

Politechnika Warszawska
Wydział Elektroniki i Technik
Informacyjnych

Projektowanie układów sterowania, projekt
grupowy (PUST): materiały dydaktyczne

Algorytmy rozmyte PID i DMC

Maciej Ławryńczuk

Warszawa, 2020

1. Regulacja rozmyta

Właściwości nieliniowego obiektu (np. wzmocnienie) zmieniają się w zależności od punktu pracy. Zastosowanie do takiego procesu klasycznych, liniowych algorytmów regulacji, np. PID lub DMC, może prowadzić do niezadowalającej jakości regulacji w niektórych punktach pracy (niestabilności lub przeciwnie – bardzo wolnych zmian), podczas gdy w innych punktach pracy regulator będzie pracował zadowalająco.

Opisana obserwacja była motywacją do zastosowania regulatorów rozmytych. Koncepcja ta jest bardzo prosta – polega na użyciu nie jednego regulatora, a zbioru n_r lokalnych regulatorów liniowych. Mają one inne parametry (np. wzmocnienie), są przełączane w zależności od aktualnego punktu pracy.

1.1. Regulacja rozmyta PID

Sygnał sterujący w aktualnej dyskretniej chwili $k = 0, 1, 2, \dots$ klasycznego, liniowego regulatora PID ma postać

$$u(k) = r_0 e(k) + r_1 e(k-1) + r_2 e(k-2) + u(k-1)$$

gdzie $e(k) = y^{\text{zad}}(k) - y(k)$ jest uchybem dla aktualnej chwili k . Współczynniki r_0, r_1, r_2 najłatwiej obliczyć na podstawie parametrów regulatora z czasem ciągłym.

Bardzo często regulatory rozmyte opisuje się podając bazę reguł. Załóżmy, że aktualnie mierzony sygnał wyjściowy procesu, tzn. $y(k)$, określa, która reguła (regulator lokalny) będzie aktywny i w jakim stopniu. Liczba reguł jest równa liczbie lokalnych regulatorów liniowych. Baza reguł rozmytego regulatora PID ma postać

$$\text{Reguła } 1: \text{ jeżeli } y(k) \in Y^1, \text{ to } u^1(k) = r_0^1 e(k) + r_1^1 e(k-1) + r_2^1 e(k-2) + u(k-1) \quad (1)$$

\vdots

$$\text{Reguła } n_r: \text{ jeżeli } y(k) \in Y^{n_r}, \text{ to } u^{n_r}(k) = r_0^{n_r} e(k) + r_1^{n_r} e(k-1) + r_2^{n_r} e(k-2) + u(k-1) \quad (2)$$

Uwaga: po prawej stronie występuje sygnał $u(k-1)$, nie sygnały $u^1(k-1), \dots, u^{n_r}(k-1)$. Pierwszy regulator lokalny ma parametry: r_0^1, r_1^1, r_2^1 , drugi – parametry: r_0^2, r_1^2, r_2^2 , ostatni – parametry $r_0^{n_r}, r_1^{n_r}, r_2^{n_r}$. gdzie Y^1, \dots, Y^{n_r} oznaczają zbiory rozmyte, np. trójkątne, trapezowe, Gaussowskie, sigmoidalne. Na podstawie wartości sygnału wybierającego regulatory lokalne, u nas jest to sygnał $y(k)$, oblicza się współczynniki przynależności dla poszczególnych zbiorów. Na przykład, stosując funkcję Gaussa postaci

$$\mu(k) = \mu(y(k)) = \exp\left(\frac{-(y(k) - c)^2}{2\sigma^2}\right) \quad (3)$$

oblicza się współczynniki przynależności poszczególnych regulatorów lokalnych, zależne od aktualnego punktu pracy

$$w^1(k) = \mu(y(k)) = \exp\left(\frac{-(y(k) - c^1)^2}{2(\sigma^1)^2}\right) \quad (4)$$

\vdots

$$w^{n_r}(k) = \mu(y(k)) = \exp\left(\frac{-(y(k) - c^{n_r})^2}{2(\sigma^{n_r})^2}\right) \quad (5)$$

gdzie c^1, \dots, c^{n_r} oznaczają centra poszczególnych zbiorów, natomiast $\sigma^1, \dots, \sigma^{n_r}$ decydują o nachyleniu poszczególnych funkcji Gaussa. Alternatywnie, do wyboru poszczególnych regulatorów lokalnych można zastosować najbardziej aktualny (dostępny) sygnał sterujący procesem, tzn. $u(k-1)$.

Całkowity sygnał sterujący procesem jest sumą warzoną sygnałów wszystkich regulatorów lokalnych

$$u(k) = w^1(k)u^1(k) + \dots + w^{n_r}(k)u^{n_r}(k) \quad (6)$$

Jeżeli funkcje przynależności nie sumują się do 1 (dla dowolnej wartości zmiennej wybierającej), należy zastosować wzór

$$u(k) = \frac{w^1(k)u^1(k) + \dots + w^{n_r}(k)u^{n_r}(k)}{w^1(k) + \dots + w^{n_r}(k)} \quad (7)$$

Warto podkreślić zaletę rozmytego algorytmu DMC w porównaniu z rozmytym algorytmem PID. Nastawy lokalnych algorytmów PID należy dobierać indywidualnie, często metodą wielu prób i błędów. Synteza rozmytego algorytmu DMC jest pod tym względem znacznie uproszczona, ponieważ należy jedynie pozyskać odpowiedzi skokowe z otoczenia wybranych punktów pracy (stosując „małe” wymuszenia, a nie skoki jednostkowe), a następnie zastosować n_r lokalnych algorytmów DMC, każdy z nich korzysta z innej odpowiedzi skokowej. Oczywiście, zarówno w rozmytym algorytmie PID jak i algorytmie DMC, jakość regulacji zależy od liczby, rodzaju i kształtu funkcji przynależności. Z tego też powodu należy zaczynać od możliwie najprostszej struktury regulatora rozmytego ($n_r = 2$).

1.2. Regulacja rozmyta DMC

W algorytmie DMC do predykcji wykorzystuje się model odpowiedzi skokowej, zdefiniowany zestawem D liczb: s_1, \dots, s_D , gdzie D jest horyzontem dynamiki. Wektor zmienionych decyzyjnych w aktualnej dyskretniej chwili $k = 0, 1, 2, \dots$ klasycznego, liniowego regulatora DMC ma postać

$$\Delta U(k) = \begin{bmatrix} \Delta u(k|k) \\ \vdots \\ \Delta u(k + N_u - 1|k) \end{bmatrix} \quad (8)$$

gdzie N_u jest horyzontem sterowania. W wersji analitycznej, bez ograniczeń, optymalny wektor przyrostów oblicza się ze wzoru

$$\Delta U(k) = \mathbf{K}(Y^{\text{zad}}(k) - Y(k) - \mathbf{M}^p(k)\Delta U^p(k)) \quad (9)$$

gdzie macierz o wymiarowości $N_u \times N$ ma postać

$$\mathbf{K} = (\mathbf{M}^T \mathbf{M} + \mathbf{\Lambda})^{-1} \mathbf{M}^T = \begin{bmatrix} \overline{\mathbf{K}}_{1,1} \\ \vdots \\ \overline{\mathbf{K}}_{N_u,1} \end{bmatrix} = \begin{bmatrix} K_{1,1} & \dots & K_{1,N} \\ \vdots & \ddots & \vdots \\ K_{N_u,1} & \dots & K_{N_u,N} \end{bmatrix} \quad (10)$$

Macierz dynamiczna ma wymiary $N \times N_u$ i złożona jest ze współczynników odpowiedzi skokowej

$$\mathbf{M} = \begin{bmatrix} s_1 & 0 & \dots & 0 \\ s_2 & s_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ s_N & s_{N-1} & \dots & s_{N-N_u+1} \end{bmatrix} \quad (11)$$

Druga macierz złożona ze współczynników modelu ma wymiary $N \times (D-1)$ i strukturę

$$\mathbf{M}^p = \begin{bmatrix} s_2 - s_1 & \dots & s_D - s_{D-1} \\ s_3 - s_1 & \dots & s_{D+1} - s_{D-1} \\ \vdots & \ddots & \vdots \\ s_{N+1} - s_1 & \dots & s_{N+D-1} - s_{D-1} \end{bmatrix} = [\mathbf{M}_1^p \dots \mathbf{M}_{D-1}^p]^T \quad (12)$$

Wektory o długości N mają postać

$$Y^{\text{zad}}(k) = y^{\text{zad}}(k) \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}, \quad Y(k) = y(k) \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \quad (13)$$

natomiast wektor o długości $D-1$ ma strukturę

$$\Delta U^p(k) = \begin{bmatrix} u(k-1) \\ \vdots \\ u(k-(D-1)) \end{bmatrix} \quad (14)$$

W wersji oszczędnej algorytmu DMC oblicza się jedynie pierwszy element wektora (8), tzn. przyrost sygnału sterującego dla aktualnej chwili dyskretnej k . Ze wzoru (9) otrzymuje się

$$\Delta u(k|k) = K^e(y^{\text{zad}}(k) - y(k)) - \sum_{i=1}^{D-1} K_i^u \Delta u(k-i) \quad (15)$$

gdzie

$$K^e = \sum_{p=1}^N K_{1,p} \quad (16)$$

$$K_i^u = \overline{\mathbf{K}}_1 \mathbf{M}_i^p, \quad i = 1, \dots, D-1 \quad (17)$$

Skalary $K_{1,p}$ są elementami macierzy (10), wektor \mathbf{K}_1 jest pierwszym wierszem tej macierzy, natomiast \mathbf{M}_i^p są kolejnymi kolumnami macierzy (12).

Baza reguł rozmytego regulatora DMC ma postać

$$\text{Reguła 1: jeżeli } y(k) \in Y^1, \text{ to } u^1(k) = K^{e,1}(y^{\text{zad}}(k) - y(k)) \quad (18)$$

$$- \sum_{i=1}^{D-1} K_i^{u,1} \Delta u(k-i) + u(k-1) \quad (19)$$

\vdots

$$\text{Reguła } n_r: \text{ jeżeli } y(k) \in Y^{n_r}, \text{ to } u^{n_r}(k) = K^{e,n_r}(y^{\text{zad}}(k) - y(k)) \quad (20)$$

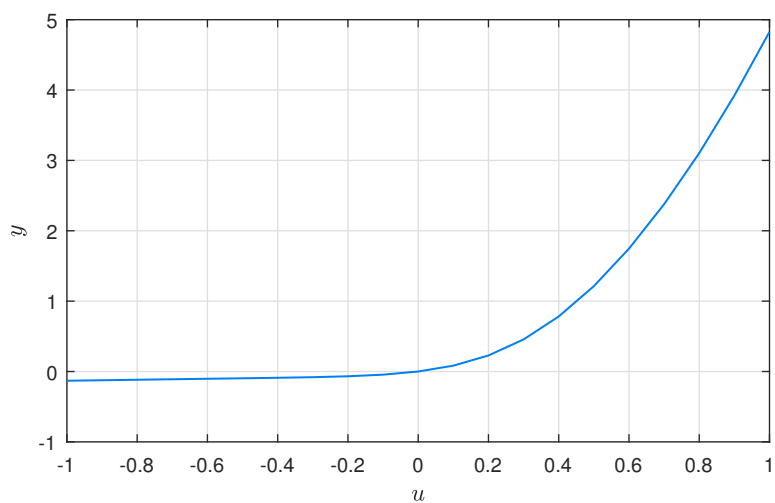
$$- \sum_{i=1}^{D-1} K_i^{u,n_r} \Delta u(k-i) + u(k-1) \quad (21)$$

Uwaga: w prawach regulacji regulatorów lokalnych, po prawej stronie, występują sygnały $y(k)$, $\Delta u(k-i)$, $u(k-1)$. Pierwszy regulator lokalny ma parametry: $K^{e,1}$ i $K_i^{u,1}$ (dla $i = 1, \dots, D-1$), drugi – parametry: $K^{e,2}$ i $K_i^{u,2}$, ostatni – parametry K^{e,n_r} i K_i^{u,n_r} .

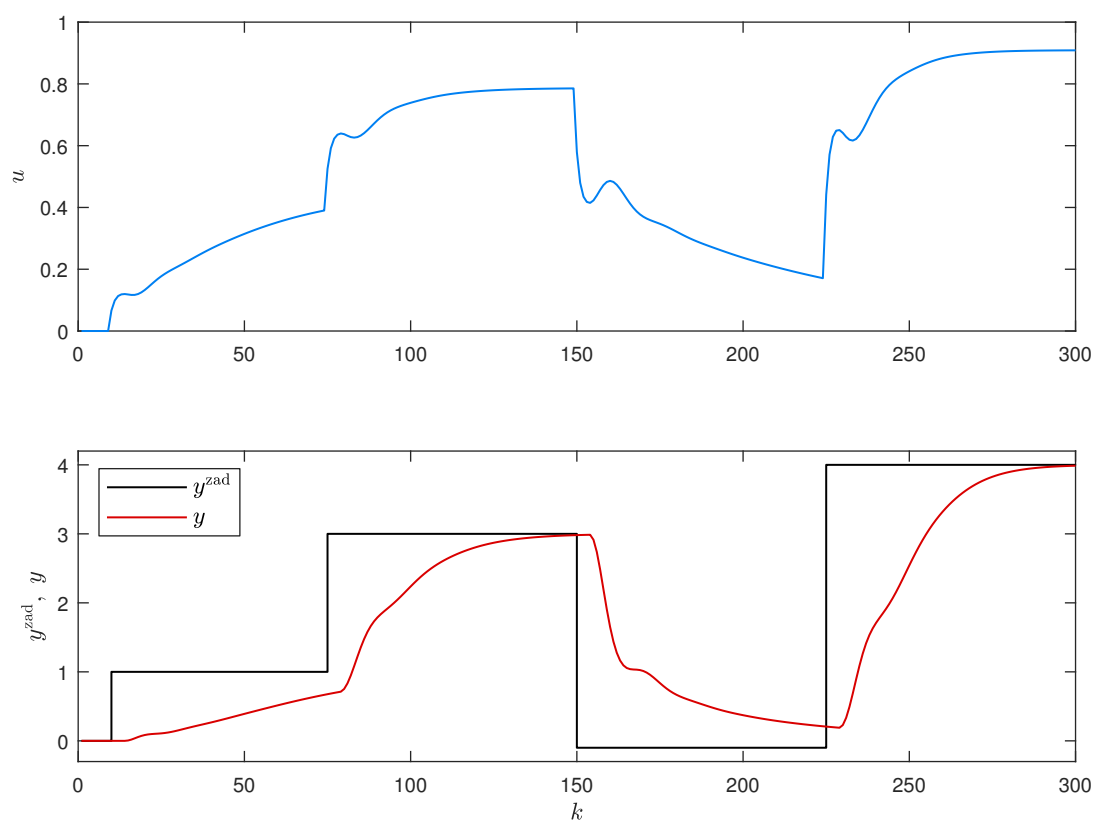
1.3. Wyniki symulacji

Charakterystyka statyczna pewnego procesu nieliniowego została przedstawiona na rys. 1. Należy zaprojektować algorytm DMC oraz rozmyty algorytm DMC i przetestować je przy kilku, różnych zmianach wartości zadanej, z możliwie szerokiego zakresu.

Wyniki klasycznego algorytmu DMC przedstawiono na rys. 2. Algorytm bazuje na modelu odpowiedzi skokowej z punktu pracy $u = 0$. Przyjęto parametry: $D = 50$, $N = 50$, $N_u = 2$, $\lambda = 200$. Jakość regulacji jest niska. Algorytm działa wolno, dwie wartości zadane nie są osiągnęte.

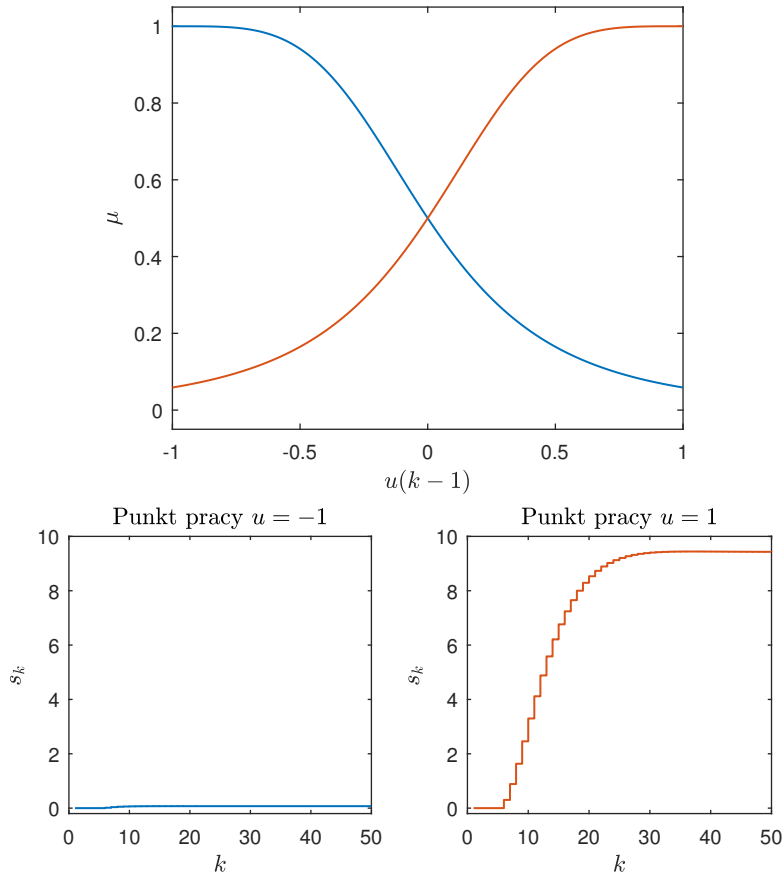


Rys. 1. Nieliniowa charakterystyka statyczna procesu

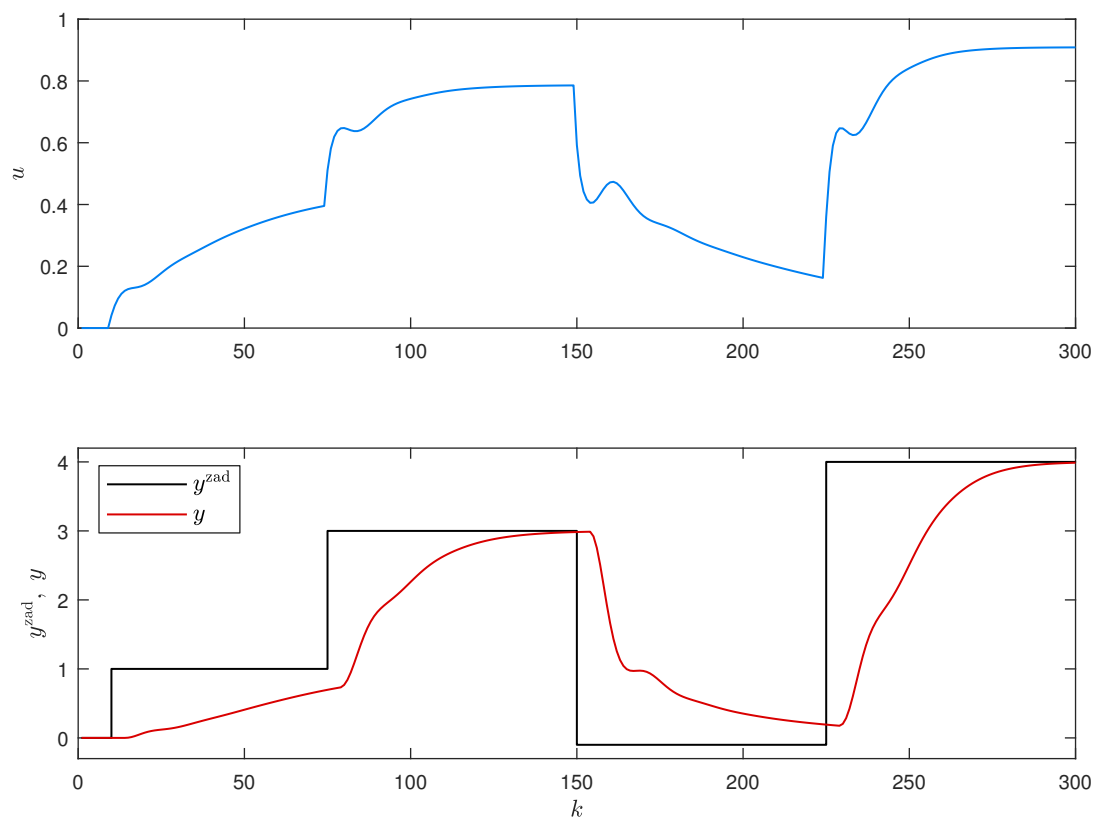


Rys. 2. Wyniki symulacji algorytmu DMC

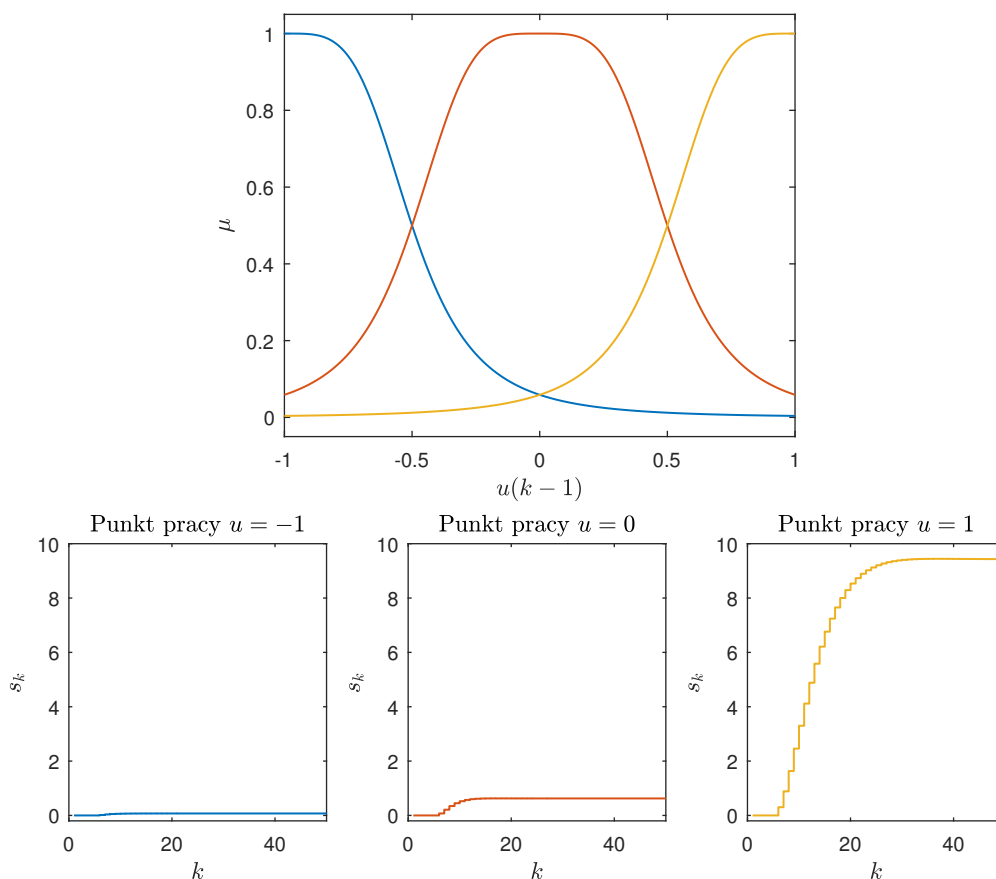
Na początku rozważmy rozmyty algorytm DMC, złożony z dwóch ($n_r = 2$) regulatorów lokalnych. Na rys. 3 przedstawiono funkcje przynależności oraz lokalne odpowiedzi skokowe. Punkt pracy uzależniono od sygnału $u(k-1)$. Wyniki symulacji rozmytego algorytmu DMC pokazano na rys. 4. W porównaniu z klasycznym algorytmem DMC uzyskano bardzo niewielką poprawę jakości regulacji.



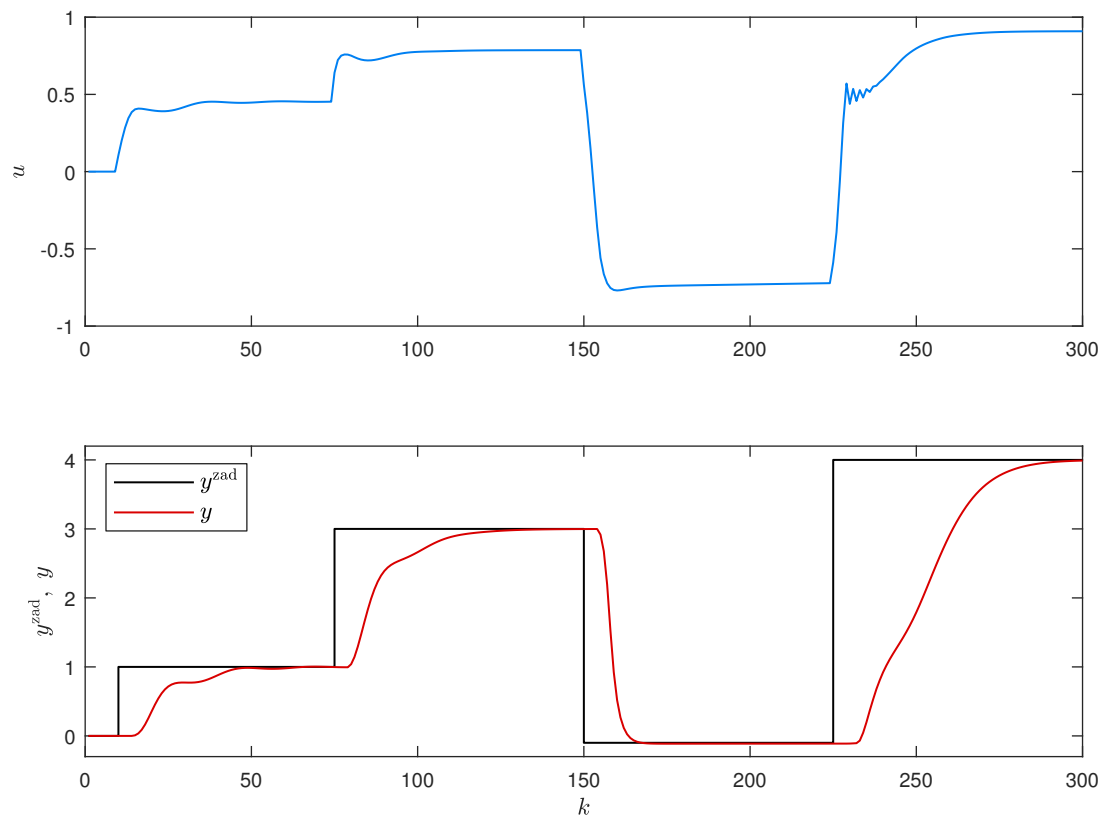
Rys. 3. Zastosowane funkcje przynależności oraz lokalne odpowiedzi skokowe dla $n_r = 2$

Rys. 4. Wyniki symulacji rozmytego algorytmu DMC dla $n_r = 2$

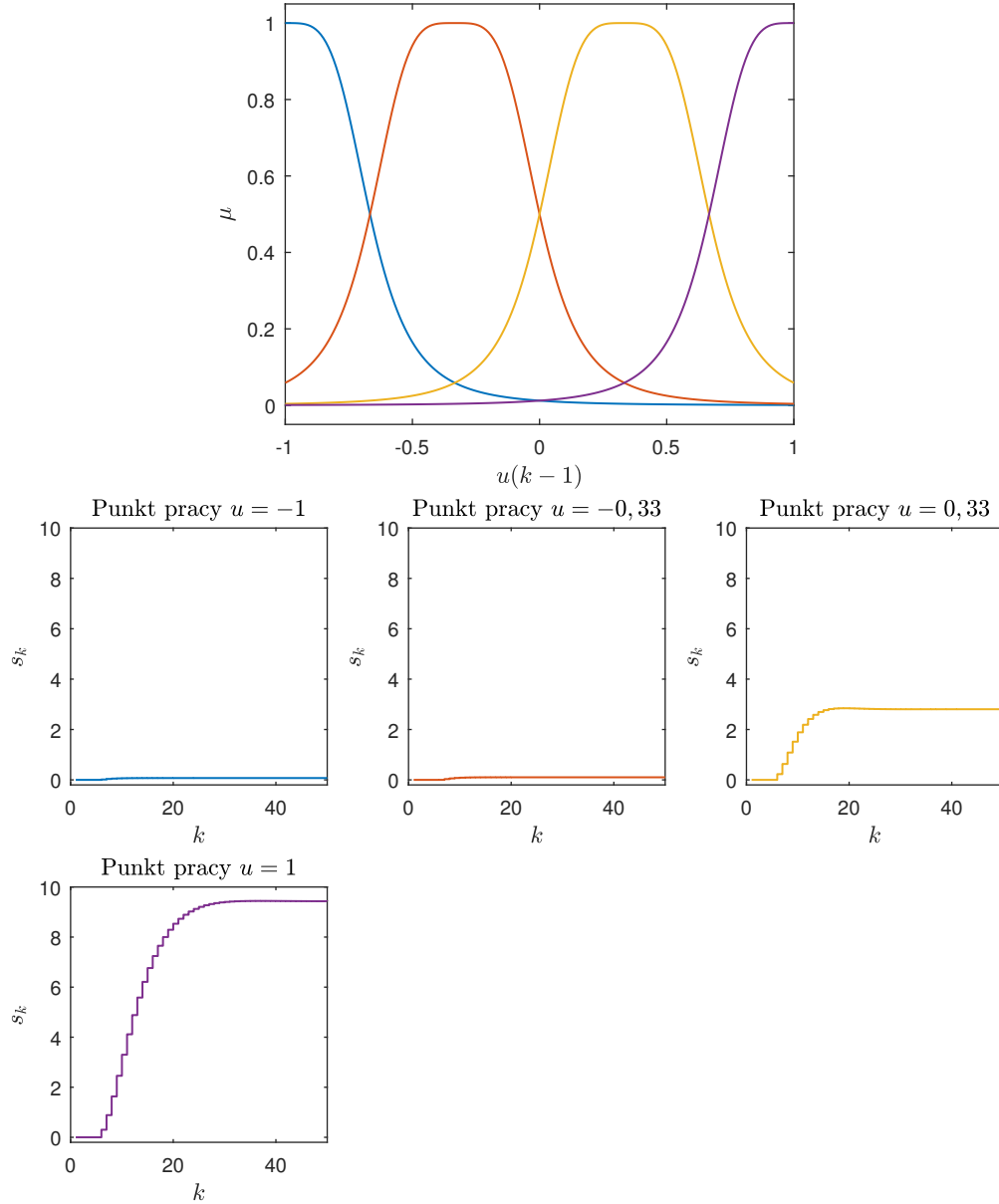
Następnie został zaprojektowany rozmyty algorytm DMC, złożony z trzech ($n_r = 3$) regulatorów lokalnych. Na rys. 5 przedstawiono funkcje przynależności oraz lokalne odpowiedzi skokowe. Wyniki symulacji rozmytego algorytmu DMC pokazano na rys. 6. W porównaniu z klasycznym algorytmem DMC oraz algorytmem rozmytym złożonym z dwóch regulatorów lokalnych, uzyskano istotną poprawę jakości regulacji. Sygnał wyjściowy procesu szybko dąży do nowej wartości zadanej. Niepokojącym zjawiskiem są szybko tłumione oscylacje przy ostatniej zmianie punktu pracy. Mogą być one wyeliminowane przy zwiększeniu współczynnika λ wybranych regulatorów lokalnych.



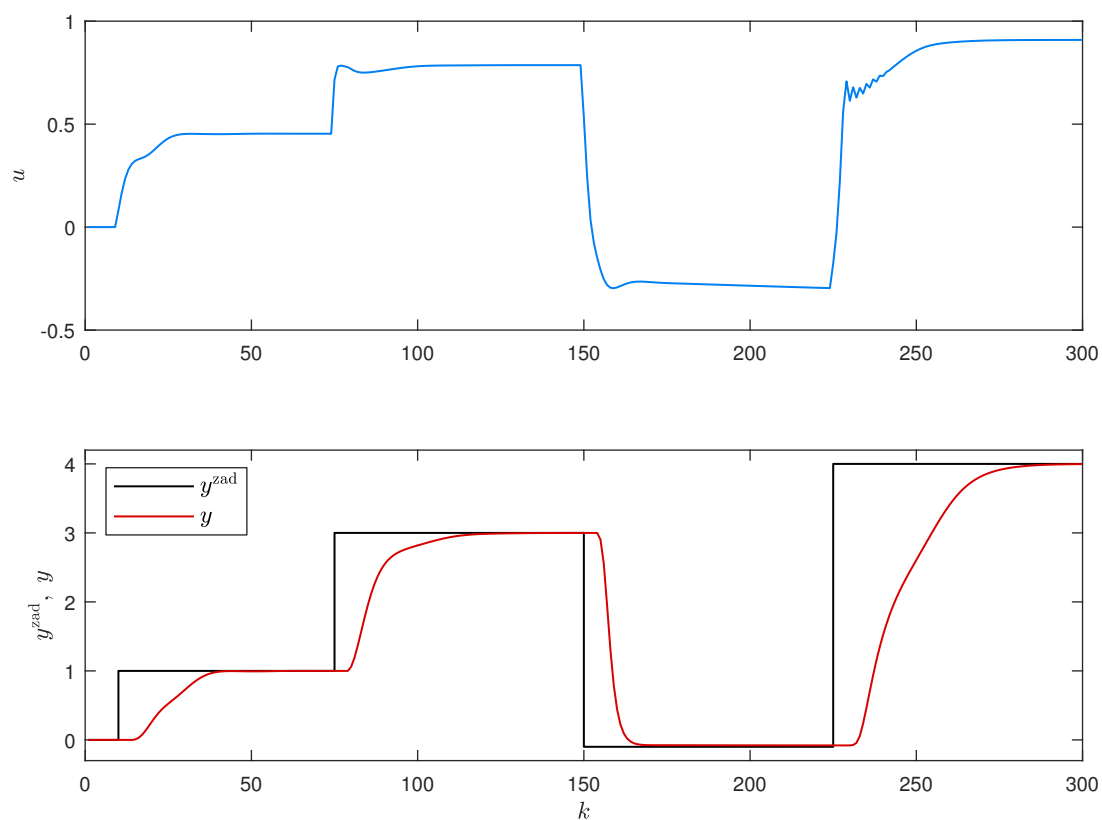
Rys. 5. Zastosowane funkcje przynależności oraz lokalne odpowiedzi skokowe dla $n_r = 3$

Rys. 6. Wyniki symulacji rozmytego algorytmu DMC dla $n_r = 3$

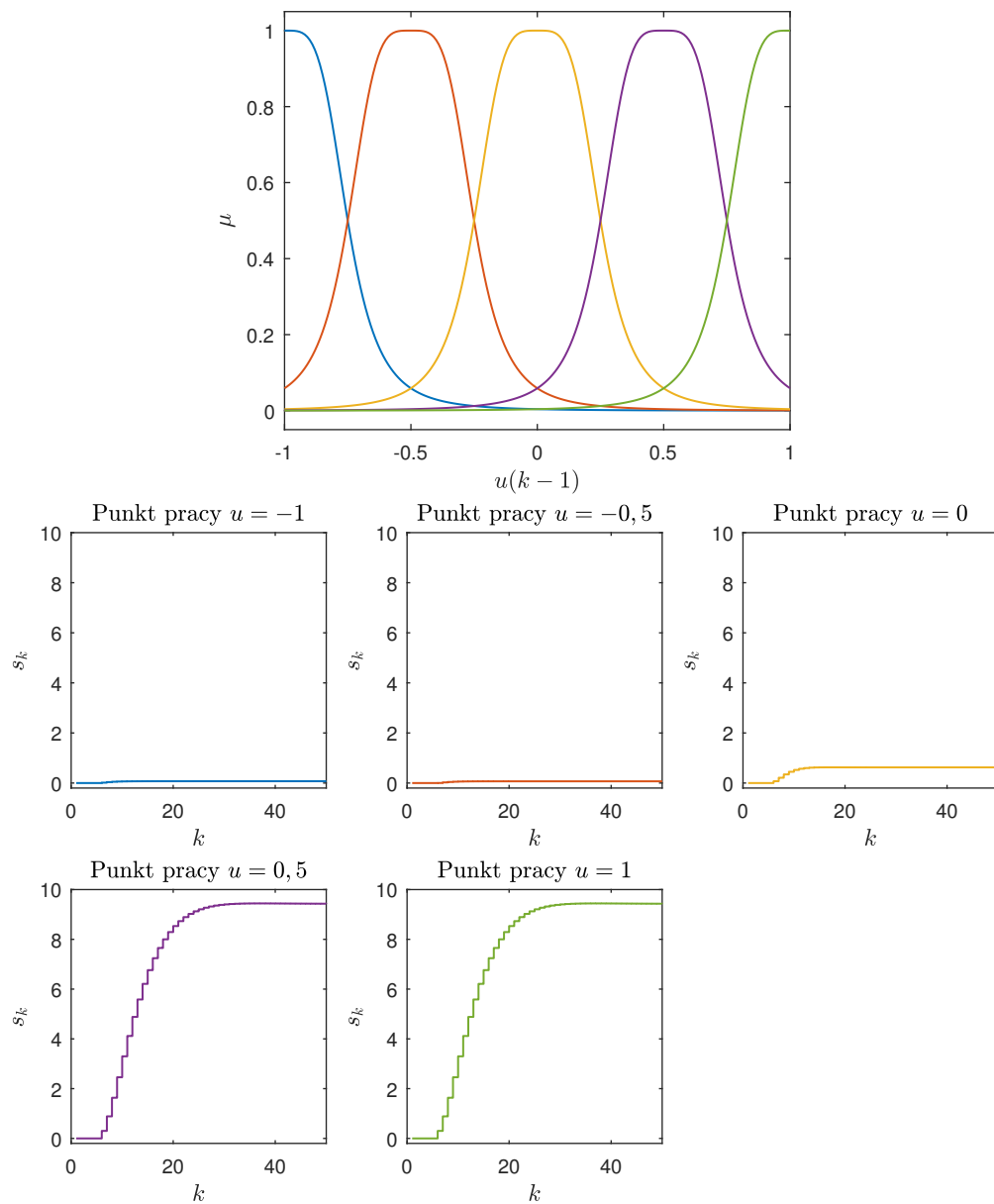
Rozmyty algorytm DMC, złożony z czterech ($n_r = 4$) regulatorów lokalnych wykorzystuje funkcje przynależności oraz lokalne odpowiedzi skokowe pokazane na rys. 7. Wyniki symulacji rozmytego algorytmu DMC pokazano na rys. 8. Zwiększenie liczby funkcji przynależności pozwoliło jeszcze zwiększyć jakość regulacji.



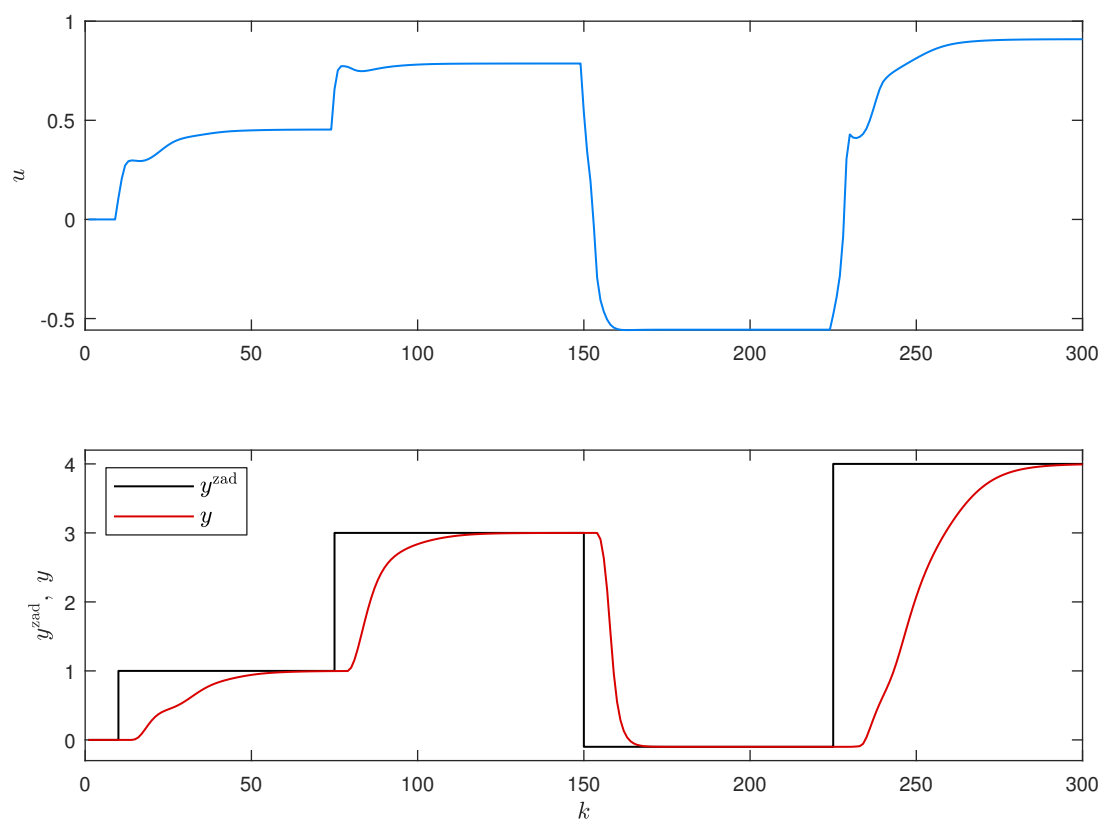
Rys. 7. Zastosowane funkcje przynależności oraz lokalne odpowiedzi skokowe dla $n_r = 4$

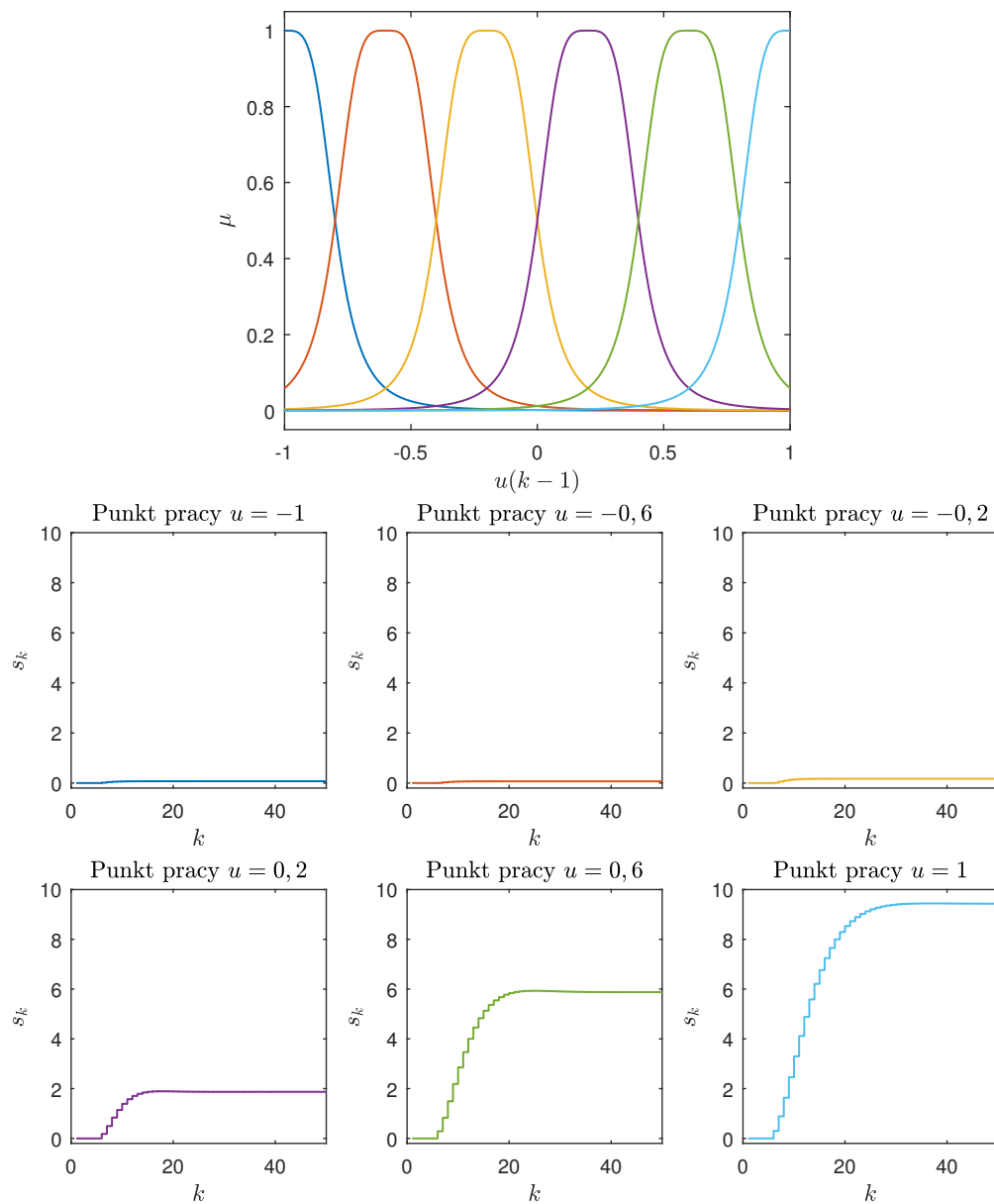
Rys. 8. Wyniki symulacji rozmytego algorytmu DMC dla $n_r = 4$

Zaprojektowano również rozmyty algorytm DMC, złożony z pięciu ($n_r = 5$) oraz sześciu ($n_r = 6$) regulatorów lokalnych. Wykorzystywane funkcje przynależności oraz lokalne odpowiedzi skokowe pokazano na rys. 9 i 11. Wyniki symulacji algorytmów przedstawiono na rys. 10 i 12. Zwiększenie liczby regulatorów lokalnych może jeszcze poprawić jakość regulacji, ale poprawa ta jest niewielka. Projektant musi uwzględnić kompromis między złożonością algorytmu (liczbą regulatorów lokalnych) a jakością regulacji.

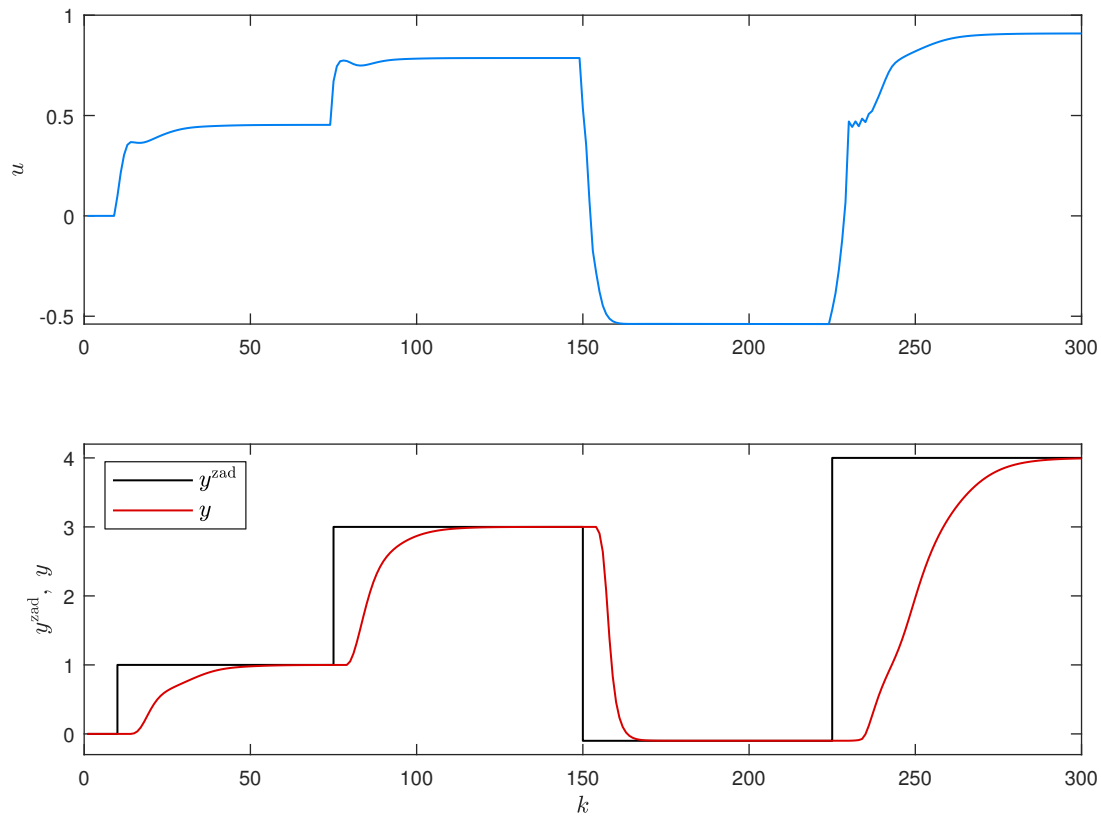


Rys. 9. Zastosowane funkcje przynależności oraz lokalne odpowiedzi skokowe dla $n_r = 5$

Rys. 10. Wyniki symulacji rozmytego algorytmu DMC dla $n_r = 5$



Rys. 11. Zastosowane funkcje przynależności oraz lokalne odpowiedzi skokowe dla $n_r = 6$

Rys. 12. Wyniki symulacji rozmytego algorytmu DMC dla $n_r = 6$

We wszystkich rozważanych lokalnych algorytmach DMC zastosowano te same parametry. Aby zwiększyć jakość regulacji warto niezależnie dobrać współczynniki λ dla poszczególnych regulatorów. Można to zrobić metodą prób i błędów albo na drodze optymalizacji numerycznej (pod warunkiem znajomości modelu procesu do symulacji). W opisanym rozwiązaniu zastosowano równomierny podział zakresu zmienności punktu pracy (sygnał $u(k-1)$), w praktyce natomiast zamiast nadmiernie zwiększać liczbę regulatorów lokalnych, warto dobrać nierównomierne, czasami również niesymetryczne funkcje przynależności.