

1 Wstęp

1.1 Nieliniowe zadanie najmniejszych kwadratów

Nieliniowa metoda najmniejszych kwadratów jest jednym z podstawowych sposobów estymacji parametrów modeli nieliniowych. Metoda ta jest najczęściej realizowana za pomocą algorytmu Levenberga–Marquardta (LM). Algorytm Levenberga–Marquardta służy do rozwiązywania zadań optymalizacji postaci

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \|f(x)\|^2, \quad (1)$$

gdzie $f : \mathbb{R}^n \rightarrow \mathbb{R}^N$

$$f(x) = \begin{bmatrix} f_1(x) \\ \vdots \\ f_N(x) \end{bmatrix} \quad (2)$$

jest zadaną (różniczkowalną) funkcją [1]. Oznaczmy jego rozwiązanie przez x^* .

Zadanie (1) można zapisać w postaci

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \sum_{i=1}^N f_i(x)^2, \quad (3)$$

gdzie $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, m$ są składowymi funkcji f . Jeżeli $f(x) = Ax - y$, gdzie $A \in \mathbb{R}^{N \times n}$, $y \in \mathbb{R}^N$, to wówczas mówimy o „liniowym” zadaniu najmniejszych kwadratów, którego rozwiązanie dane jest wzorem

$$x^* = A^\dagger y. \quad (4)$$

Jeżeli $\text{rank } A = n$, to istnieje dokładnie jeden minimalizator, zgodnie ze wzorem (4), przy czym w tym przypadku, tzn. $\text{rank } A = n$, czyli gdy kolumny macierzy A są liniowo niezależne, mamy

$$A^\dagger = (A^T A)^{-1} A^T. \quad (5)$$

Natomiast jeśli $\text{rank } A < n$ to wówczas dowolny wektor postaci $A^\dagger y + \zeta$, gdzie ζ jest wektorem, dla którego zachodzi $A\zeta = 0$, jest rozwiązaniem zadania optymalizacji (1). Jeżeli f jest innej postaci niż $f(x) = Ax - y$, to mówimy o nieliniowym zadaniu najmniejszych kwadratów, lub krócej o nieliniowym zadaniem LS (ang. *nonlinear least-squares problem*).

1.2 Algorytm Gaussa-Newtona

Zauważmy, że w bliskim otoczeniu pewnego ustalonego punktu $x^{(k)}$ mamy

$$f(x) \approx f(x^{(k)}) + Df(x^{(k)})(x - x^{(k)}), \quad (6)$$

gdzie

$$Df(x^{(k)}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(x^{(k)}) & \frac{\partial f_1}{\partial x_2}(x^{(k)}) & \dots & \frac{\partial f_1}{\partial x_n}(x^{(k)}) \\ \frac{\partial f_2}{\partial x_1}(x^{(k)}) & \frac{\partial f_2}{\partial x_2}(x^{(k)}) & \dots & \frac{\partial f_2}{\partial x_n}(x^{(k)}) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_N}{\partial x_1}(x^{(k)}) & \frac{\partial f_N}{\partial x_2}(x^{(k)}) & \dots & \frac{\partial f_N}{\partial x_n}(x^{(k)}) \end{bmatrix}, \quad (7)$$

zatem

$$\|f(x)\|^2 \approx \|f(x^{(k)}) + Df(x^{(k)})(x - x^{(k)})\|^2 \quad (8)$$

Przyjmijmy oznaczenie $v = x - x^{(k)}$, wiemy, że minimalizatorem wyrażenia

$$\|f(x^{(k)}) + Df(x^{(k)})v\|^2. \quad (9)$$

względem v jest

$$v_{(k)}^* = -[Df(x^{(k)})]^\dagger f(x^{(k)}), \quad (10)$$

zatem minimalizatorem wyrażenia

$$\|f(x^{(k)}) + Df(x^{(k)})(x - x^{(k)})\|^2 \quad (11)$$

względem x jest

$$x_{(k)}^* = x^{(k)} - [Df(x^{(k)})]^\dagger f(x^{(k)}). \quad (12)$$

Jeżeli $\text{rank}[Df(x^{(k)})] = n$, to

$$[Df(x^{(k)})]^\dagger = \left[[Df(x^{(k)})]^T Df(x^{(k)}) \right]^{-1} [Df(x^{(k)})]^T, \quad (13)$$

i kładąc $x^{(k+1)} = x_{(k)}^*$ otrzymujemy tzw. rekurencyjny wzór Gaussa-Newtona

$$x^{(k+1)} = x^{(k)} - \left[[Df(x^{(k)})]^T Df(x^{(k)}) \right]^{-1} [Df(x^{(k)})]^T f(x^{(k)}). \quad (14)$$

Podsumowując, w metodzie Gaussa-Newtona punkt $x^{(k+1)}$ jest rozwiązaniem zadania optymalizacji

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \|f(x^{(k)}) + Df(x^{(k)})(x - x^{(k)})\|^2, \quad (15)$$

możemy się spodziewać, że dla odpowiedniego punktu startowego $x^{(0)}$, w sprzyjających „okolicznościach przyrody”, otrzymany ciąg

$$x^{(0)}, x^{(1)}, x^{(2)}, \dots \quad (16)$$

będzie zbieżny do rozwiązania wyjściowego zadania (1), tzn.

$$x^{(0)}, x^{(1)}, x^{(2)}, \dots \xrightarrow{k \rightarrow \infty} x^*. \quad (17)$$

Niestety, wzór Gaussa-Newtona ma pewne wady. Pierwszą z nich jest to, że jeśli macierz $[Df(x^{(k)})]^T Df(x^{(k)})$ we wzorze (14) jest osobliwa, to wzór ten traci sens. Drugą wadą jest to, że otrzymany ciąg może nie być rozbieżny, tzn., że w kolejnych iteracjach nie zachodzi

$$\|f(x^{(k+1)})\|^2 \leq \|f(x^{(k)})\|^2. \quad (18)$$

Sytuacja taka ma miejsce wtedy, gdy w punkcie $x_{(k)}^*$ przybliżenie (8) nie jest spełnione, z uwagi na zbyt dużą odległość punktu $x_{(k)}^*$ od punktu $x^{(k)}$.

1.3 Algorytm Levenberga-Marquardta

Metodę Levenberga-Marquardta (LM) można traktować jako modyfikację metody Gaussa-Netwona. Modyfikacja polega na tym, że w metodzie LM punkt $x^{(k+1)}$ jest rozwiązaniem zadania optymalizacji

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \|f(x^{(k)}) + \text{D}f(x^{(k)})(x - x^{(k)})\|^2 + \lambda^{(k)}\|x - x^{(k)}\|^2, \quad (19)$$

dla pewnego ustalonego $\lambda^{(k)} > 0$. Innymi słowy, modyfikacja polega na dodaniu do funkcji celu w zadaniu (15) wyrazu regularyzacyjnego (czasami mówi się o składniku kary, ang. *penalty term*)

$$\lambda^{(k)}\|x - x^{(k)}\|^2, \quad (20)$$

który nakłada dodatkowy koszt (tzn. wzrost wartości funkcji celu) proporcjonalny do oddalenia od punktu $x^{(k)}$.

Rozwiązanie zadania (19) można wyznaczyć korzystając z tego, że

$$\lambda^{(k)}\|x - x^{(k)}\|^2 = \|\sqrt{\lambda^{(k)}}(x - x^{(k)})\|^2 \quad (21)$$

oraz

$$\begin{aligned} & \|f(x^{(k)}) + \text{D}f(x^{(k)})(x - x^{(k)})\|^2 + \|\sqrt{\lambda^{(k)}}(x - x^{(k)})\|^2 = \\ & \left\| \begin{bmatrix} f(x^{(k)}) + \text{D}f(x^{(k)})(x - x^{(k)}) \\ \sqrt{\lambda^{(k)}}(x - x^{(k)}) \end{bmatrix} \right\|^2. \end{aligned} \quad (22)$$

zatem zadanie (19) jest równoważne zadaniu

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \left\| \begin{bmatrix} f(x^{(k)}) \\ 0 \end{bmatrix} + \begin{bmatrix} \text{D}f(x^{(k)}) \\ \sqrt{\lambda^{(k)}}I \end{bmatrix} (x - x^{(k)}) \right\|^2, \quad (23)$$

skąd otrzymujemy

$$x_{(k)}^* = x^{(k)} - \begin{bmatrix} \text{D}f(x^{(k)}) \\ \sqrt{\lambda^{(k)}}I \end{bmatrix}^\dagger \begin{bmatrix} f(x^{(k)}) \\ 0 \end{bmatrix}. \quad (24)$$

Kładąc $x^{(k+1)} = x_{(k)}^*$ oraz biorąc pod uwagę, że

$$\begin{aligned} & \begin{bmatrix} \text{D}f(x^{(k)}) \\ \sqrt{\lambda^{(k)}}I \end{bmatrix}^\dagger \begin{bmatrix} f(x^{(k)}) \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} \text{D}f(x^{(k)}) \\ \sqrt{\lambda^{(k)}}I \end{bmatrix}^\text{T} \begin{bmatrix} \text{D}f(x^{(k)}) \\ \sqrt{\lambda^{(k)}}I \end{bmatrix}^{-1} \begin{bmatrix} \text{D}f(x^{(k)}) \\ \sqrt{\lambda^{(k)}}I \end{bmatrix}^\text{T} \begin{bmatrix} f(x^{(k)}) \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} \text{D}f(x^{(k)}) \end{bmatrix}^\text{T} \text{D}f(x^{(k)}) + \lambda^{(k)}I \end{bmatrix}^{-1} \begin{bmatrix} \text{D}f(x^{(k)}) \end{bmatrix}^\text{T} f(x^{(k)}). \end{aligned} \quad (25)$$

otrzymujemy zależność

$$x^{(k+1)} = x^{(k)} - \begin{bmatrix} \text{D}f(x^{(k)}) \end{bmatrix}^\text{T} \text{D}f(x^{(k)}) + \lambda^{(k)}I \end{bmatrix}^{-1} \begin{bmatrix} \text{D}f(x^{(k)}) \end{bmatrix}^\text{T} f(x^{(k)}), \quad (26)$$

którą nazywamy **rekurencyjnym wzorem metody Levenberga-Marquardta**. W praktyce ze wzoru (26) korzystamy w ten sposób, że najpierw rozwiązujemy (względem Δ) równanie

$$\begin{bmatrix} \text{D}f(x^{(k)}) \end{bmatrix}^\text{T} \text{D}f(x^{(k)}) + \lambda^{(k)}I \end{bmatrix} \Delta = -\begin{bmatrix} \text{D}f(x^{(k)}) \end{bmatrix}^\text{T} f(x^{(k)}) \quad (27)$$

a następnie wyznaczamy

$$x^{(k+1)} = x^{(k)} + \Delta. \quad (28)$$

Postępowanie takie wynika stąd, że numeryczny koszt rozwiązania układu równań liniowych jest znacznie niższy niż koszt

odwrócenia macierzy. Zauważmy jeszcze, że dla $\lambda^{(k)} > 0$, macierz $\begin{bmatrix} \text{D}f(x^{(k)}) \end{bmatrix}^\text{T} \text{D}f(x^{(k)}) + \lambda^{(k)}I \end{bmatrix}$ zawsze jest odwracalna i w związku z tym równanie (27) ma zawsze jednoznaczne rozwiązanie.

Powiedzmy jeszcze kilka słów na temat warunku stopu. Mamy

$$\nabla \|f(x)\|_2^2 = 2\begin{bmatrix} \text{D}f(x) \end{bmatrix}^\text{T} f(x). \quad (29)$$

Wyprowadzenie powyższego wzoru znajduje się w Dodatku. Stąd, oraz z warunku koniecznego na minimum, wynika, że rozwiązanie x^* zadania (1) spełnia warunek

$$2\begin{bmatrix} \text{D}f(x^*) \end{bmatrix}^\text{T} f(x^*) = 0. \quad (30)$$

Kryterium stopu metody LM może mieć różną postać. Najprostszą jest założenie pewnej maksymalnej liczby iteracji. Innym rozwiązaniem jest zatrzymanie algorytmu dla małej wartości wyrażenia $\|2\begin{bmatrix} \text{D}f(x) \end{bmatrix}^\text{T} f(x)\|^2$ (ang. *optimality condition residual*), tzn.

$$\|2\begin{bmatrix} \text{D}f(x) \end{bmatrix}^\text{T} f(x)\|^2 \approx 0. \quad (31)$$

co oznacza, że warunek konieczny optymalności (30) jest w przybliżeniu spełniony.

Należy wyraźnie podkreślić, że metoda LM jest jedynie pewną heurystyką i nie zawsze jest zbieżna. Okazuje się jednak, że w praktyce sprawdza się dość dobrze.

Zapisany w pseudokodzie algorytm LM, w jego podstawowej wersji, jest podany w poniższej ramce (symbol \leftarrow oznacza przypisanie).

Algorytm Levenberga-Marquardta (A)

dane: różniczkowalna funkcja $f : \mathbb{R}^n \rightarrow \mathbb{R}^N$, punkt startowy $x^{(0)}$, startowa wartość parametru ufności $\lambda^{(0)} > 0$, maksymalna liczba iteracji k_{\max}

$k \leftarrow 0$

while $k < k_{\max}$

$J \leftarrow \text{D}f(x^{(k)})$

$x^{(k+1)} \leftarrow x^{(k)} - \begin{bmatrix} J^\text{T} J + \lambda^{(k)} I \end{bmatrix}^{-1} J^\text{T} f(x^{(k)})$

if $\|f(x^{(k+1)})\|^2 < \|f(x^{(k)})\|^2$

$\lambda^{(k+1)} \leftarrow 0.8\lambda^{(k)}$

else

$\lambda^{(k+1)} \leftarrow 2\lambda^{(k)}$

$x^{(k+1)} \leftarrow x^{(k)}$

end

$k \leftarrow k + 1$

end

Zapisany w pseudokodzie algorytm LM, z bardziej wyrafinowanym kryterium stopu, jest podany w poniższej ramce.

Algorytm Levenberga-Marquardta (B)

dane: różniczkowalna funkcja $f : \mathbb{R}^n \rightarrow \mathbb{R}^N$, punkt startowy $x^{(0)}$, startowa wartość parametru ufności $\lambda^{(0)} > 0$, dokładność rozwiązania $\epsilon > 0$

```

k ← 0
J ← Df(x(k))
while ||2JTf(x(k))||2 > ε
    J ← Df(x(k))
    x(k+1) ← x(k) - [JTJ + λ(k)I]-1 JTf(x(k))
    if ||f(x(k+1))||2 < ||f(x(k))||2
        λ(k+1) ← 0.8λ(k)
    else
        λ(k+1) ← 2λ(k)
        x(k+1) ← x(k)
    end
    k ← k + 1
end

```

1.4 Estymacja parametrów metodą LM

Założmy, że dysponujemy N pomiarami y_1, \dots, y_m pewnego procesu o modelu $\varphi(x, u)$ dla odpowiednich chwil czasowych u_1, \dots, u_m . Chcemy wyznaczyć wektor parametrów x , tak aby zminimalizować wyrażenie

$$\sum_{i=1}^m [\varphi(x, u_i) - y_i]^2. \quad (32)$$

Zauważmy, że jeśli wprowadzimy oznaczenie

$$f_i(x) = \varphi(x, u_i) - y_i, \quad (33)$$

to minimalizacja wyrażenia (32) sprowadza się do zadania optymalizacji (1). Mamy wówczas

$$f(x) = \begin{bmatrix} f_1(x) \\ \vdots \\ f_N(x) \end{bmatrix} = \begin{bmatrix} \varphi(x, u_1) - y_1 \\ \vdots \\ \varphi(x, u_N) - y_N \end{bmatrix} \quad (34)$$

oraz

$$Df(x) = \begin{bmatrix} \frac{\partial \varphi(x, u_1)}{\partial x_1} & \frac{\partial \varphi(x, u_1)}{\partial x_2} & \cdots & \frac{\partial \varphi(x, u_1)}{\partial x_n} \\ \frac{\partial \varphi(x, u_2)}{\partial x_1} & \frac{\partial \varphi(x, u_2)}{\partial x_2} & \cdots & \frac{\partial \varphi(x, u_2)}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \varphi(x, u_N)}{\partial x_1} & \frac{\partial \varphi(x, u_N)}{\partial x_2} & \cdots & \frac{\partial \varphi(x, u_N)}{\partial x_n} \end{bmatrix} \quad (35)$$

Przykład 1. Założmy, że dysponujemy N pomiarami y_1, \dots, y_m procesu

$$\varphi(x, t) = A \sin(\omega t + \phi), \quad (36)$$

dla odpowiednich chwil czasowych t_1, \dots, t_m . Chcemy znaleźć wartości parametrów A , ω oraz ϕ tak aby otrzymać najlepsze dopasowanie modelu $\varphi(x, t)$ do posiadanych danych pomiarowych. Innymi słowy, chcemy zminimalizować wyrażenie

$$\sum_{i=1}^m [A \sin(\omega t_i + \phi) - y_i]^2, \quad (37)$$

względem parametrów A , ω i ϕ , co jest równoważne rozwiązaniu zadania (1) dla

$$f_i(x) = x_1 \sin(x_2 t + x_3) - y_i, \quad (38)$$

gdzie

$$x = [A, \omega, \phi]^T \quad (39)$$

jest wektorem zmiennych optymalizacyjnych (które w rozpatrywanym kontekście mają interpretację parametrów modelu). Dla funkcji o składowych (38) mamy

$$\varphi(x, t) = x_1 \sin(x_2 t + x_3) \quad (40)$$

oraz

$$\frac{\partial \varphi(x, t_k)}{\partial x_1} = \sin(x_2 t_k + x_3), \quad (41a)$$

$$\frac{\partial \varphi(x, t_k)}{\partial x_2} = x_1 t_k \cos(x_2 t_k + x_3), \quad (41b)$$

$$\frac{\partial \varphi(x, t_k)}{\partial x_3} = x_1 \cos(x_2 t_k + x_3), \quad (41c)$$

dla $k = 1, \dots, N$, zatem zgodnie ze wzorem (35) otrzymujemy

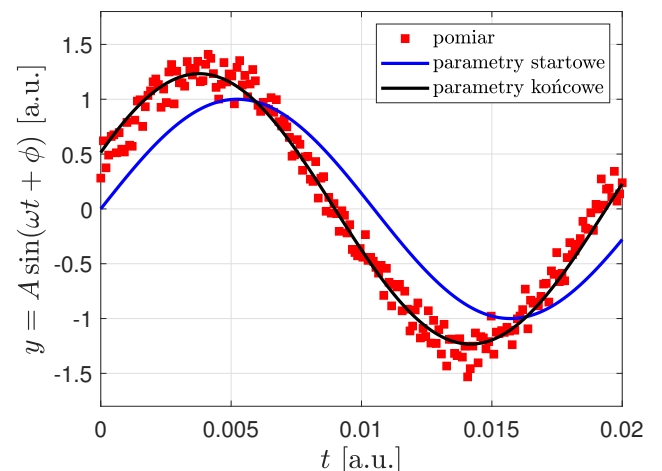
$$Df(x) = \begin{bmatrix} \sin(\alpha_1) & x_1 t_1 \cos(\alpha_1) & x_1 \cos(\alpha_1) \\ \sin(\alpha_2) & x_1 t_2 \cos(\alpha_2) & x_1 \cos(\alpha_2) \\ \vdots & \vdots & \vdots \\ \sin(\alpha_N) & x_1 t_N \cos(\alpha_N) & x_1 \cos(\alpha_N) \end{bmatrix}, \quad (42)$$

gdzie $\alpha_k = x_2 t_k + x_3$ dla $k = 1, \dots, N$, oraz

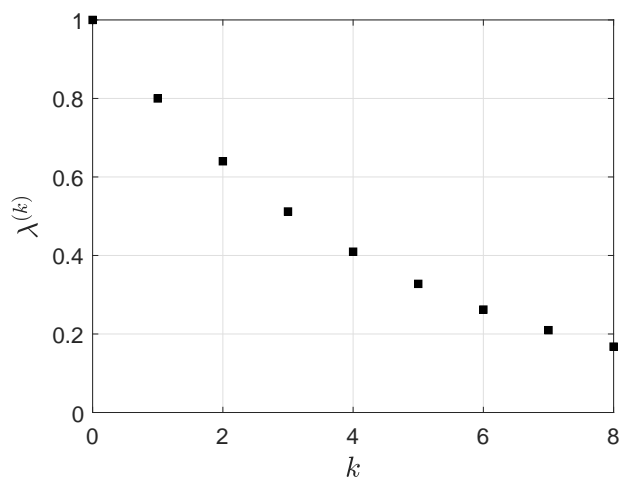
$$f(x) = \begin{bmatrix} x_1 \sin(\alpha_1) - y_1 \\ x_1 \sin(\alpha_2) - y_2 \\ \vdots \\ x_1 \sin(\alpha_N) - y_N \end{bmatrix}. \quad (43)$$

2 Zadania

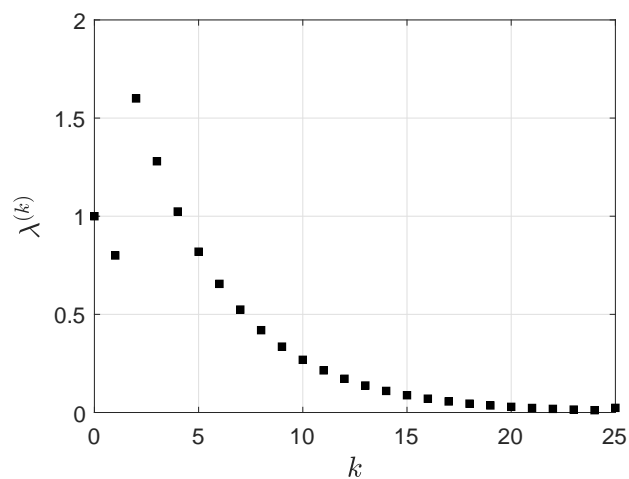
Zadanie 1. Korzystając z algorytmu LM dopasować parametry modelu $y = A \sin(\omega t + \phi)$ do danych pomiarowych z pliku `LM01Data.mat`. Przyjąć $\lambda^{(0)} = 1.0$. Napisać odpowiedni skrypt w środowisku Matlab i wygenerować wykresy przedstawione na Rys. 1-3.



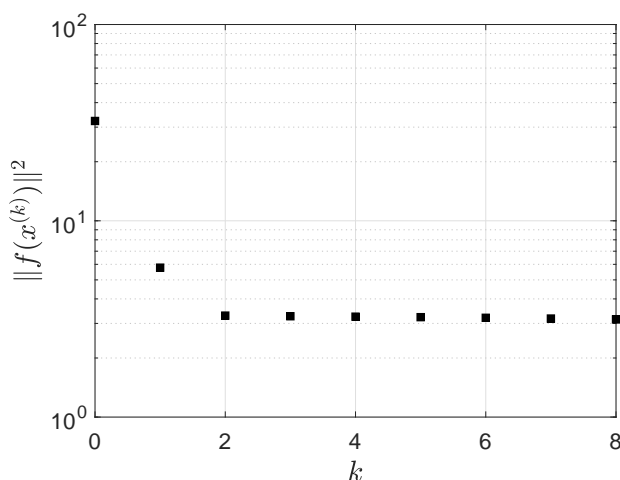
Rysunek 1: Rekonstrukcja sygnału sinusoidalnego $y = A \sin(\omega t + \phi)$ [Zadanie 1].



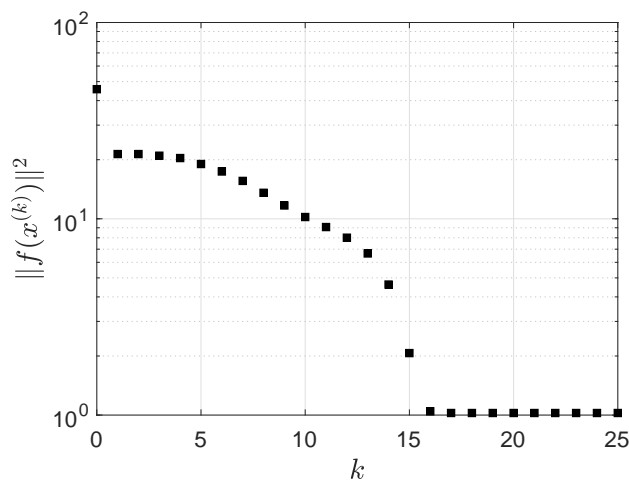
Rysunek 2: Zmiany wartości współczynnika ufnosci $\lambda^{(k)}$ w kolejnych iteracjach algorytmu LM [Zadanie 1].



Rysunek 5: Zmiany wartości współczynnika ufnosci $\lambda^{(k)}$ w kolejnych iteracjach algorytmu LM [Zadanie 2].

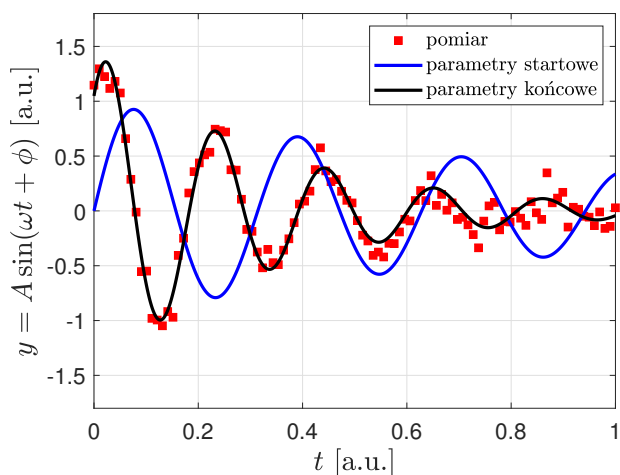


Rysunek 3: Zmiany wartości funkcji celu w kolejnych iteracjach algorytmu LM [Zadanie 1].



Rysunek 6: Zmiany wartości funkcji celu w kolejnych iteracjach algorytmu LM [Zadanie 2].

Zadanie 2. Korzystając z algorytmu LM dopasować parametry modelu $y = Ae^{-at} \sin(\omega t + \phi)$ do danych pomiarowych z pliku LM04Data.mat. Napisać odpowiedni skrypt w środowisku Matlab i wygenerować wykresy przedstawione na Rys. 4-6.



Rysunek 4: Rekonstrukcja tłumionego sygnału sinusoidalnego $y = Ae^{-at} \sin(\omega t + \phi)$ [Zadanie 2].

Zadanie 3. W pracy [2] rozpatruje się transmitancję

$$G(s) = -M\omega_s \frac{Ks + 2\Delta}{s^4 + Ks^3 + 2\Sigma s^2 + K\Sigma s + \Delta^2}, \quad (44)$$

gdzie

$$M = \frac{2\sqrt{V_{in}^2 - V_{out}^2}}{\pi|1 - \omega_s^2 LC|}, \quad (45a)$$

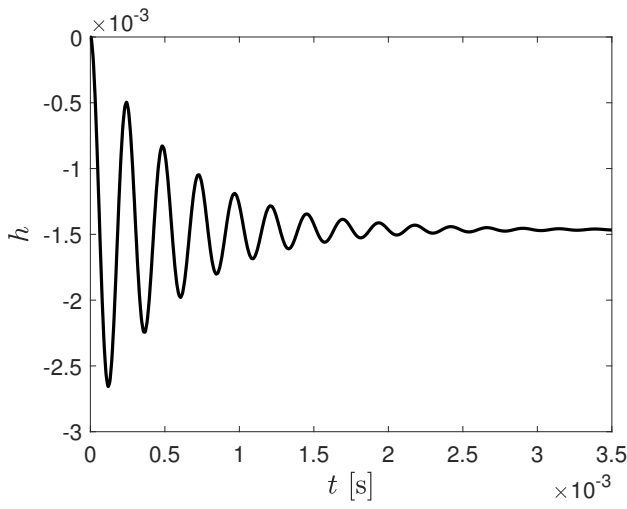
$$K = \frac{2\omega_0^2 V_{out}}{\pi M \omega_s}, \quad (45b)$$

$$\omega_0^2 = \frac{1}{LC}, \quad (45c)$$

$$\Sigma = \omega_s^2 + \omega_0^2, \quad (45d)$$

$$\Delta = \omega_s^2 - \omega_0^2 \quad (45e)$$

oraz $L = 197 \times 10^{-6}$ H, $C = 100 \times 10^{-9}$ F, $V_{in} = 14.0$ V, $V_{out} = 1.95$ V, $\omega_s = 40 \times 2\pi \times 10^3$ rad/s. Rysunek 7 przedstawiono wykres odpowiedzi skokowej dla tej transmitancji.



Rysunek 7: Odpowiedź skokowa układu opisanego transmitancją (44) [Zadanie 3].

Odpowiedź ta przypomina odpowiedź skokową członu oscylacyjnego. Zauważmy, że człon oscylacyjny jest członem drugiego rzędu, natomiast transmitancja (44) jest rzędu czwartego. Na ogół łatwiej jest zaprojektować regulator dla członu drugiego rzędu niż dla członu czwartego rzędu, dlatego chcielibyśmy znaleźć taki człon drugiego rzędu, którego odpowiedź skokowa jest możliwie najbardziej zbliżona do odpowiedzi skokowej członu (44). Wiemy, że odpowiedź skokowa członu oscylacyjnego

$$G_{\text{osc}}(s) = \frac{k}{1 + 2\xi Ts + T^2 s^2}, \quad (46)$$

jest dana wzorem

$$h_{\text{osc}}(t) = k \left[1 - e^{-\gamma t} \left(\cos(\beta t) + \frac{\gamma}{\beta} \sin(\beta t) \right) \right], \quad (47)$$

gdzie

$$\gamma = \frac{\xi}{T}, \quad \beta = \frac{\sqrt{1 - \xi^2}}{T}. \quad (48)$$

Ze wzoru (48) wynika, że

$$\xi = \frac{\gamma}{\sqrt{\beta^2 + \gamma^2}}, \quad T = \frac{1}{\sqrt{\beta^2 + \gamma^2}}, \quad (49)$$

innymi słowy

$$T = \frac{1}{\sqrt{\beta^2 + \gamma^2}}, \quad \xi = \gamma T. \quad (50)$$

Dysponując wartościami odpowiedzi skokowej $h(t)$ transmitancji (44) i oznaczając, $y_i = h(t_i)$, $i = 1, \dots, N$, możemy wyznaczyć poszukiwane wartości parametrów γ i β minimalizując wyrażenie

$$\sum_{i=1}^N \left\{ k \left[1 - e^{-\gamma t_i} \left(\cos(\beta t_i) + \frac{\gamma}{\beta} \sin(\beta t_i) \right) \right] - y_i \right\}^2, \quad (51)$$

co przy oznaczeniu

$$f_i(k, \gamma, \xi) = k \left[1 - e^{-\gamma t_i} \left(\cos(\beta t_i) + \frac{\gamma}{\beta} \sin(\beta t_i) \right) \right] - y_i, \quad (52)$$

dla $i = 1, \dots, N$ oraz $k = x_1$, $\gamma = x_2$, $\xi = x_3$ sprowadza się do zadania (1). Zauważmy, że

$$\frac{\partial f_i}{\partial k} = 1 - e^{-\gamma t_i} \left(\cos(\beta t_i) + \frac{\gamma}{\beta} \sin(\beta t_i) \right), \quad (53a)$$

$$\frac{\partial f_i}{\partial \gamma} = k e^{-\gamma t_i} \left(t_i \cos(\beta t_i) - \frac{1 - t_i \gamma}{\beta} \sin(\beta t_i) \right), \quad (53b)$$

$$\frac{\partial f_i}{\partial \beta} = k e^{-\gamma t_i} \left[\left(t_i + \frac{\gamma}{\beta^2} \right) \sin(\beta t_i) - \frac{\gamma}{\beta} t_i \cos(\beta t_i) \right], \quad (53c)$$

dla $i = 1, \dots, N$. Zatem, biorąc pod uwagę wspomniane już oznaczenia $k = x_1$, $\gamma = x_2$, $\xi = x_3$ mamy

$$\frac{\partial f_i}{\partial x_1} = 1 - e^{-x_2 t_i} \left(\cos(x_3 t_i) + \frac{x_2}{x_3} \sin(x_3 t_i) \right), \quad (54a)$$

$$\frac{\partial f_i}{\partial x_2} = x_1 e^{-x_2 t_i} \left(t_i \cos(x_3 t_i) - \frac{1 - t_i x_2}{x_3} \sin(x_3 t_i) \right), \quad (54b)$$

$$\frac{\partial f_i}{\partial x_3} = x_1 e^{-x_2 t_i} \left[\left(t_i + \frac{x_2}{x_3^2} \right) \sin(x_3 t_i) - \frac{x_2}{x_3} t_i \cos(x_3 t_i) \right] \quad (54c)$$

dla $i = 1, \dots, N$.

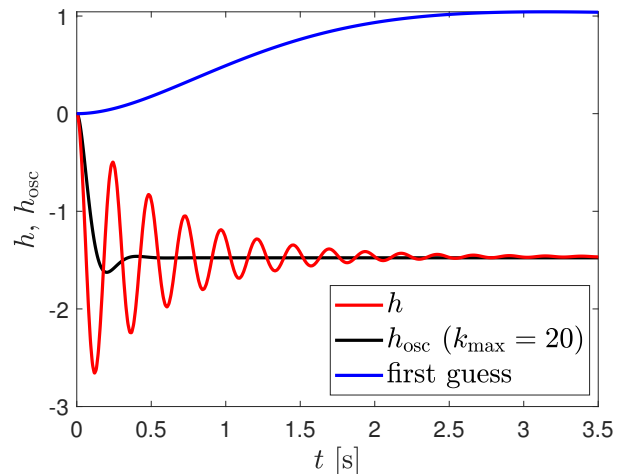
Korzystając z metody LM wyznacz odpowiednik drugiego rzędu w postaci członu oscylacyjnego dla transmitancji (44). Odpowiedź skokową transmitancji (44) możesz wyznaczyć samodzielnie lub skorzystać z danych w pliku `reductionData.mat`. Wykonaj odpowiednie wykresy. Jako punkt startowy przyjmij $x^{(0)} = [1, 1, 1]^T$. Więcej informacji na temat transmitancji i odpowiedzi skokowej układu można znaleźć np. w [3] i [4].

Uwaga: Z Rys. 7 wynika, że wartości y_1, y_2, \dots, y_N i t_1, t_2, \dots, t_N są stosunkowo małe, rzędu 10^{-3} . Wygodnie jest obliczenia prowadzić dla przeskalowanych wartości $\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_N$ i $\tilde{t}_1, \tilde{t}_2, \dots, \tilde{t}_N$ a następnie odzyskać „prawdziwe” wartości parametrów przeskalowując je ponownie. Jeśli $\tilde{y}_i = 10^3 \times y_i$ i $\tilde{t}_i = 10^3 \times t_i$ dla $i = 1, \dots, N$, to po zakończeniu trzeba otrzymane parametry k, γ i β przemnożyć przez 10^{-3} . Ostatecznie, zidentyfikowane wartości parametrów, już po przeskalowaniu, to

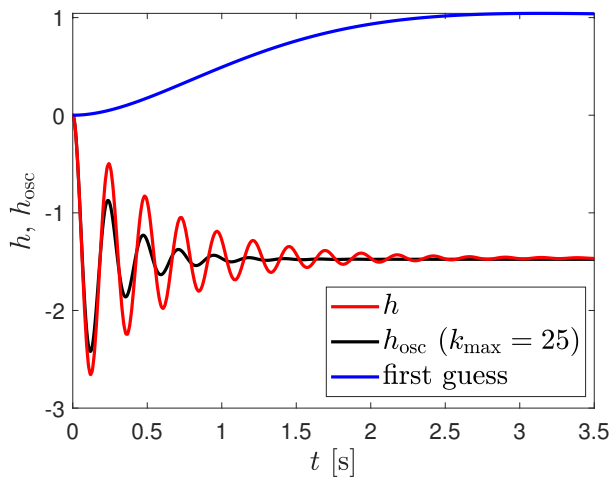
$$k = -1.4639 \times 10^{-3}, \quad (55a)$$

$$\gamma = 1.7303 \times 10^{-3}, \quad (55b)$$

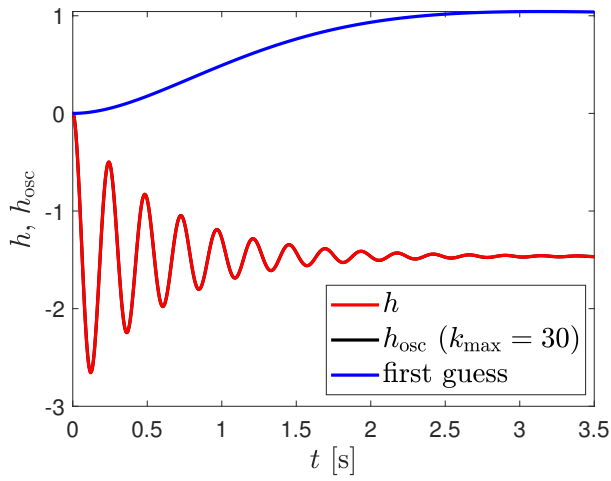
$$\beta = -25.9807 \times 10^{-3}. \quad (55c)$$



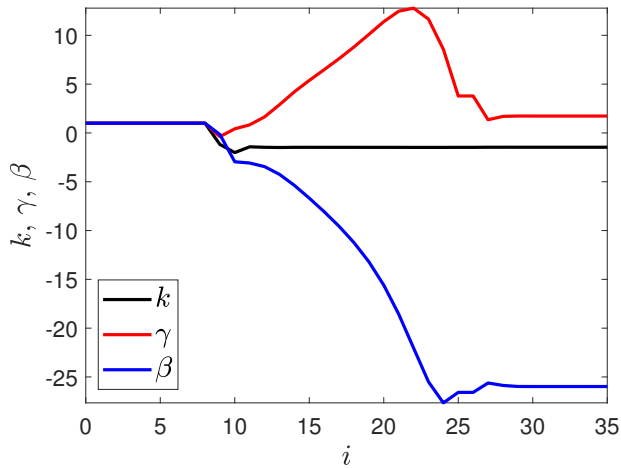
Rysunek 8: Wykresy odpowiedzi skokowa $h(t)$ (czerwony) układu opisanego transmitancją (44) oraz odpowiedzi skokowe $h_{\text{osc}}(t)$ członu oscylacyjnego opisanego transmitancją (46) dla wartości parametrów początkowych (niebieski) oraz po 20 iteracjach (czarny). [Zadanie 3].



Rysunek 9: Wykresy odpowiedzi skokowa $h(t)$ (czerwony) układu opisanego transmitancją (44) oraz odpowiedzi skokowe $h_{osc}(t)$ członu oscylacyjnego opisanego transmitancją (46) dla wartości parametrów początkowych (niebieski) oraz po 25 iteracjach (czarny). [Zadanie 3].



Rysunek 10: Wykresy odpowiedzi skokowa $h(t)$ (czerwony) układu opisanego transmitancją (44) oraz odpowiedzi skokowe $h_{osc}(t)$ członu oscylacyjnego opisanego transmitancją (46) dla wartości parametrów początkowych (niebieski) oraz po 30 iteracjach (czarny). [Zadanie 3].



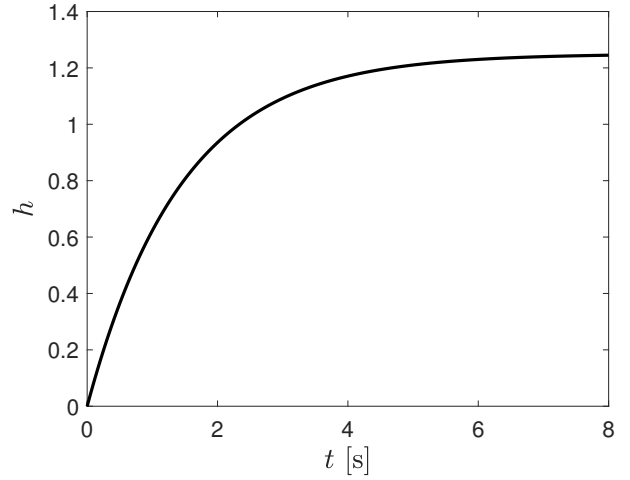
Rysunek 11: Zmiany wartości parametrów k , γ i β w kolejnych iteracjach algorytmu LM. [Zadanie 3].

Zadanie 4. Człon inercyjny jest opisany transmitancją

$$G(s) = \frac{k}{Ts + 1}. \quad (56)$$

Odpowiedź skokowa tego członu jest dana wzorem

$$h(t) = k \left(1 - e^{-\frac{t}{T}}\right). \quad (57)$$



Rysunek 12: Wykres odpowiedzi skokowej $h(t)$ członu inercyjnego. [Zadanie 4].

Założmy, że dysponujemy próbkami odpowiedzi skokowej y_1, \dots, y_N dla chwil t_1, \dots, t_N i chcemy wyznaczyć wartości wzmocnienia k i stałej czasowej T dopasowując wzór (57) metodą LM. Innymi słowy chcemy metodą LM zminimalizować względem k i T wartość wyrażenia

$$\sum_{i=1}^N [h(t_i) - y_i]^2 = \sum_{i=1}^N \left[k \left(1 - e^{-\frac{t_i}{T}}\right) - y_i \right]^2. \quad (58)$$

Odpowiada to zadaniu (1) dla

$$f_i = k \left(1 - e^{-\frac{t_i}{T}}\right) - y_i, \quad i = 1, \dots, N. \quad (59)$$

Mamy

$$\frac{\partial f_i}{\partial k} = 1 - e^{-\frac{t_i}{T}}, \quad (60a)$$

$$\frac{\partial f_i}{\partial T} = -\frac{k}{T^2} e^{-\frac{t_i}{T}}, \quad (60b)$$

dla $i = 1, \dots, N$.

Dla danych z pliku `inertialData.mat` wyznacz metodą LM wartości parametrów k i T , wykonaj odpowiednie wykresy.

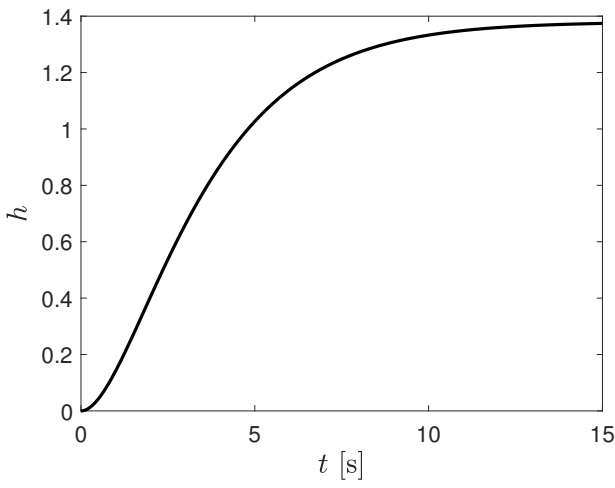
(Odpowiedź: $k = 1.45$, $T = 1.25$.)

Zadanie 5. Człon dwuinercyjny jest opisany transmitancją

$$G(s) = \frac{k}{(T_1s + 1)(T_2s + 1)}. \quad (61)$$

Odpowiedź skokowa tego członu jest dana wzorem

$$h(t) = k \left[1 - \frac{1}{T_1 - T_2} \left(T_1 e^{-\frac{t}{T_1}} - T_2 e^{-\frac{t}{T_2}} \right) \right]. \quad (62)$$



Rysunek 13: Wykres odpowiedzi skokowej $h(t)$ członu dwu-inercyjnego. [Zadanie 5].

Założmy, że dysponujemy próbkami odpowiedzi skokowej y_1, \dots, y_N dla chwil t_1, \dots, t_N i chcemy wyznaczyć wartości wzmocnienia k i stałych czasowych T_1 i T_2 dopasowując wzór (62) metodą LM. Innymi słowy chcemy metodą LM zminimalizować względem k, T_1 i T_2 wartość wyrażenia

$$\sum_{i=1}^N [h(t_i) - y_i]^2. \quad (63)$$

czyli

$$\sum_{i=1}^N \left\{ k \left[1 - \frac{1}{T_1 - T_2} \left(T_1 e^{-\frac{t_i}{T_1}} - T_2 e^{-\frac{t_i}{T_2}} \right) \right] - y_i \right\}^2. \quad (64)$$

Odpowiada to zadaniu (1) dla

$$f_i = k \left[1 - \frac{1}{T_1 - T_2} \left(T_1 e^{-\frac{t_i}{T_1}} - T_2 e^{-\frac{t_i}{T_2}} \right) \right] - y_i \quad (65)$$

oraz $i = 1, \dots, N$.

Mamy

$$\frac{\partial f_i}{\partial k} = 1 - \frac{1}{T_1 - T_2} \left(T_1 e^{-\frac{t_i}{T_1}} - T_2 e^{-\frac{t_i}{T_2}} \right), \quad (66a)$$

$$\frac{\partial f_i}{\partial T_1} = \frac{k}{T_2 - T_1} \left[\frac{t_i}{T_1} e^{-\frac{t_i}{T_1}} + \frac{T_2}{T_1 - T_2} \left(e^{-\frac{t_i}{T_2}} - e^{-\frac{t_i}{T_1}} \right) \right], \quad (66b)$$

$$\frac{\partial f_i}{\partial T_2} = \frac{k}{T_1 - T_2} \left[\frac{t_i}{T_2} e^{-\frac{t_i}{T_2}} + \frac{T_1}{T_2 - T_1} \left(e^{-\frac{t_i}{T_1}} - e^{-\frac{t_i}{T_2}} \right) \right], \quad (66c)$$

dla $i = 1, \dots, N$.

Dla danych z pliku `doubleInertialData.mat` wyznacz metodą LM wartości parametrów k i T , wykonaj odpowiednie wykresy.

(Odpowiedź: $k = 1.38, T_1 = 1.45, T_2 = 2.32$.)

Zadanie 6. Powtórzyć wszystkie wcześniejsze zadania korzystając z funkcji `lsqnonlin` lub `lsqcurvefit`, w wersji z jacobianem i bez jacobianu. Więcej informacji

<https://www.mathworks.com/help/optim/ug/lsqnonlin.html#buuhch7-problem>

oraz

https://www.mathworks.com/help/optim/ug/lsqcurvefit.html?searchHighlight=lsqcurvefit&s_tid=doc_srchttitle

W szczególności proszę zwrócić uwagę na informację na temat przekazywania jacobianu (ang. *Jacobian*).

Dodatek

Wyprowadzenie wzoru (29) Wzór (29) można otrzymać następująco.

$$\begin{aligned} \nabla \|f(x)\|_2^2 &= \nabla \left(\sum_{i=1}^N f_i(x)^2 \right) \\ &= \sum_{i=1}^N \nabla (f_i(x)^2) \\ &= \sum_{i=1}^N \begin{bmatrix} \frac{\partial f_i^2}{\partial x_1}(x) \\ \frac{\partial f_i^2}{\partial x_2}(x) \\ \vdots \\ \frac{\partial f_i^2}{\partial x_n}(x) \end{bmatrix} \\ &= \sum_{i=1}^N \begin{bmatrix} 2f_i(x) \frac{\partial f_i}{\partial x_1}(x) \\ 2f_i(x) \frac{\partial f_i}{\partial x_2}(x) \\ \vdots \\ 2f_i(x) \frac{\partial f_i}{\partial x_n}(x) \end{bmatrix} \\ &= \sum_{i=1}^N 2f_i(x) \begin{bmatrix} \frac{\partial f_i}{\partial x_1}(x) \\ \frac{\partial f_i}{\partial x_2}(x) \\ \vdots \\ \frac{\partial f_i}{\partial x_n}(x) \end{bmatrix} \\ &= \sum_{i=1}^N 2f_i(x) \nabla f_i(x) \\ &= 2 \begin{bmatrix} \nabla f_1(x) & \dots & \nabla f_N(x) \end{bmatrix} \begin{bmatrix} f_1(x) \\ \vdots \\ f_N(x) \end{bmatrix} \\ &= 2[Df(x)]^T f(x). \end{aligned} \quad (67)$$

Alternatywnie można skorzystać z ogólnego wzoru

$$\nabla (f^T g) = (Df)^T g + (Dg)^T f, \quad (68)$$

podstawiając $g = f$. Wzór (68) można wyprowadzić następująco. Dla różniczkowalnych funkcji, w ogólnym przypadku wektorowych, $f: \mathbb{R}^n \rightarrow \mathbb{R}^m, g: \mathbb{R}^n \rightarrow \mathbb{R}^m$,

$$f(x) = \begin{bmatrix} f_1(x) \\ \vdots \\ f_m(x) \end{bmatrix}, \quad g(x) = \begin{bmatrix} g_1(x) \\ \vdots \\ g_m(x) \end{bmatrix}, \quad (69)$$

mamy,

$$Df = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \dots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}, \quad (70)$$

skąd wynika, że

$$Df = \begin{bmatrix} Df_1 \\ Df_2 \\ \vdots \\ Df_m \end{bmatrix} \quad (71)$$

oraz

$$\begin{aligned}
 (Df)^T &= \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_2}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_1} \\ \frac{\partial f_1}{\partial x_2} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_1}{\partial x_n} & \frac{\partial f_2}{\partial x_n} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \\
 &= [\nabla f_1 \quad \nabla f_2 \quad \cdots \quad \nabla f_m], \quad (72)
 \end{aligned}$$

i analogicznie dla g .

Mamy

$$\begin{aligned}
 D(f^T g) &= D\left(\sum_{i=1}^m f_i g_i\right) \\
 &= \sum_{i=1}^m D(f_i g_i) \\
 &= \sum_{i=1}^m \left[\frac{\partial(f_i g_i)}{\partial x_1} \quad \cdots \quad \frac{\partial(f_i g_i)}{\partial x_n} \right] \\
 &= \sum_{i=1}^m \left[g_i \frac{\partial f_i}{\partial x_1} + f_i \frac{\partial g_i}{\partial x_1} \quad \cdots \quad g_i \frac{\partial f_i}{\partial x_n} + f_i \frac{\partial g_i}{\partial x_n} \right] \\
 &= \sum_{i=1}^m \left[g_i \frac{\partial f_i}{\partial x_1} \quad \cdots \quad g_i \frac{\partial f_i}{\partial x_n} \right] + \sum_{i=1}^m \left[f_i \frac{\partial g_i}{\partial x_1} \quad \cdots \quad f_i \frac{\partial g_i}{\partial x_n} \right] \\
 &= \sum_{i=1}^m g_i \left[\frac{\partial f_i}{\partial x_1} \quad \cdots \quad \frac{\partial f_i}{\partial x_n} \right] + \sum_{i=1}^m f_i \left[\frac{\partial g_i}{\partial x_1} \quad \cdots \quad \frac{\partial g_i}{\partial x_n} \right] \\
 &= \sum_{i=1}^m g_i Df_i + \sum_{i=1}^m f_i Dg_i \\
 &= g^T Df + f^T Dg. \quad (73)
 \end{aligned}$$

Uwzględniając, że

$$\nabla(f^T g) = [D(f^T g)]^T \quad (74)$$

otrzymujemy ostatecznie

$$\nabla(f^T g) = (Df)^T g + (Dg)^T f. \quad (75)$$

Literatura

- [1] S. Boyd and L. Vandenberghe. *Introduction to Applied Linear Algebra: Vectors, Matrices, and Least Squares*. Cambridge University Press, 2018.
- [2] S.R Sanders, J.M Noworolski, X.Z Liu, and G.C Verghese. Generalized averaging method for power conversion circuits. *IEEE Transactions on Power Electronics*, 6(2):251–259, 1991.
- [3] T. Kaczorek, A. Dzieliński, W. Dąbrowski, and R. Łopatka. *Podstawy teorii sterowania*. Wydawnictwo WNT : Wydawnictwo Naukowe PWN, Warszawa, wyd. 3 - 1 dodr. (pwn). edition, 2016.
- [4] J. Mazurek, H. Vogt, and W. Żydanowicz. *Podstawy automatyki*. Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa, wyd. 7. edition, 2006.