnik do zallokowanego obszaru pamięci typu `gsl_eigen_symmv_workspace`.

`void gsl_eigen_symmv_free (gsl_eigen_symmv_workspace * w)` zwalnia miejsce przetrzeni zallokowanej funkcją `void gsl_eigen_symmv_alloc`.

`int gsl_eigen_symmv (gsl_matrix * A, gsl_vector * eval, gsl_matrix * evec, gsl_eigen_symmv_workspace * w)` oblicza wartości i wektory własne symetrycznej rzeczywistej macierzy `A`. Wartosci własne umieszcza w wektorze `eval`, a wektory własne w macierzy `evec`. Korzysta z pamięci `w` przydzielonej wcześniej funkcją `void gsl_eigen_symmv_alloc`. Wartości własne nie są uporządkowane. Wektor własny zapisany w n–tej kolumnie macierzy `evec` odpowiada n–tej wartości własnej z wektora `eval`. Gwarantowana jest ortonormalność wektorów własnych. Wartości i wektory własne można sortować w różnym porządk funkcją `gsl_eigen_symmv_sort` co nie będzie potrzebne w tym zadaniu.

W GSL-u dostępne są również funkcje wyznaczjące wartości i wektory własne macierzy Hermite'a

http://www.gnu.org/software/gsl/manual/gsl-ref_14.html. Do bardziej ogólnych i większych problemów autorzy GSL-a zalecają korzystanie z fortranowego pakietu LAPACK lub wersji CLAPACK dla C.

`int gsl_blas_dgemm (CBLAS_TRANSPOSE_t TransA, CBLAS_TRANSPOSE_t TransB, double alpha, const gsl_matrix * A, const gsl_matrix * B, double beta, gsl_matrix * C)` funkcja mnożenia dwóch macierzy. Wykonuje operację $\boldsymbol{C} = \alpha\,\mathrm{op}(\boldsymbol{A})\mathrm{op}(\boldsymbol{B}) + \beta\boldsymbol{C}$, czyli skalar `alpha` razy macierz `A` (na której wykonano operację `op()`) razy macierz `B` (na której wykonano operację `op()`) plus skalar `beta` razy macierz `C`. Wynik zostaje umieszczony w `C`. Przed wykonaniem działań można wykonać operację `op()` na macierzach `A` oraz `B`. Możliwe operacje to: brak operacji, transpozycja macierzy, transpozycja ze sprzężeniem liczb zespolonych, a odpowiadające tym operacjom wartości parametru `TransA` to: `CblasNoTrans`, `CblasTrans`, `CblasConjTrans`. Nie trzeba wypełniać wcześniej macierzy `C` zerami, wystarczy, że przyjmiesz `beta` równe zero.

Na następne zajęcia

Rozwiązywanie równań i układów równań nieliniowych.