Wydział Elektroniki i Technik Informacyjnych
Politechnika Warszawska

Projektowanie układów sterowania (projekt grupowy)

Sprawozdanie z projektu nr 4 zespół nr 9

Paweł Bugyi, Marcin Michalski, Krzysztof Pierczyk

Warszawa, 2020

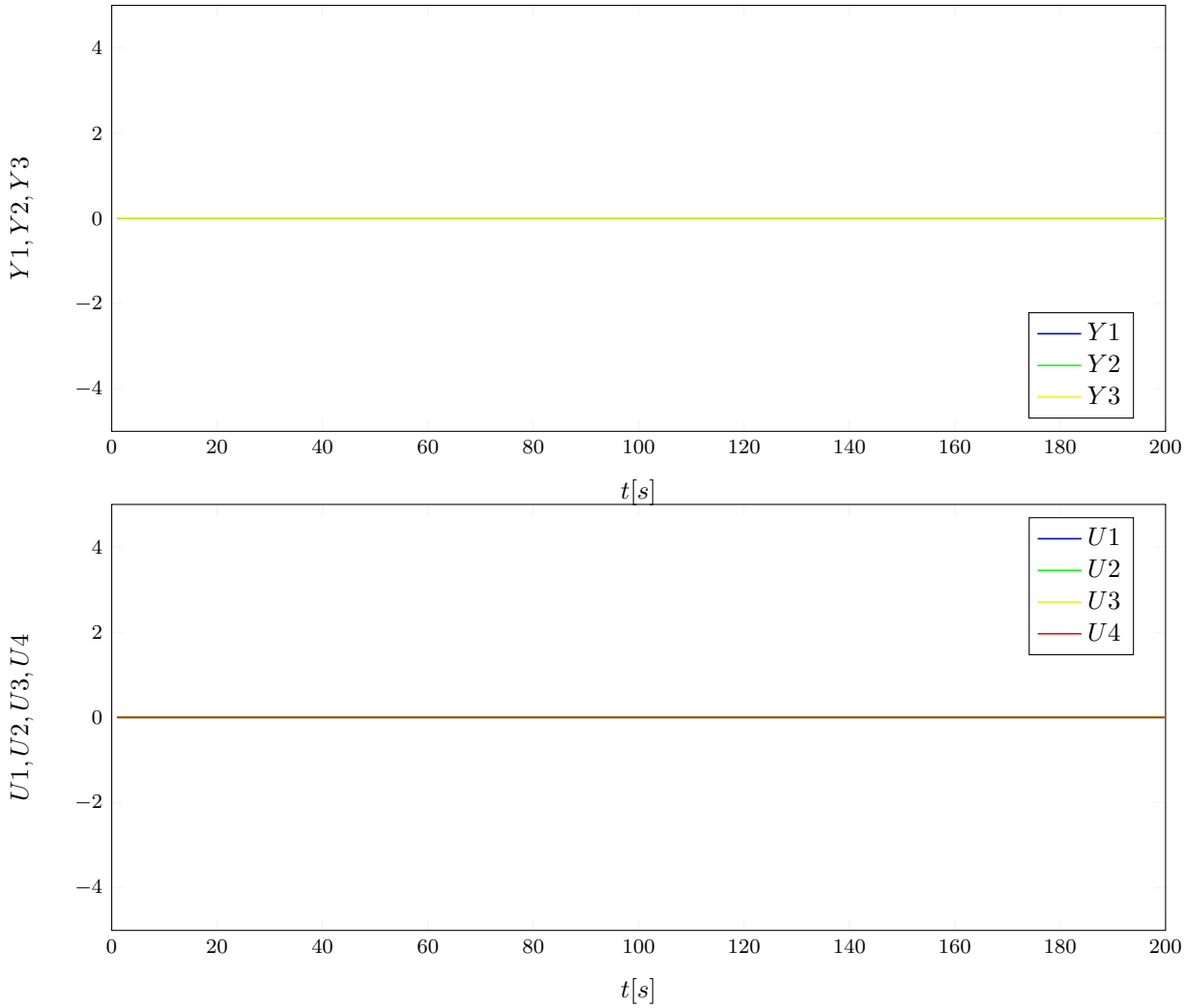
Spis treści

1. Wstęp

Czwarty i zarówno ostatni projekt z Projektowania Układów Sterowania (PUST) stanowi zwieńczenie przedmiotu. Głównym celem w projekcie jest regulacja obiektu typu **MIMO** (Multiple Inputs Multiple Outputs), czyli obiektu o wielu wejściach i o wielu wejściach. W naszym przypadku jest to obiekt o 4 wejściach i 3 wyjściach. Do regulacji wykorzystujemy dwa algorytmy regulacji PID i DMC, dzięku czemu będziemy mogli ocenić możliwości każdego z algorytmów, by w przyszłości wiedzieć, który regulator pozwoli na lepszą jakość regulacji. Poniższe sprawozdanie stanowi podsumowanie przeprowadzonych przez nas eksperymentów zarówno za dnia jak i w nocy.

2. Sprawdzenie punktu pracy

Przed rozpoczęciem eksperymentów musimy ustalić dozwolony punkt pracy. Zgodnie z treścią projektu jest to punkt dla sterowania U = [0,0,0,0] (taki zapis oznacza, że U1 = 0, U2 = 0, U3 = 0 i U4 = 0) i wyjście obiektu powinno się stabilizować na wartości Y = [0,0,0] (zapis analogiczny jak w przypadku sterowania).

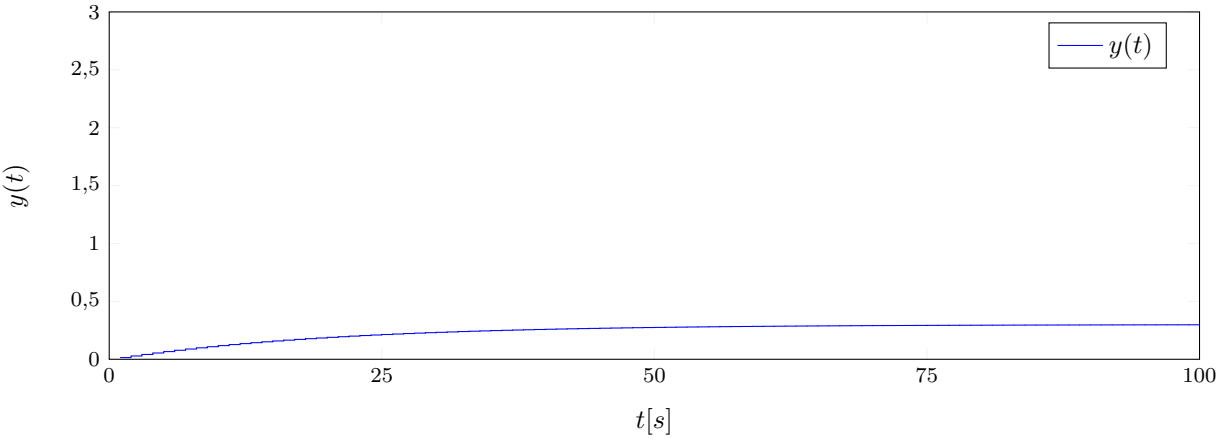


Rys. 2.1. Punkt pracy

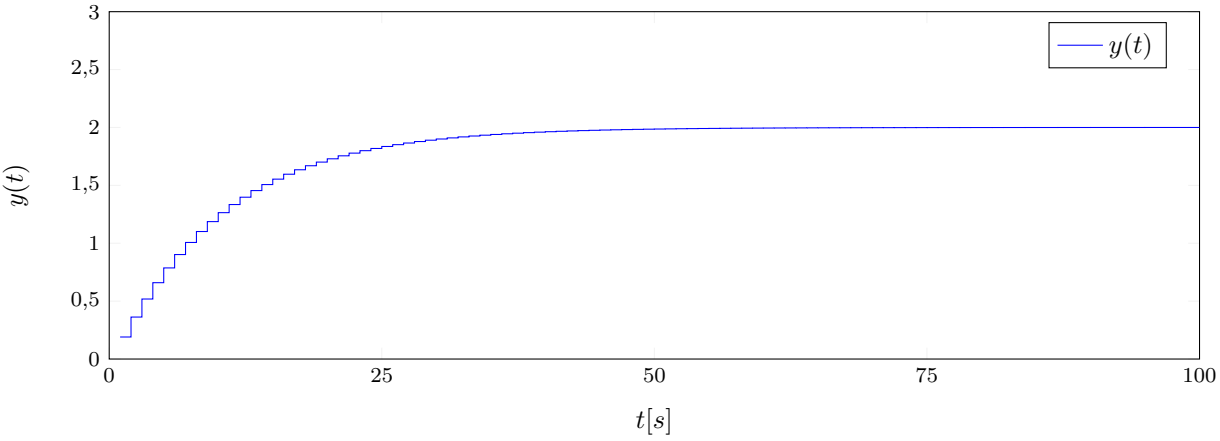
Jak widzimy podany punkt pracy w projekcie jest poprawny i obiekt się stabilizuje na podanych wartościach.

3. Odpowiedzi skokowe

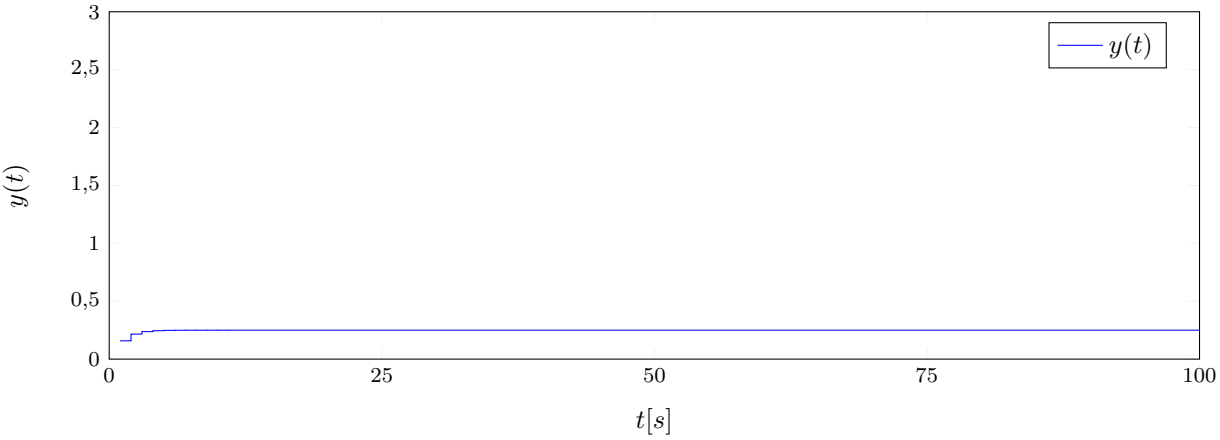
Nasz obiekt posiada 4 wejścia i 3 wyjścia i każde wejście oddziału na wszystkie wyjścia. Oznacza to, że potrzebujemy 12 odpowiedzi skokowych, żeby otrzymać pełną informację o obiekcie. Odpowiedzi skokowe są otrzymywane poprzez ustawienie jednego wejścia na wartość równą 1, a pozostałych na 0. Zgodnie ze sztuką, ponieważ skok sterowania następuje w pierwszej próbce to dopiero od drugiej próbki zaczynamy zbierać odpowiedź skokową. W naszym przypadku skoki są wykonywane z punktu pracy wymienionego w punkcie 2 i wielkość skoku wynosi 1 co powoduje, że nie musimy przeskalowywać odpowiedzi skokowej.



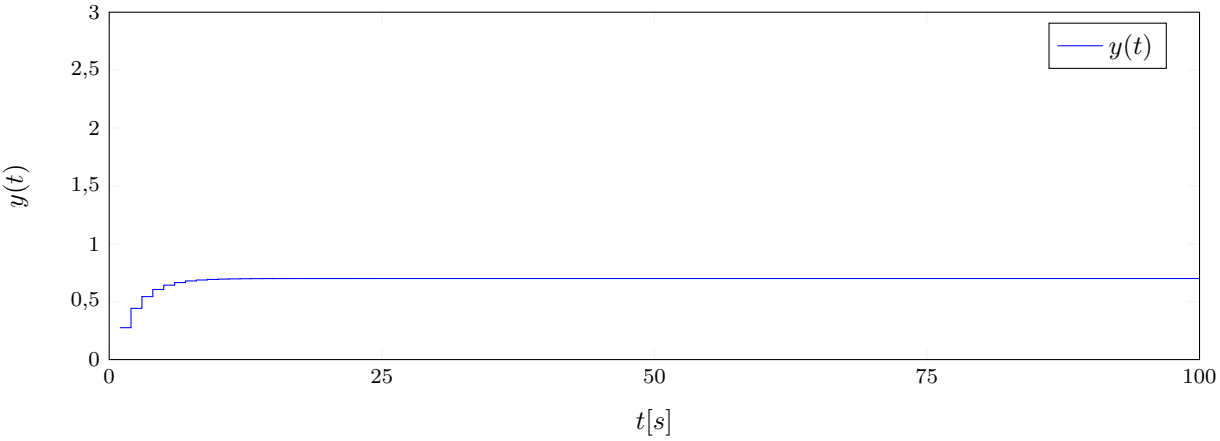
Rys. 3.1. Odpowiedź skokowa U1-Y1



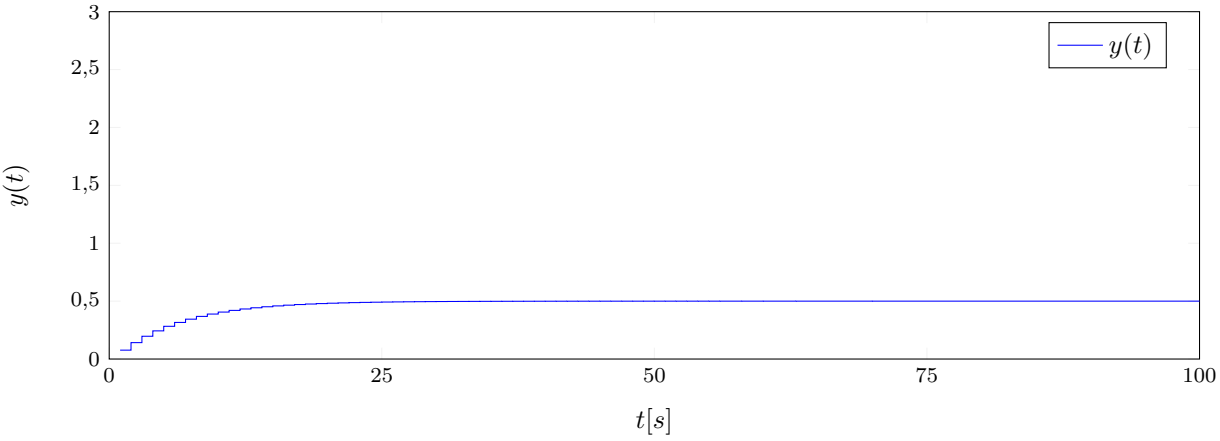
Rys. 3.2. Odpowiedź skokowa U1-Y2



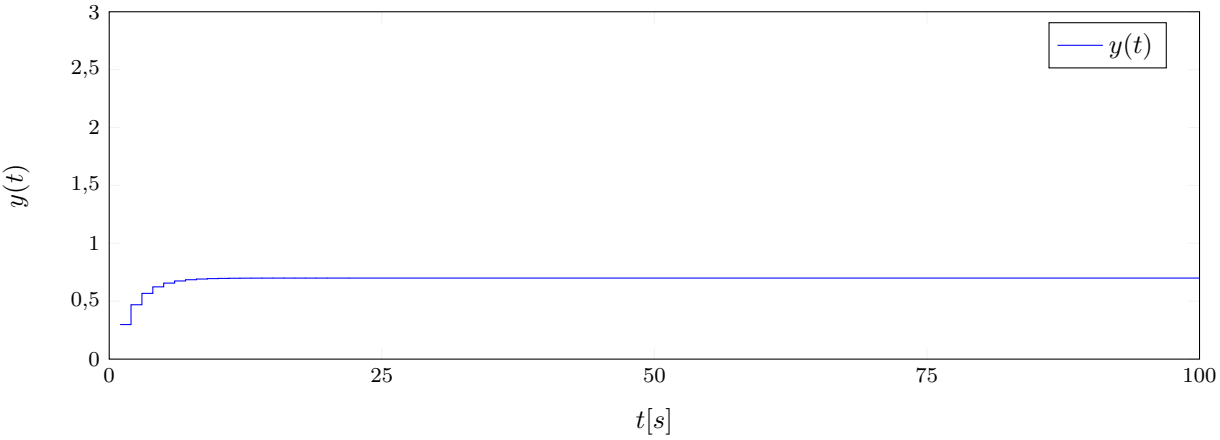
Rys. 3.3. Odpowiedź skokowa U1-Y3



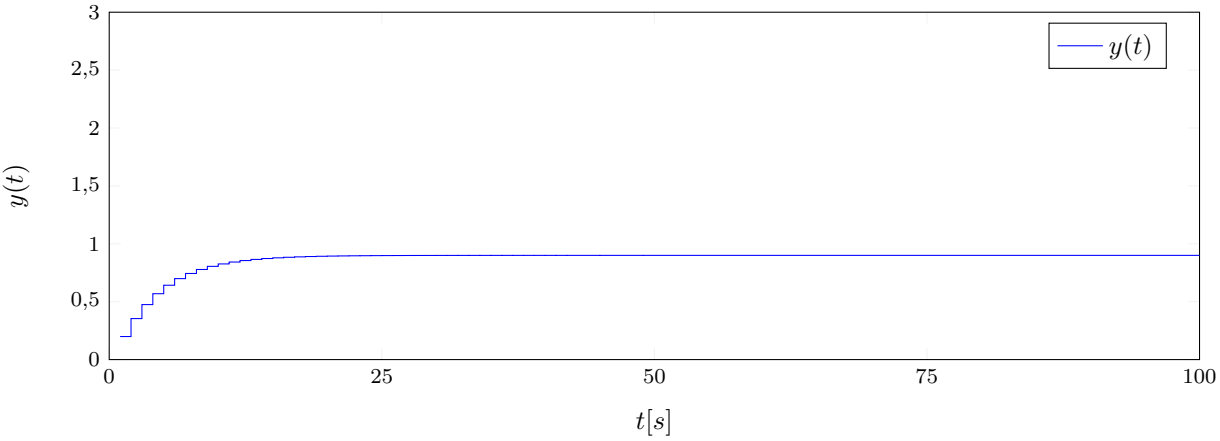
Rys. 3.4. Odpowiedź skokowa U2-Y1



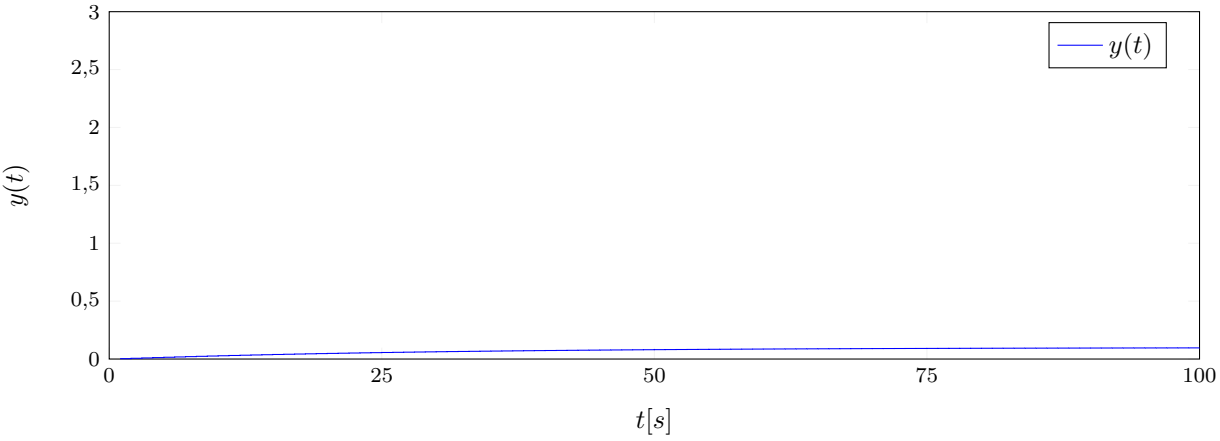
Rys. 3.5. Odpowiedź skokowa U2-Y2



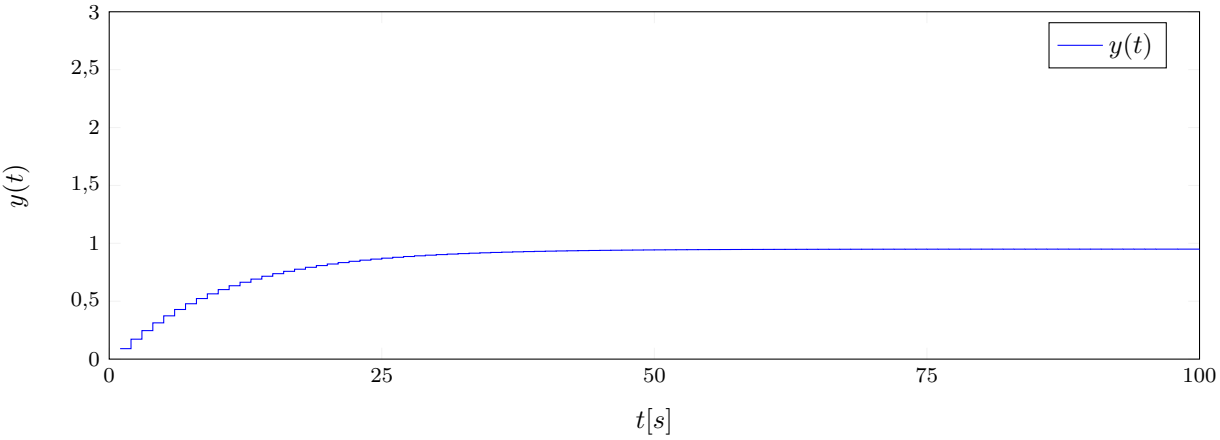
Rys. 3.6. Odpowiedź skokowa U2-Y3



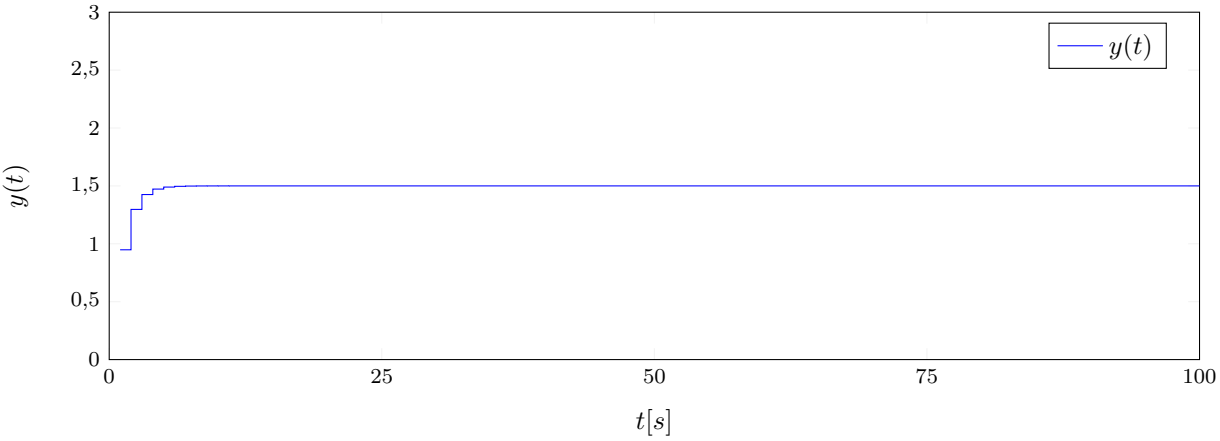
Rys. 3.7. Odpowiedź skokowa U3-Y1



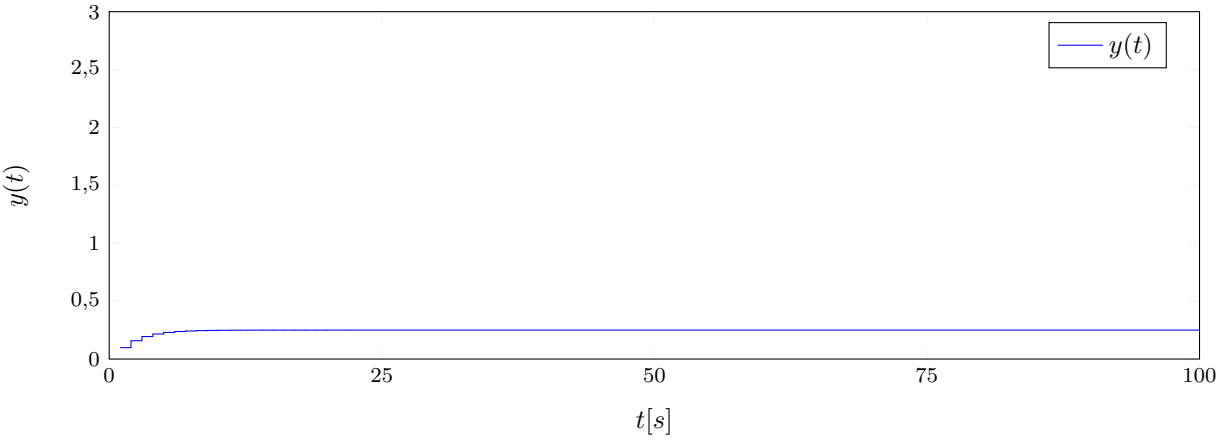
Rys. 3.8. Odpowiedź skokowa U3-Y2



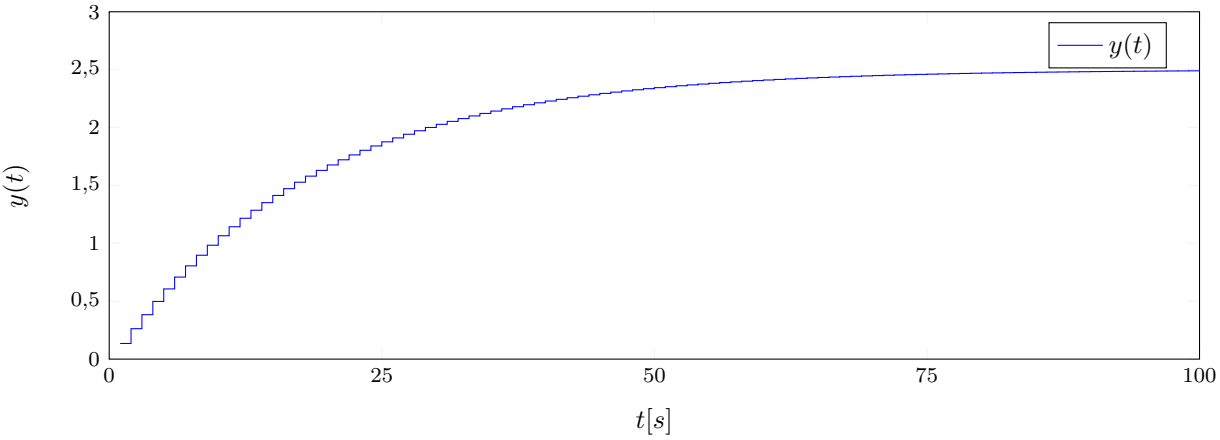
Rys. 3.9. Odpowiedź skokowa U3-Y3



Rys. 3.10. Odpowiedź skokowa U4-Y1



Rys. 3.11. Odpowiedź skokowa U4-Y2



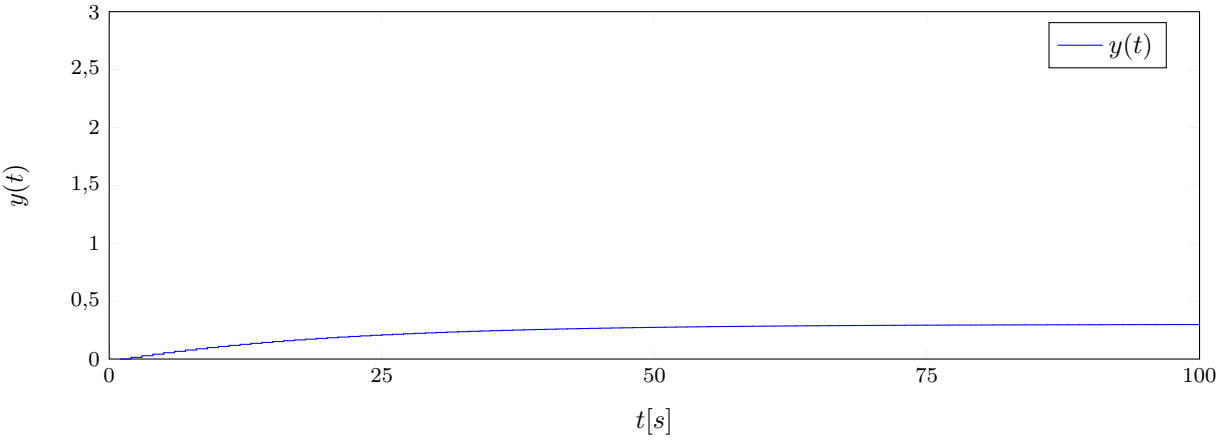
Rys. 3.12. Odpowiedź skokowa U4-Y3

Nasz algorytm DMC posiada tylko jeden parametr D (horyzont dynamiki), co oznacza, że wszystkie odpowiedzi skokowe będą tej samej długości i równej parametrowi D.

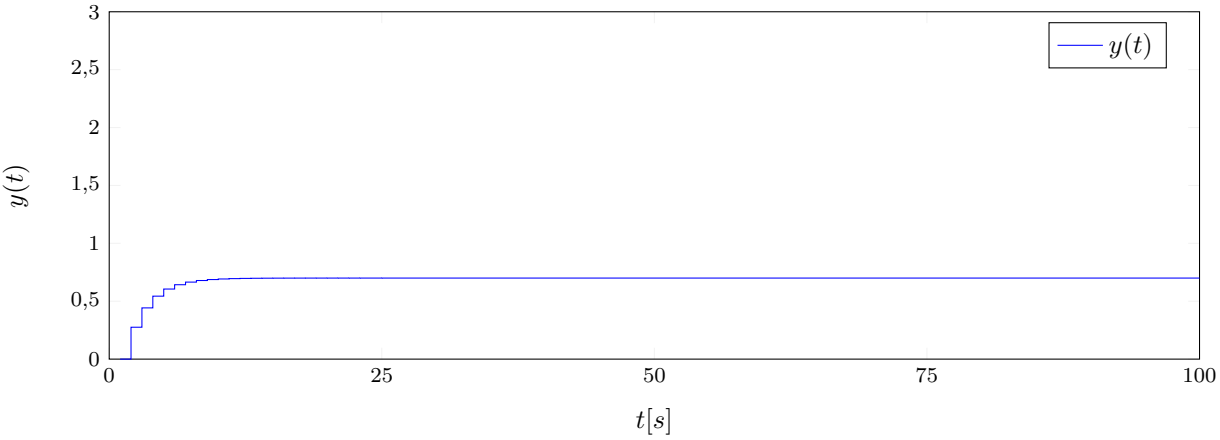
4. PID

4.1. Algorytm PID w wersji MIMO

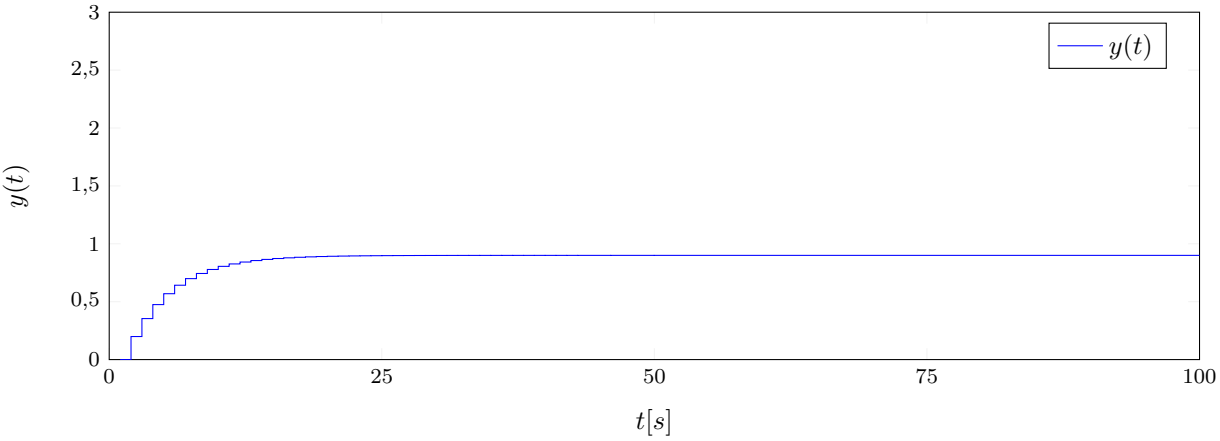
Algorytm PID w wersji MIMO składa się z kilku algorytmów PID w wersji SISO (kilku w tym wypadku oznacza ilość wyjść regulowanego obiektu). Regulatory PID sa zaimplementowane w wersji przyrostowej. Obiekt regulowany ma 4 wejścia, a tylko 3 wyjścia co powoduje, że będziemy mogli sterować tylko 3 wejściami z 4. Zgodnie ze skryptem powiniśmy wybrać dla każdego wyjścia takie wejście, które ma największy wpływ na dane wyjście. W naszym eksperymencie postanowiliśmy lekko zmodyfikować tę regułę w taki sposób, że wybieramy dla każdego wyjścia wejście o największym wpływie na to wyjście i możliwie minimalnym wpływie na pozostałe wyjścia. To podejście pozwoli nam w pewnym stopniu zamienić obiektu typu MIMO na kilka obiektów typu SISO. Jest to bardzo ważne, gdyż algorytm regulacji PID w wersji MIMO składa się z niezależnych regulatorów PID w wersji SISO, więc taki algorytm najlepiej będzie działał , gdy obiekt MIMO można zamienic na kilka obiektów SISO.



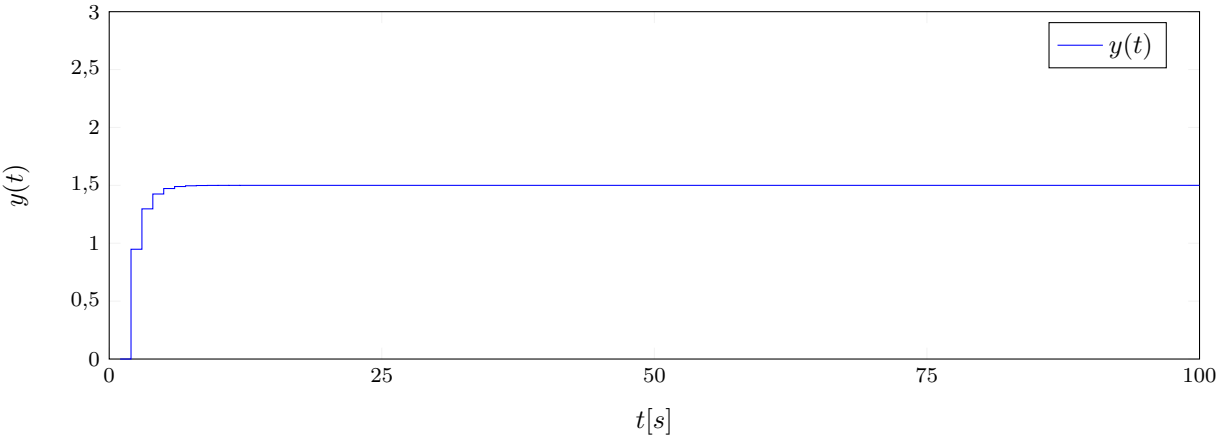
Rys. 4.1. Wpływ wejścią U1 na Y1



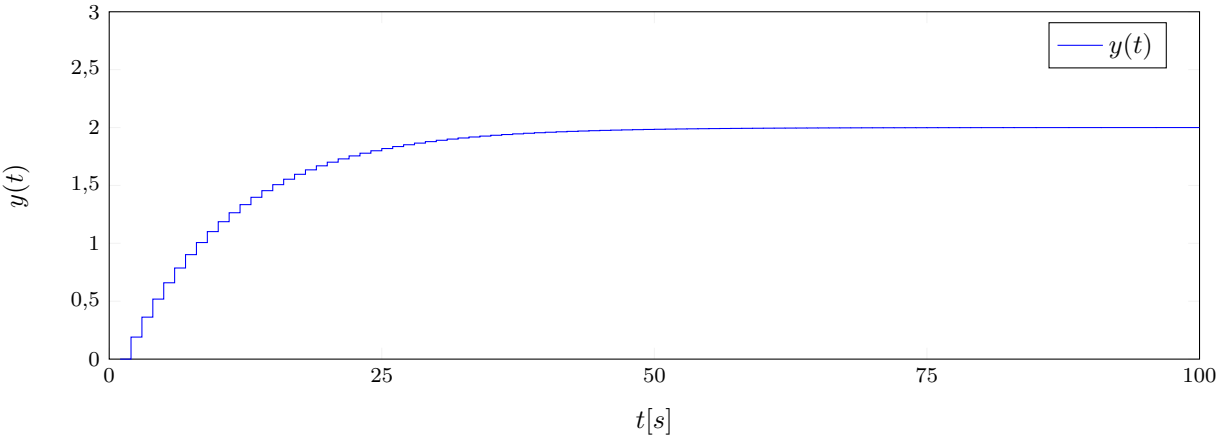
Rys. 4.2. Wpływ wejścią U2 na Y1



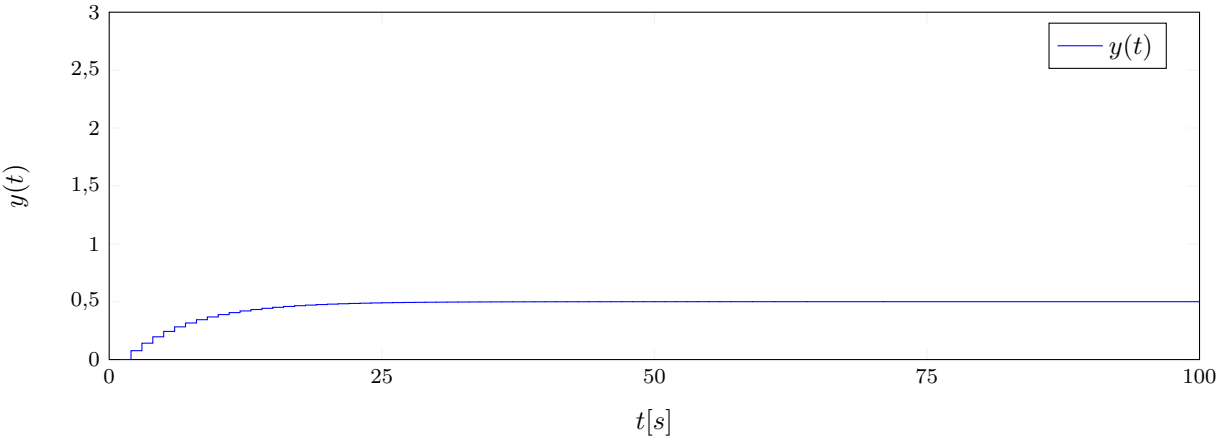
Rys. 4.3. Wpływ wejścią U3 na Y1



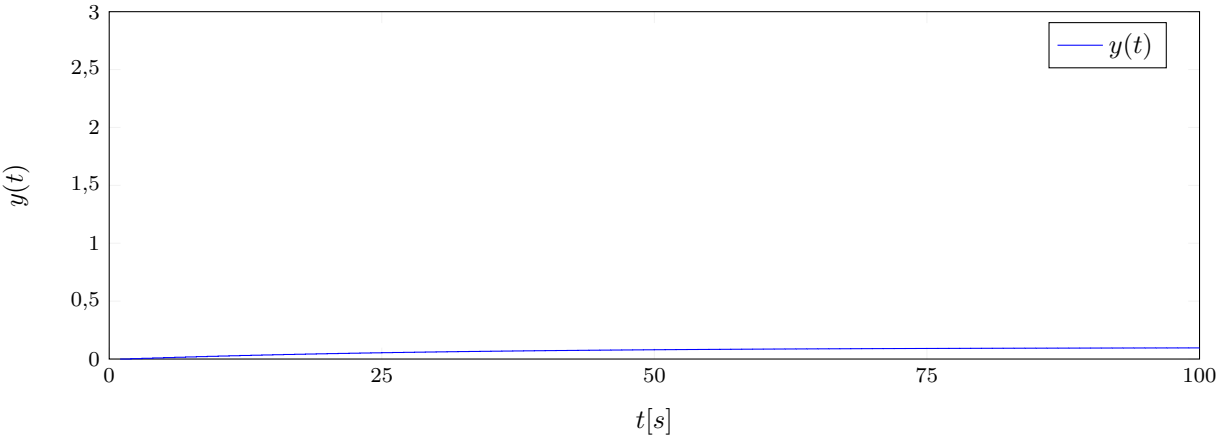
Rys. 4.4. Wpływ wejścią U4 na Y1



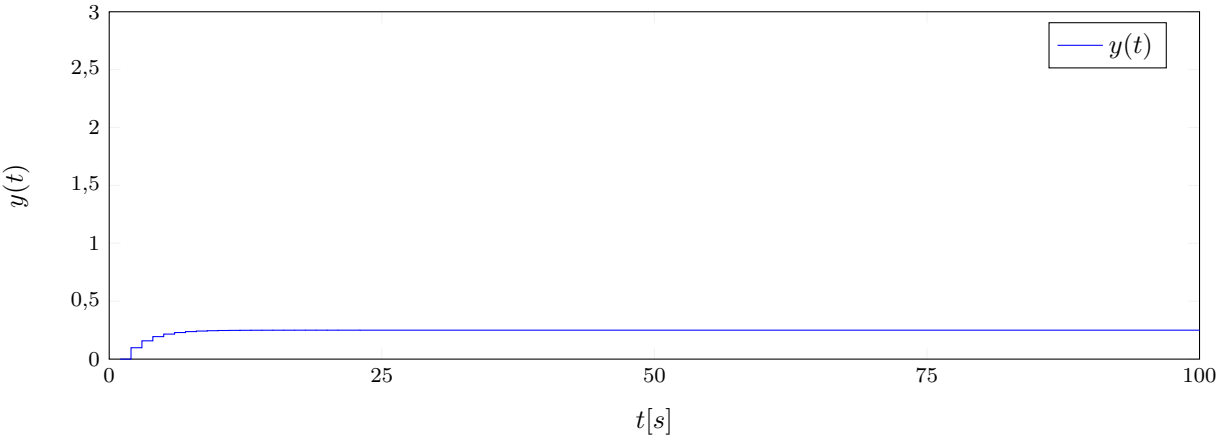
Rys. 4.5. Wpływ wejścią U1 na Y2



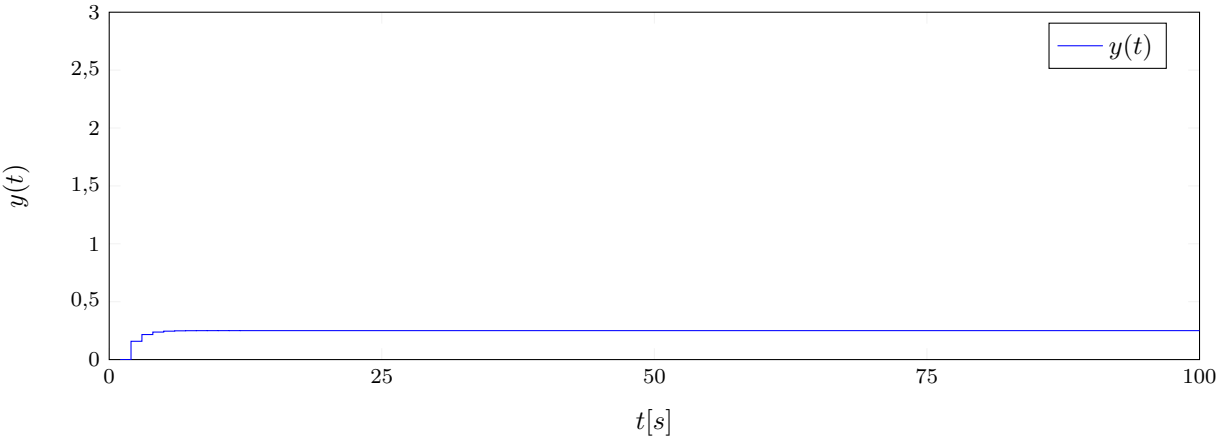
Rys. 4.6. Wpływ wejścią U2 na Y2



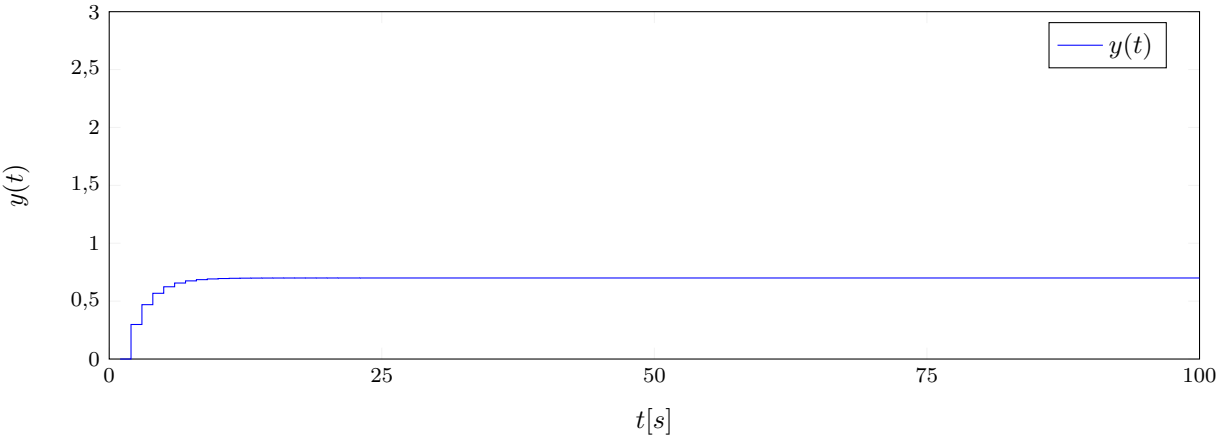
Rys. 4.7. Wpływ wejścią U3 na Y2



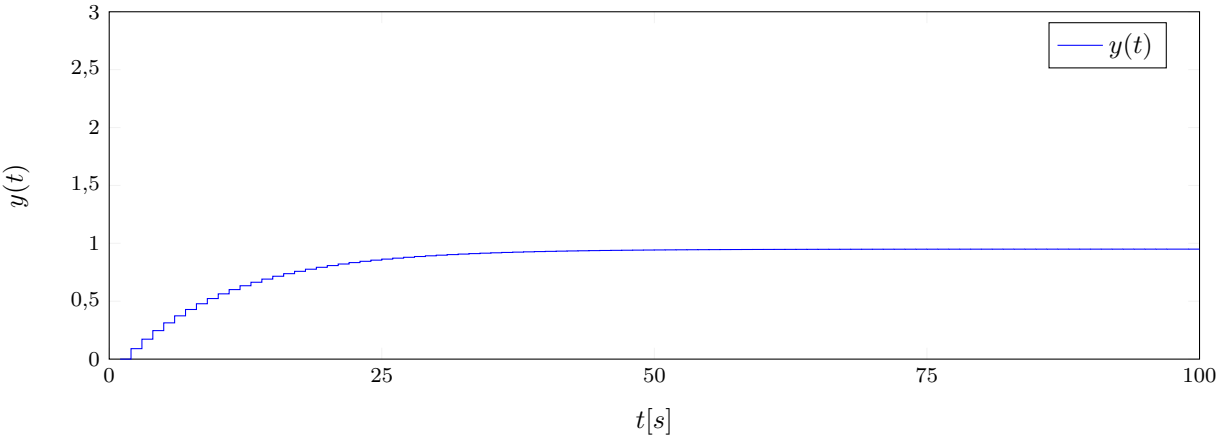
Rys. 4.8. Wpływ wejścią U4 na Y2



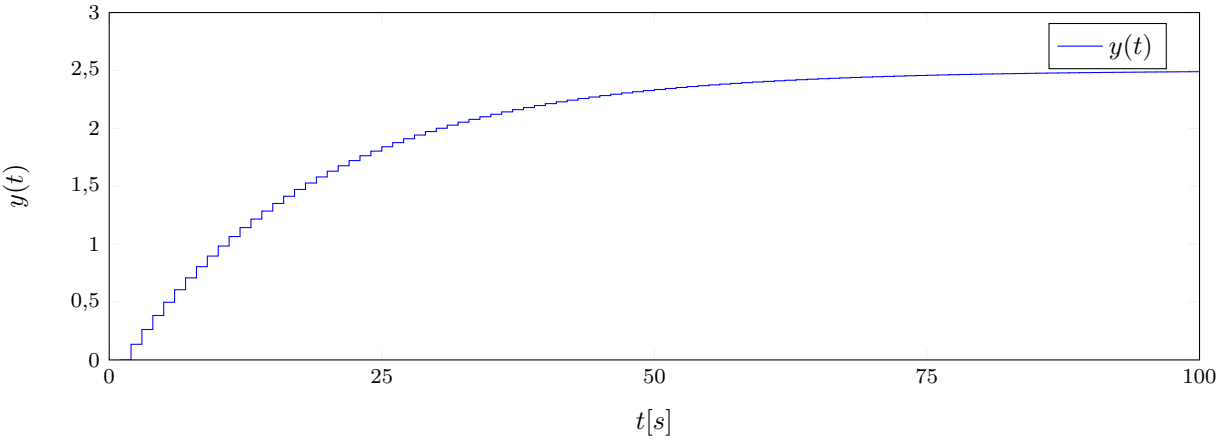
Rys. 4.9. Wpływ wejścią U1 na Y3



Rys. 4.10. Wpływ wejścią U2 na Y3



Rys. 4.11. Wpływ wejścią U3 na Y3



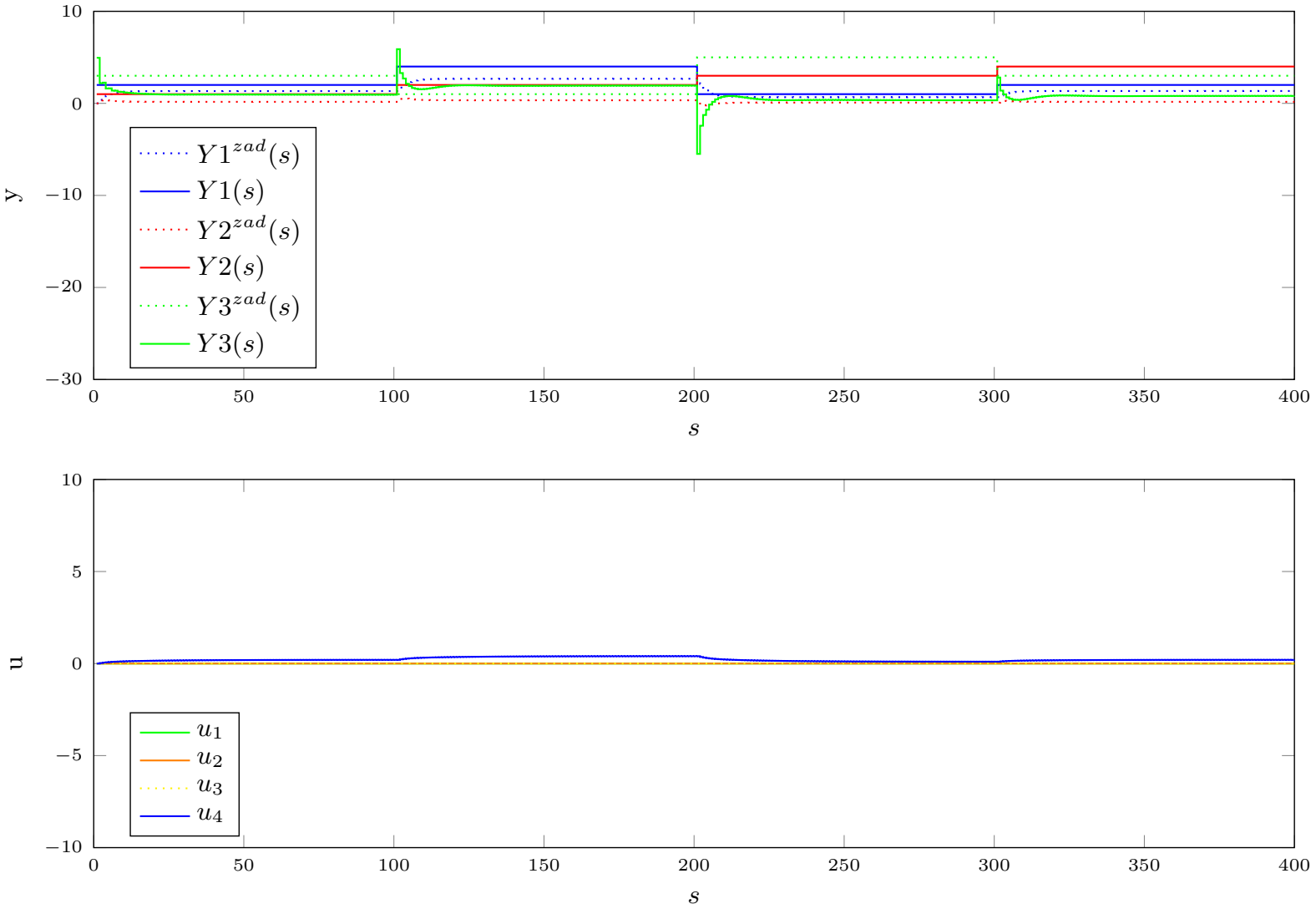
Rys. 4.12. Wpływ wejścią U4 na Y3

Zgodnie z przyjętą przez nas regułą wybieramy 3 następujące pary wejście-wyjście (para wejście-wyjście oznacza, że regulator będzie generował sterowanie na wybranym wejściu , a uchyb pobierał z wybranego wyjścia do obliczania sterowania):
1. Pierwszy PID : wejście nr 1 , wyjście nr 2
2. Drugi PID : wejście nr 3 , wyjście nr 1
3. Trzeci PID : wejście nr 4 , wyjście nr 3

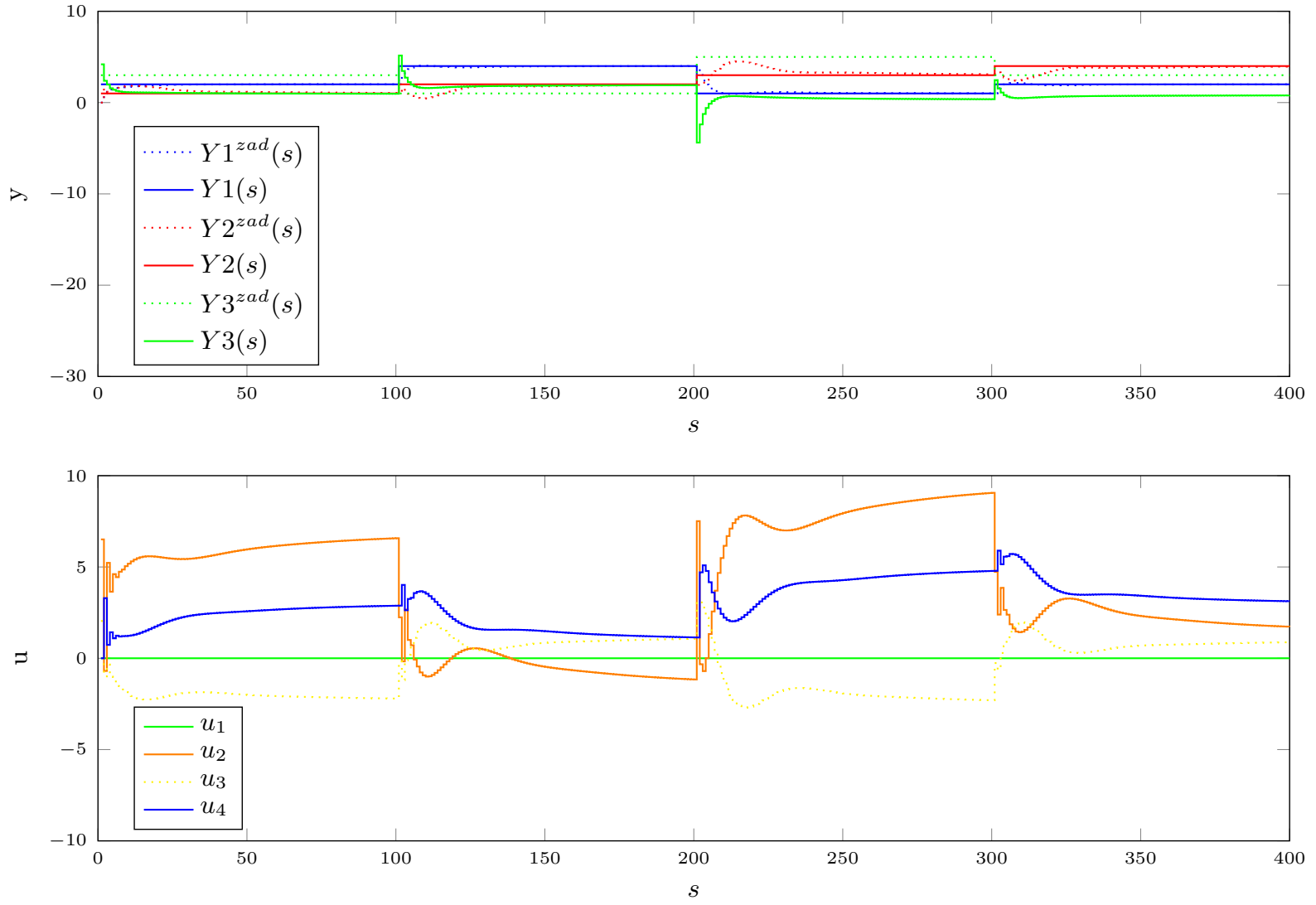
4.2. Kalibracja algorytmu PID w wersji MIMO

Kalibrację przeprowadzamy za pomocą klasycznej metody inżynierskiej. Kalibrację zaczynamy od regulatora pierwszego, gdyż wejście numer 1 ma znaczący wpływ na wyjście numer 2 i niewielki wpływ na pozostałe, co pozwala nam na skalibrowanie tego regulatora niezależnie od pozostałych (oczywiście ten wpływ jest, ale na tyle mały, że możemy go pominąć). Następnie kalibrujemy pozostałe dwa PID na raz, gdyż ich wpływ jest znaczący na więcej niż jedno wyjście. Po przeprowadzonej kalibracji wykorzystujemy algorytm ewolucjny zaimplementowany w

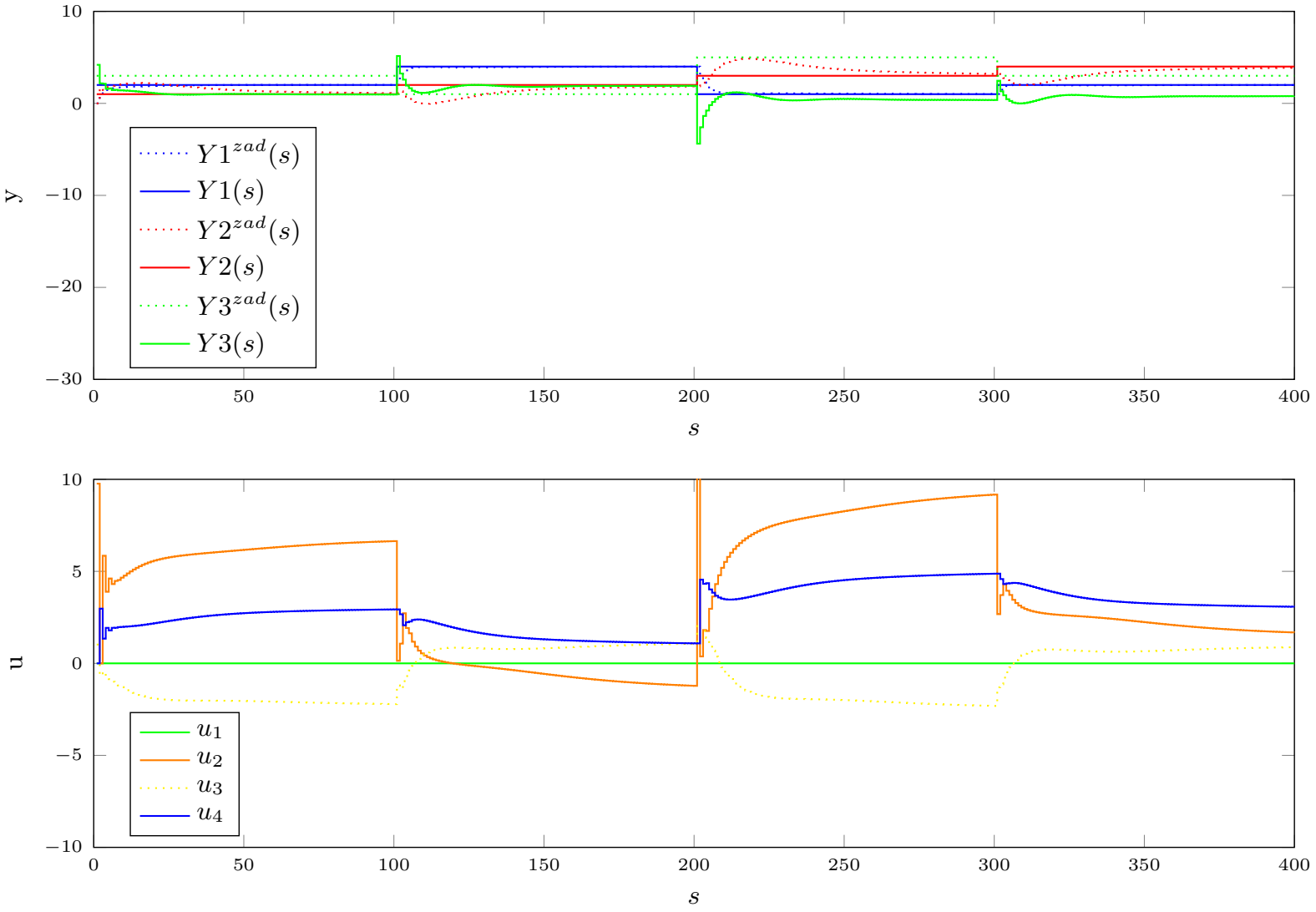
środowisku matlab do sprawdzenia, czy otrzymana jakość regulacji metodą inżynierską może być znacząco lepsza (oczywiście algorytm ewolucyjny nie daje gwarancji na znalezienie optimum globalnego, ale jest wstanie zazwyczaj znaleźć zadowalające optimum lokalne).



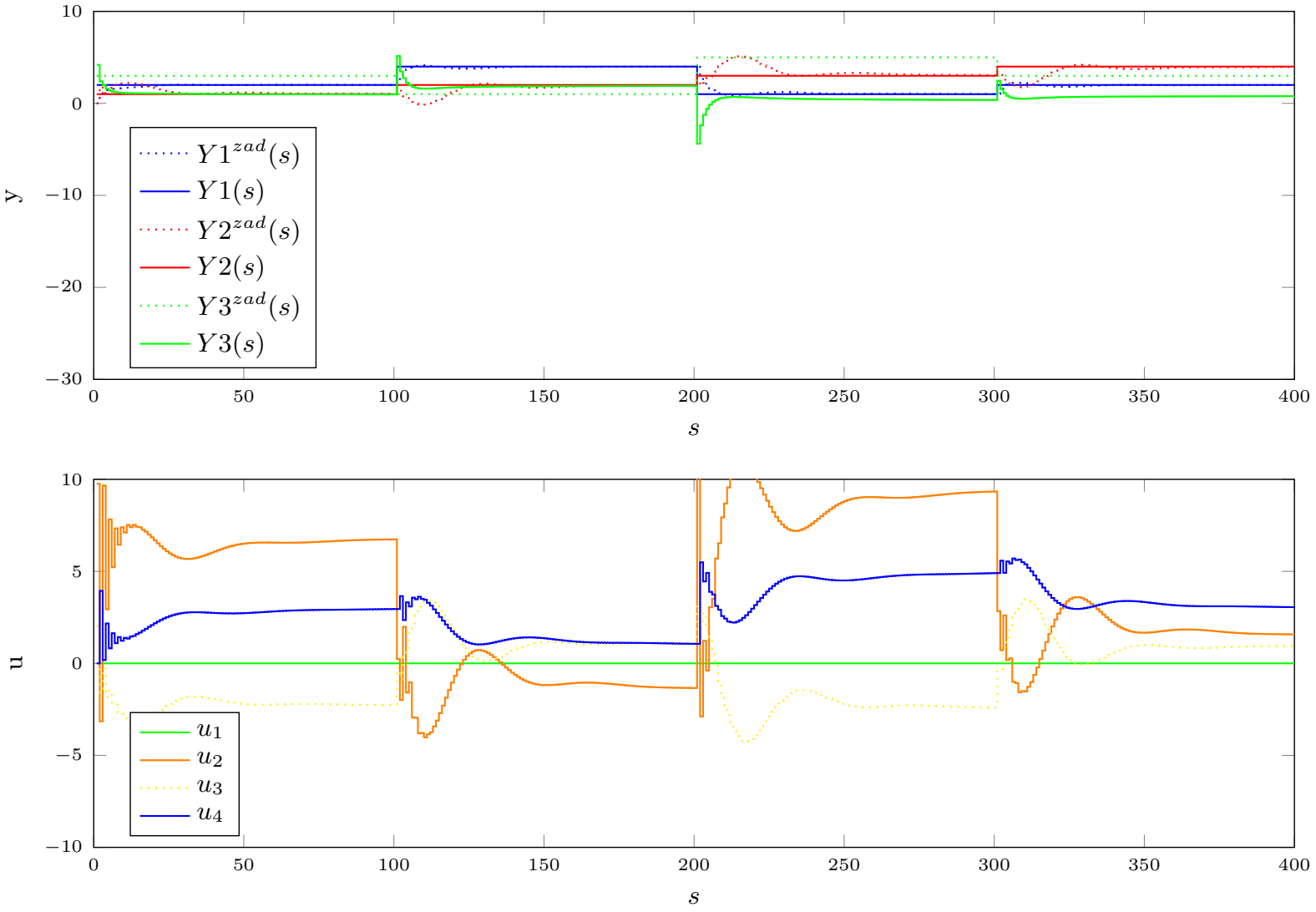
Rys. 4.13. PID1 K=1, Ti=10000, Td=0, PID2 K=0, Ti=10000, Td=0, PID3 K=0, Ti=10000, Td=0, error=6965



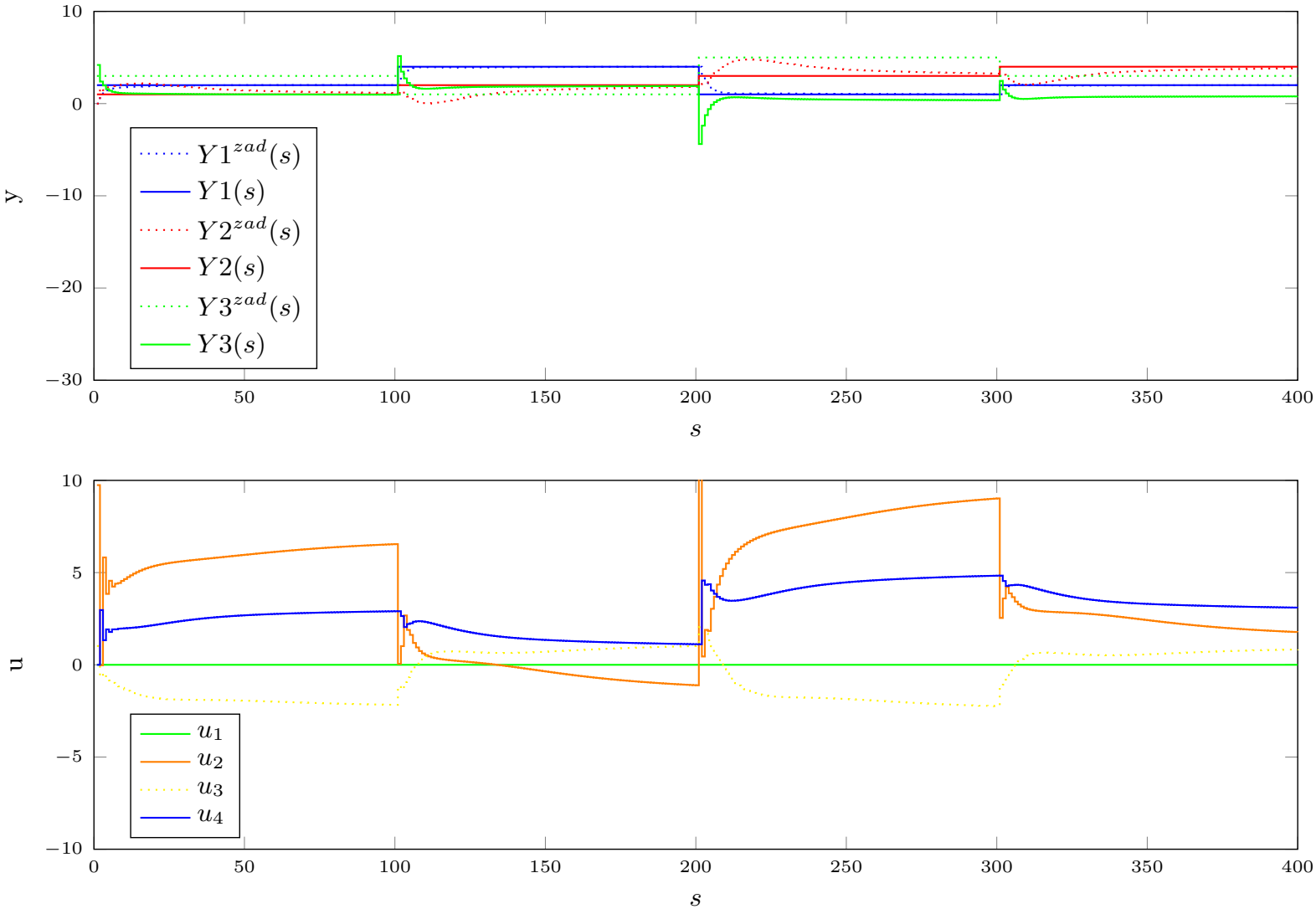
Rys. 4.14. PID1 K=2, Ti=9, Td=0.01, PID2 K=2, Ti=10, Td=0.03, PID3 K=2, Ti=12, Td=0, error=582



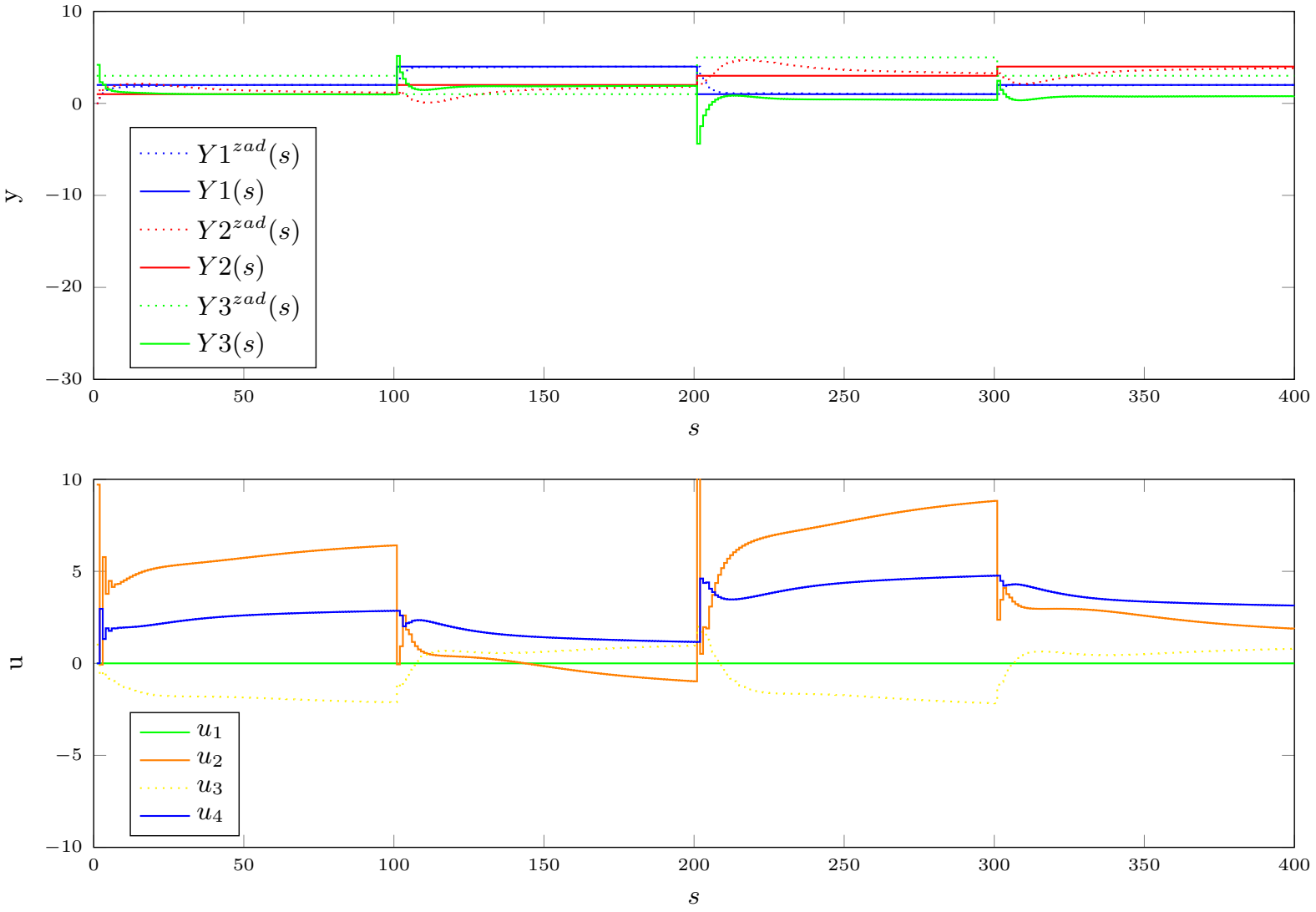
Rys. 4.15. PID1 K=2, Ti=9, Td=0.01, PID2 K=3, Ti=10, Td=0.03, PID3 K=1, Ti=12, Td=0, error=529



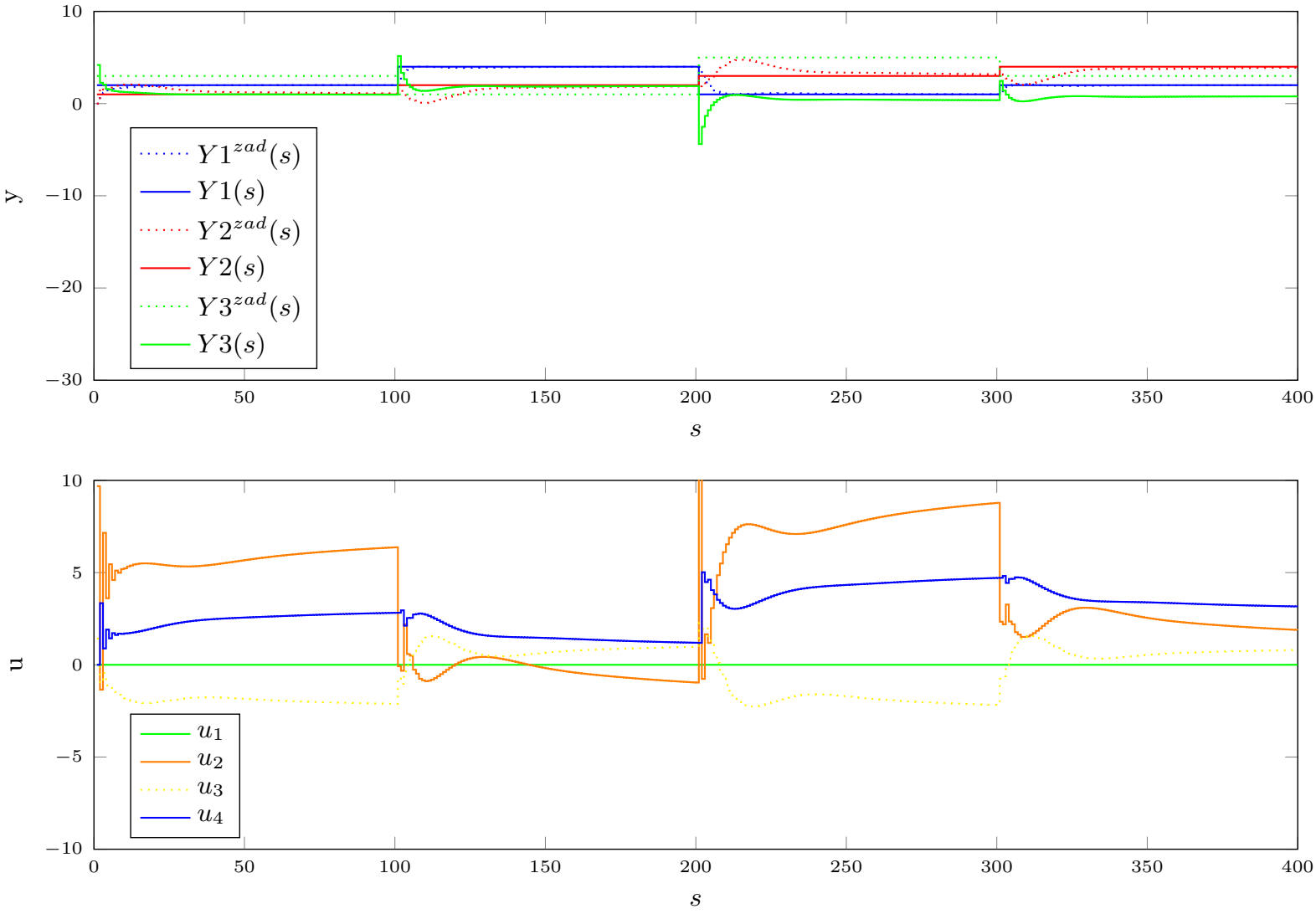
Rys. 4.16. PID1 K=2, Ti=9, Td=0.01, PID2 K=3, Ti=10, Td=0.03, PID3 K=2, Ti=12, Td=0, error=557



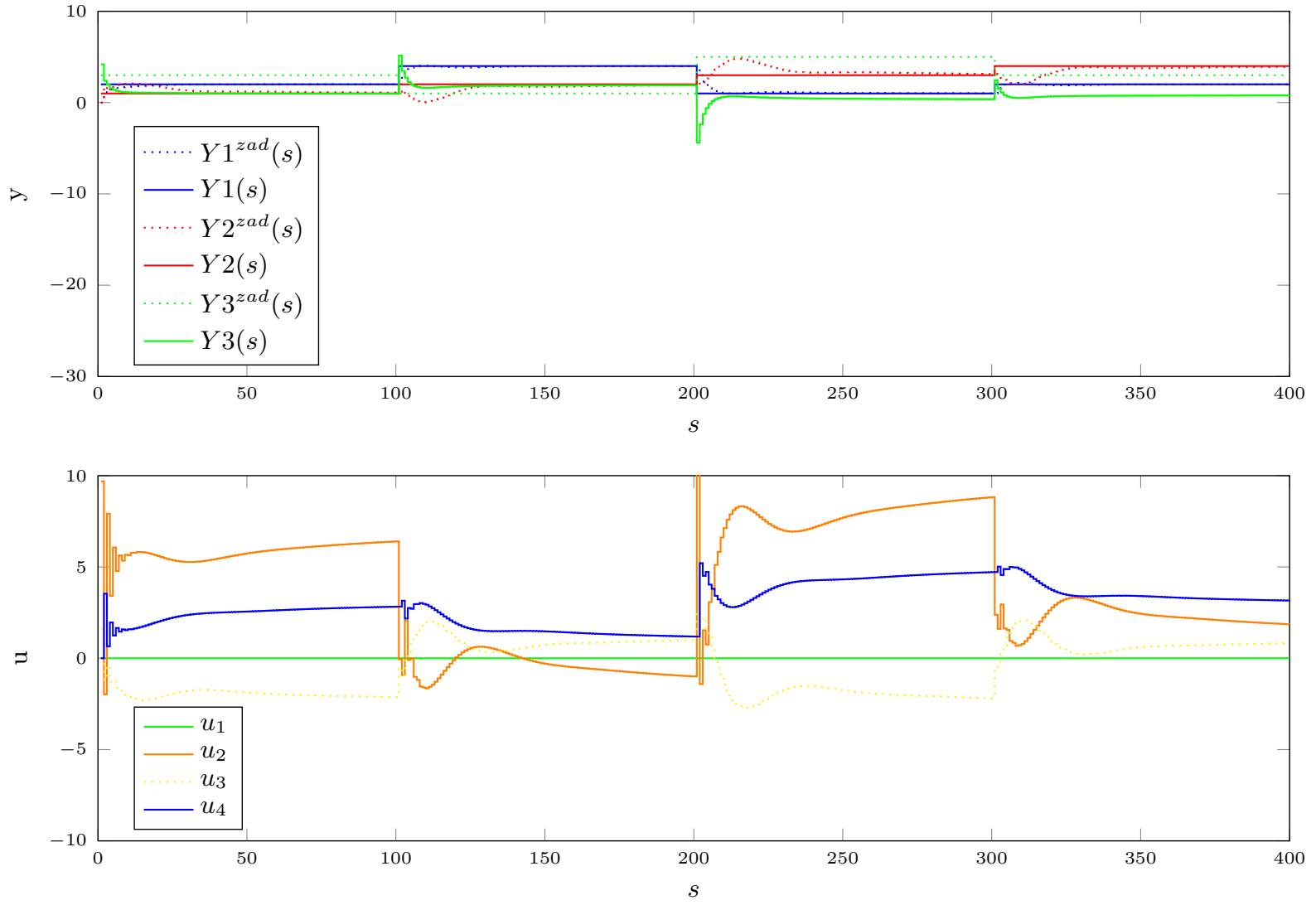
Rys. 4.17. PID1 K=2, Ti=9, Td=0.01, PID2 K=3, Ti=11, Td=0.03, PID3 K=1, Ti=13, Td=0, error=521



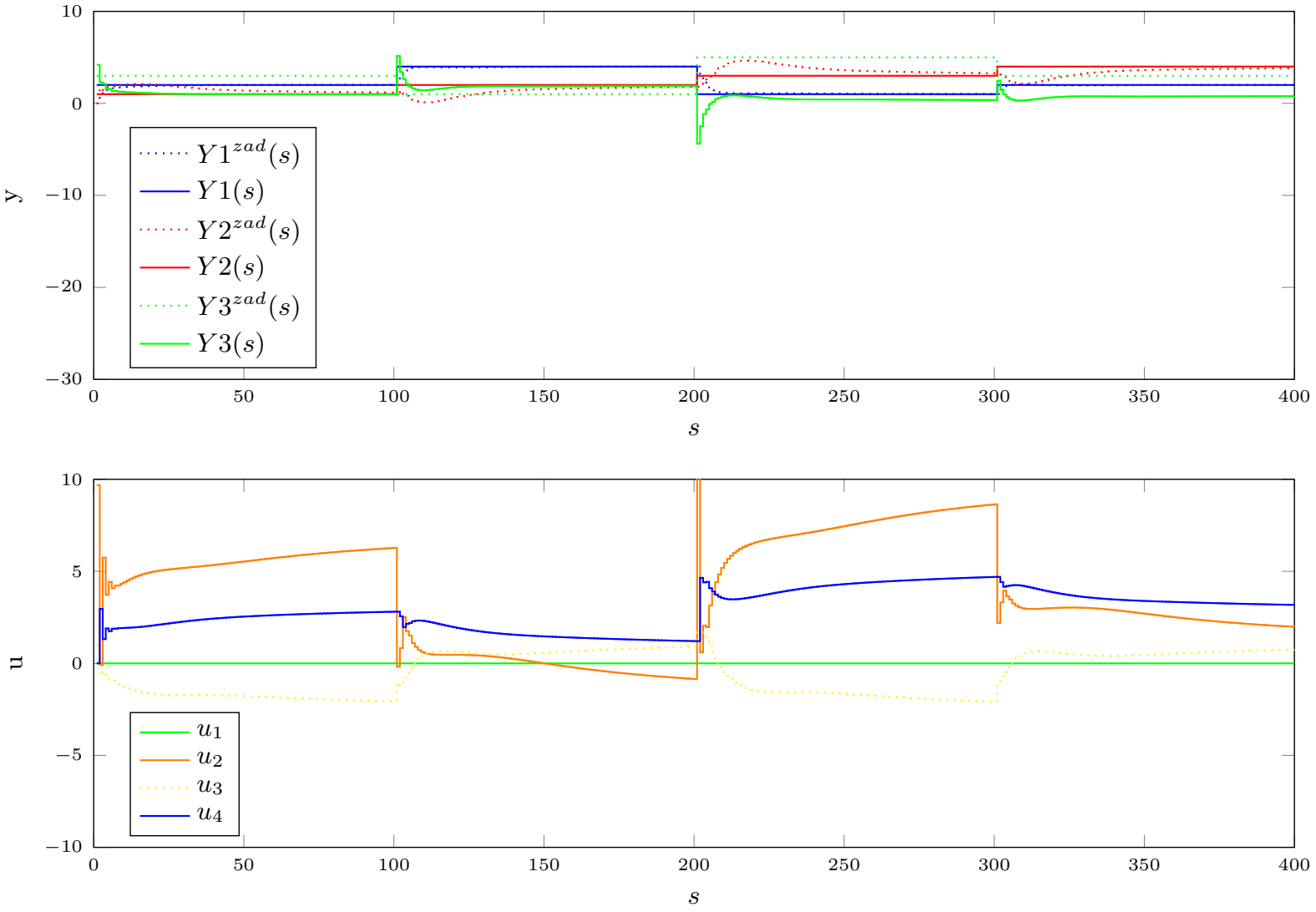
Rys. 4.18. PID1 K=2, Ti=9, Td=0.01, PID2 K=3, Ti=13, Td=0.03, PID3 K=1, Ti=13, Td=0, error=511



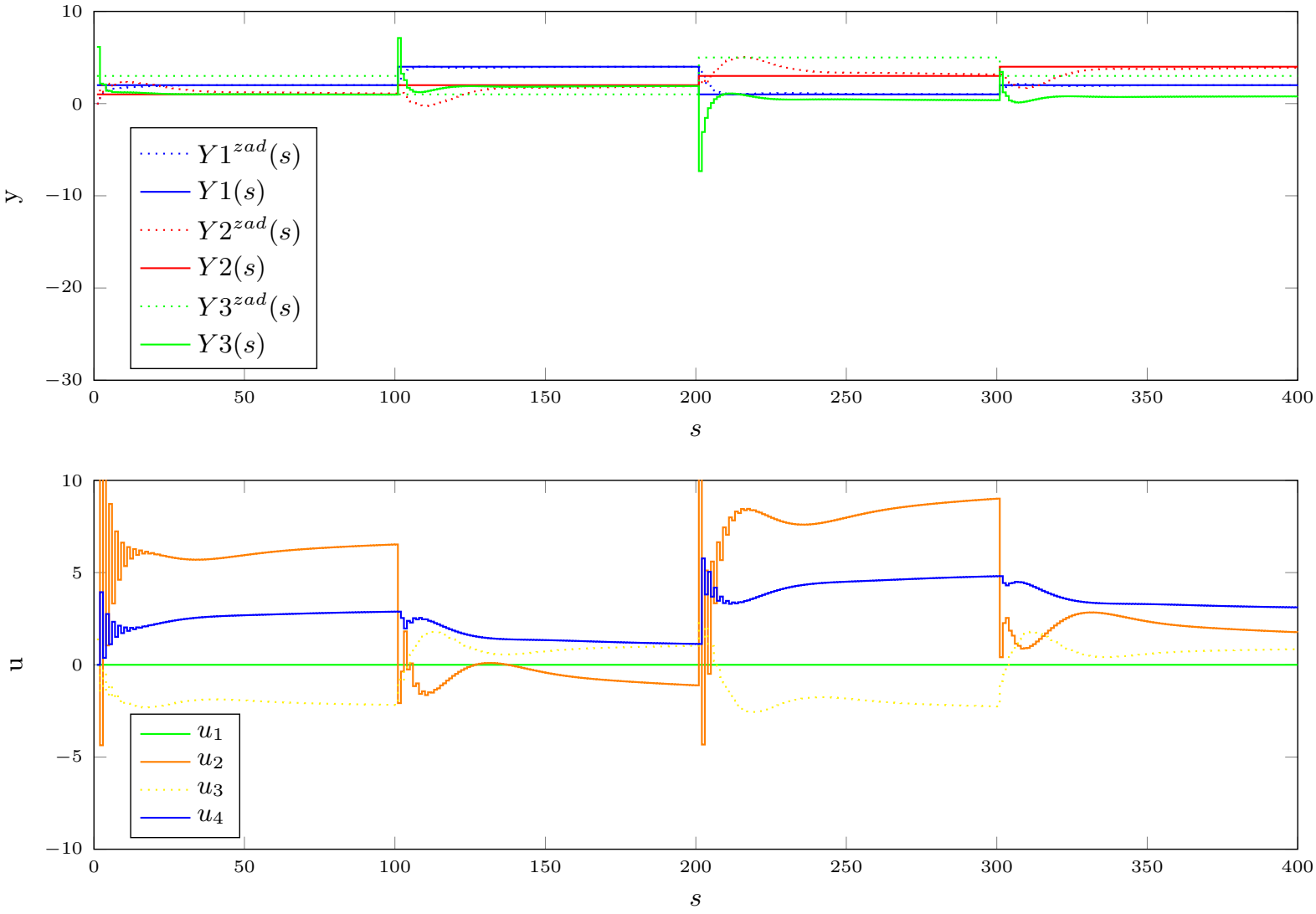
Rys. 4.19. PID1 K=2, Ti=9, Td=0.01, PID2 K=3, Ti=15, Td=0.03, PID3 K=1.4, Ti=13, Td=0, error=491



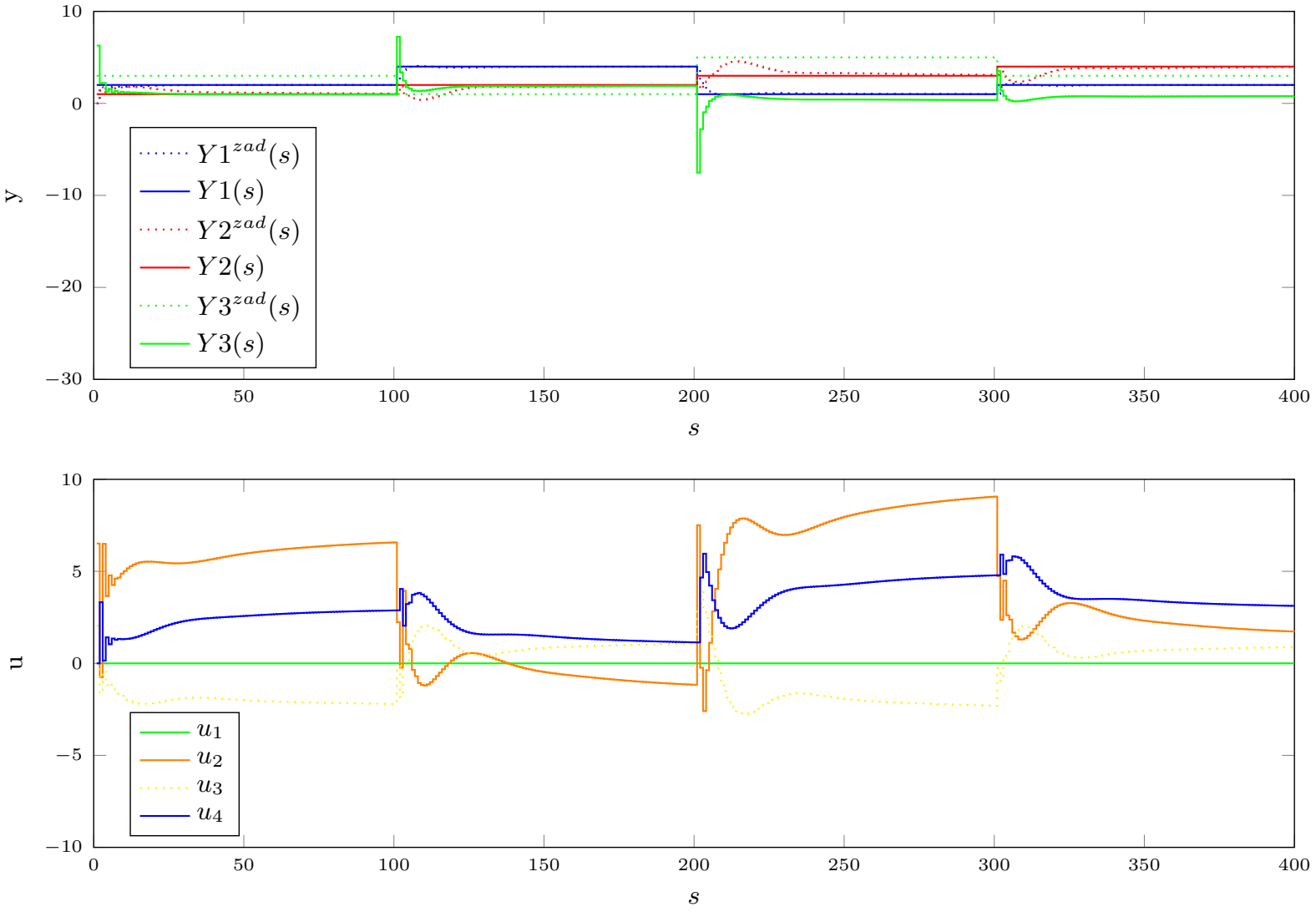
Rys. 4.20. PID1 K=2, Ti=9, Td=0.01, PID2 K=3, Ti=15, Td=0.03, PID3 K=1.6, Ti=13, Td=0, error=499



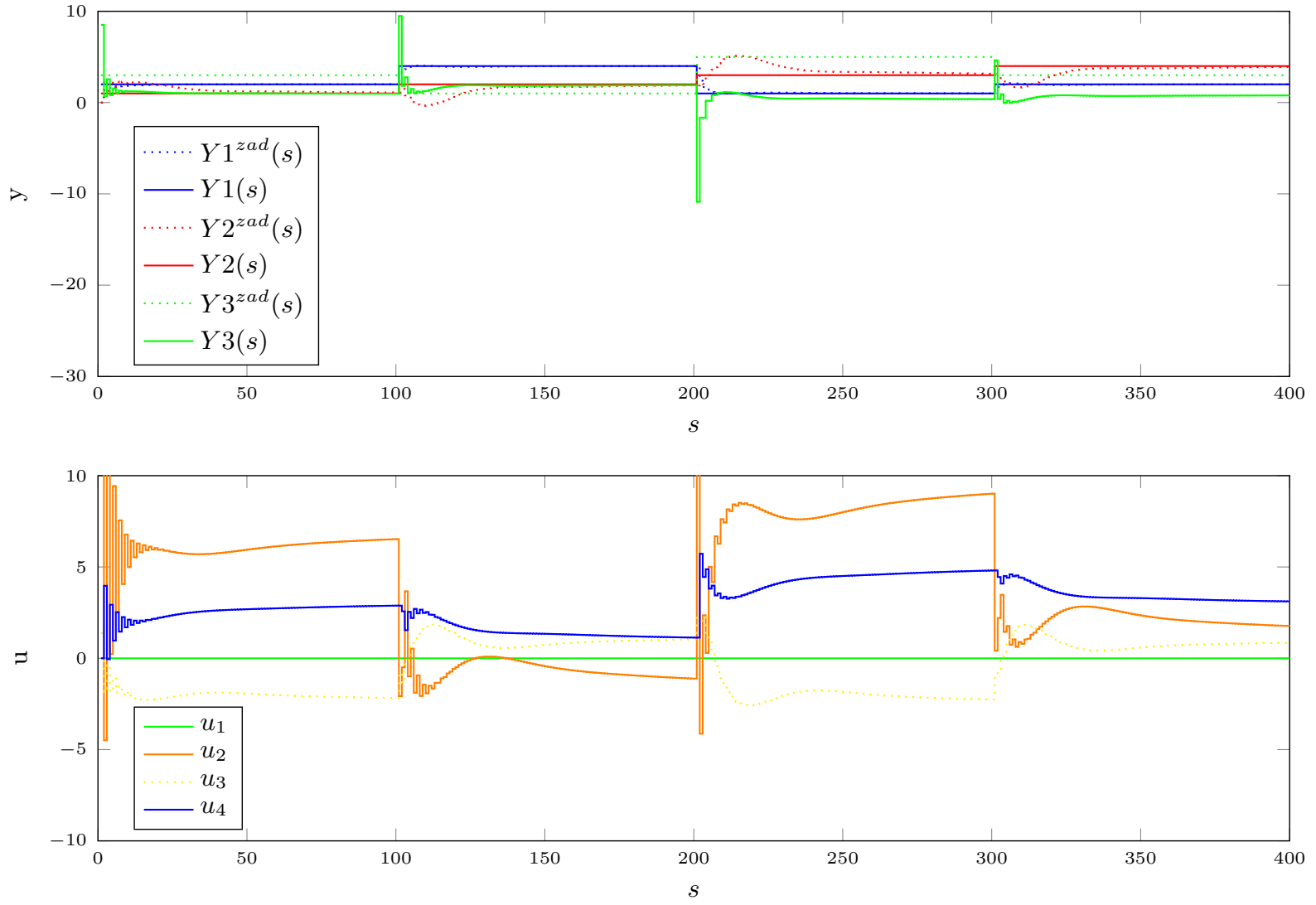
Rys. 4.21. PID1 K=2, Ti=9, Td=0.01, PID2 K=3, Ti=15, Td=0.03, PID3 K=1, Ti=13, Td=0, error=503



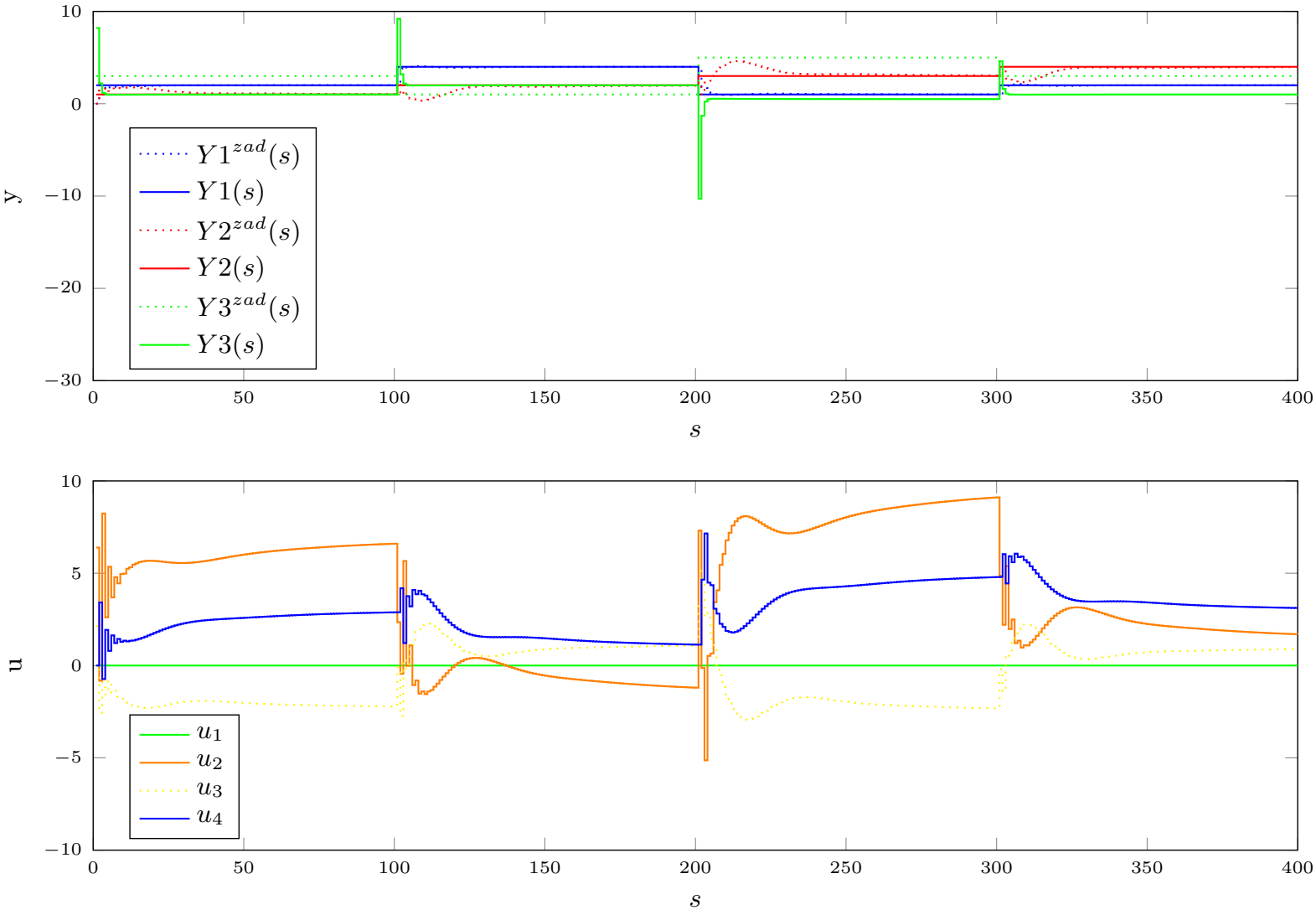
Rys. 4.22. PID1 K=2, Ti=9, Td=0.01, PID2 K=4, Ti=15, Td=0.03, PID3 K=1.3, Ti=13, Td=0.02, error=488



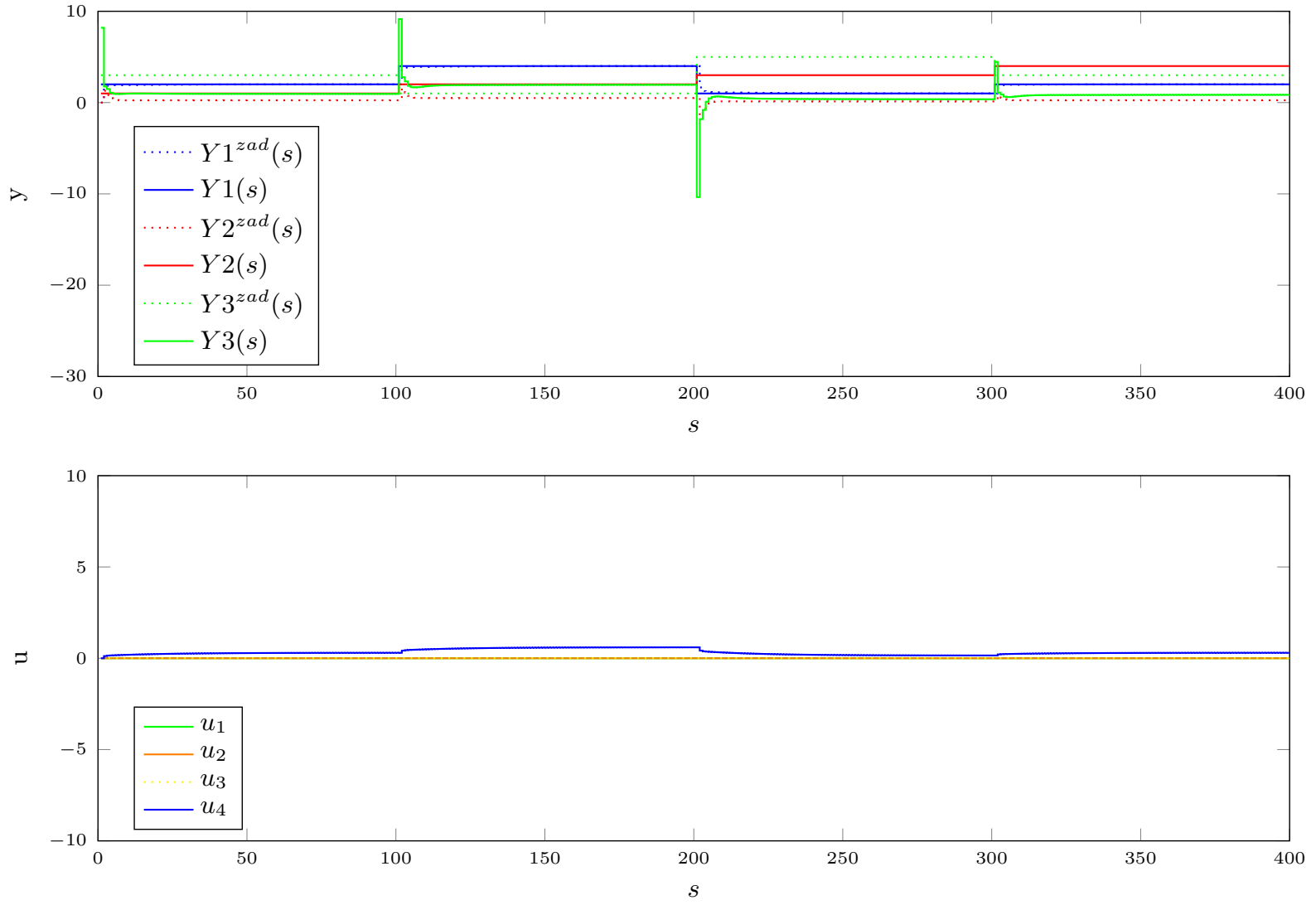
Rys. 4.23. PID1 K=3, Ti=10, Td=0, PID2 K=2, Ti=10, Td=0.03, PID3 K=2, Ti=12, Td=0, error=590



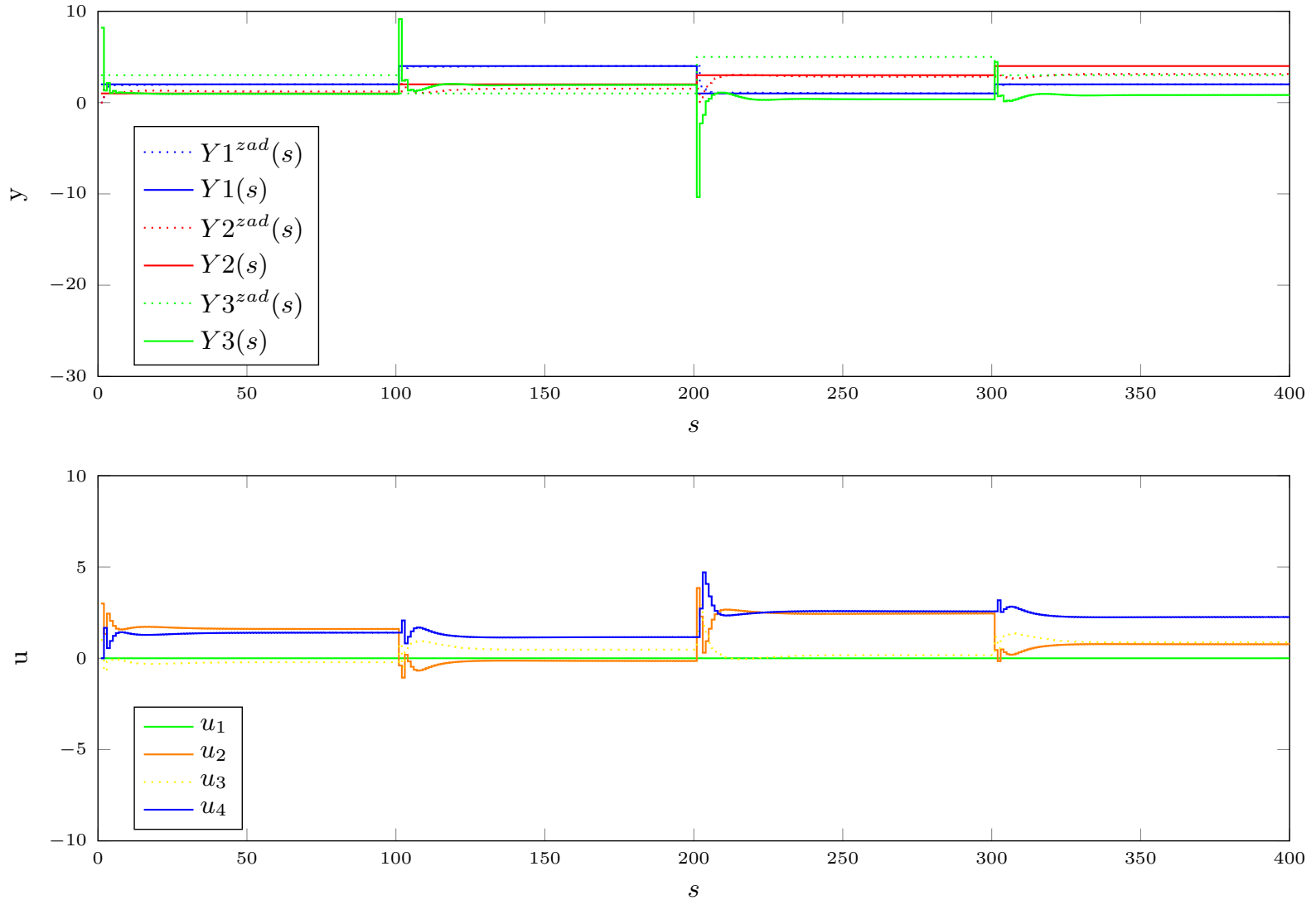
Rys. 4.24. PID1 K=3, Ti=9, Td=0.01, PID2 K=4, Ti=15, Td=0.03, PID3 K=1.3, Ti=13, Td=0.02, error=490



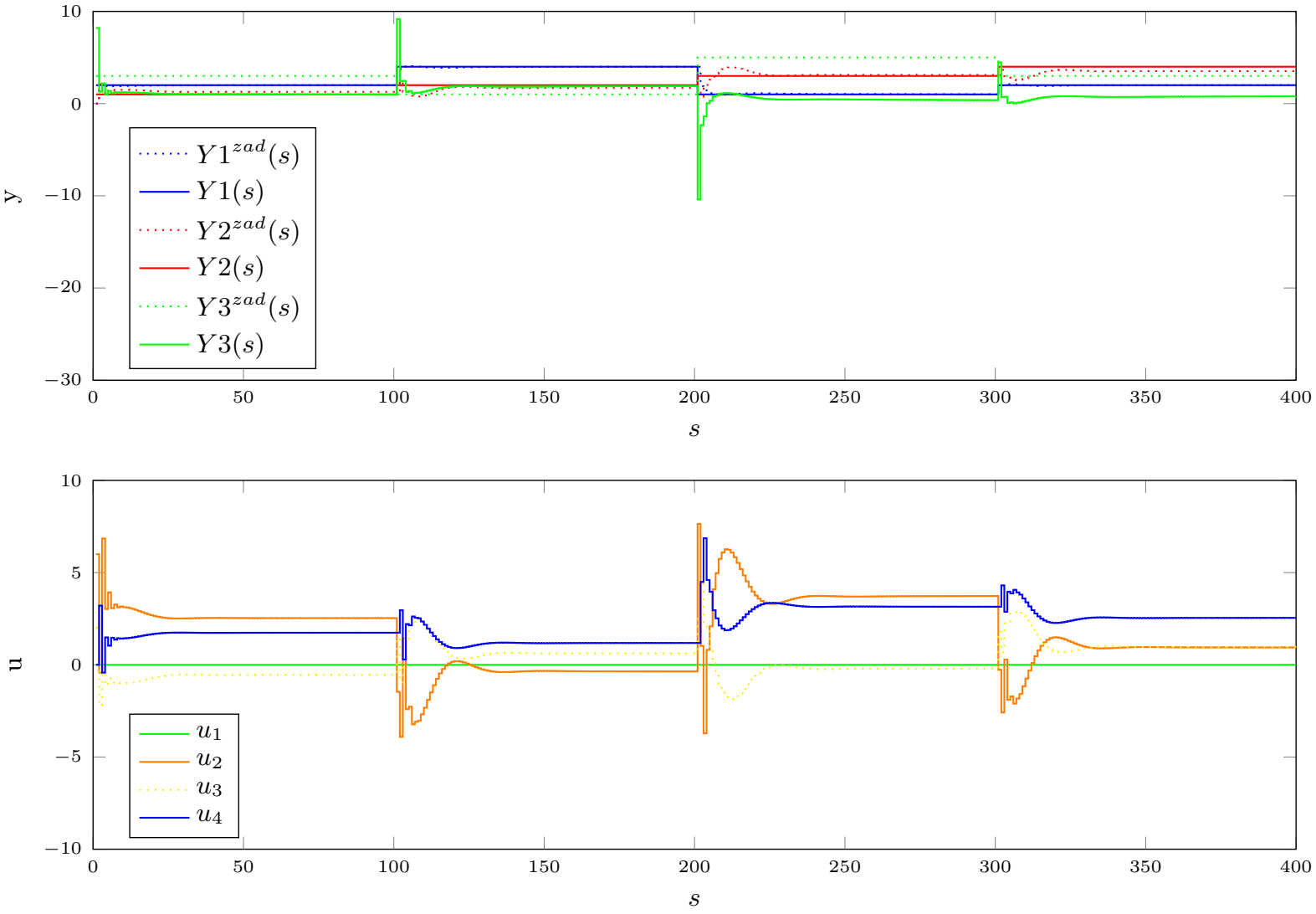
Rys. 4.25. PID1 K=4, Ti=10, Td=0.02, PID2 K=2, Ti=10, Td=0.02, PID3 K=2, Ti=10, Td=0.02, error=625



Rys. 4.26. PID1 K=4, Ti=10, Td=0, PID2 K=0, Ti=10000, Td=0, PID3 K=0, Ti=10000, Td=0, error=6348

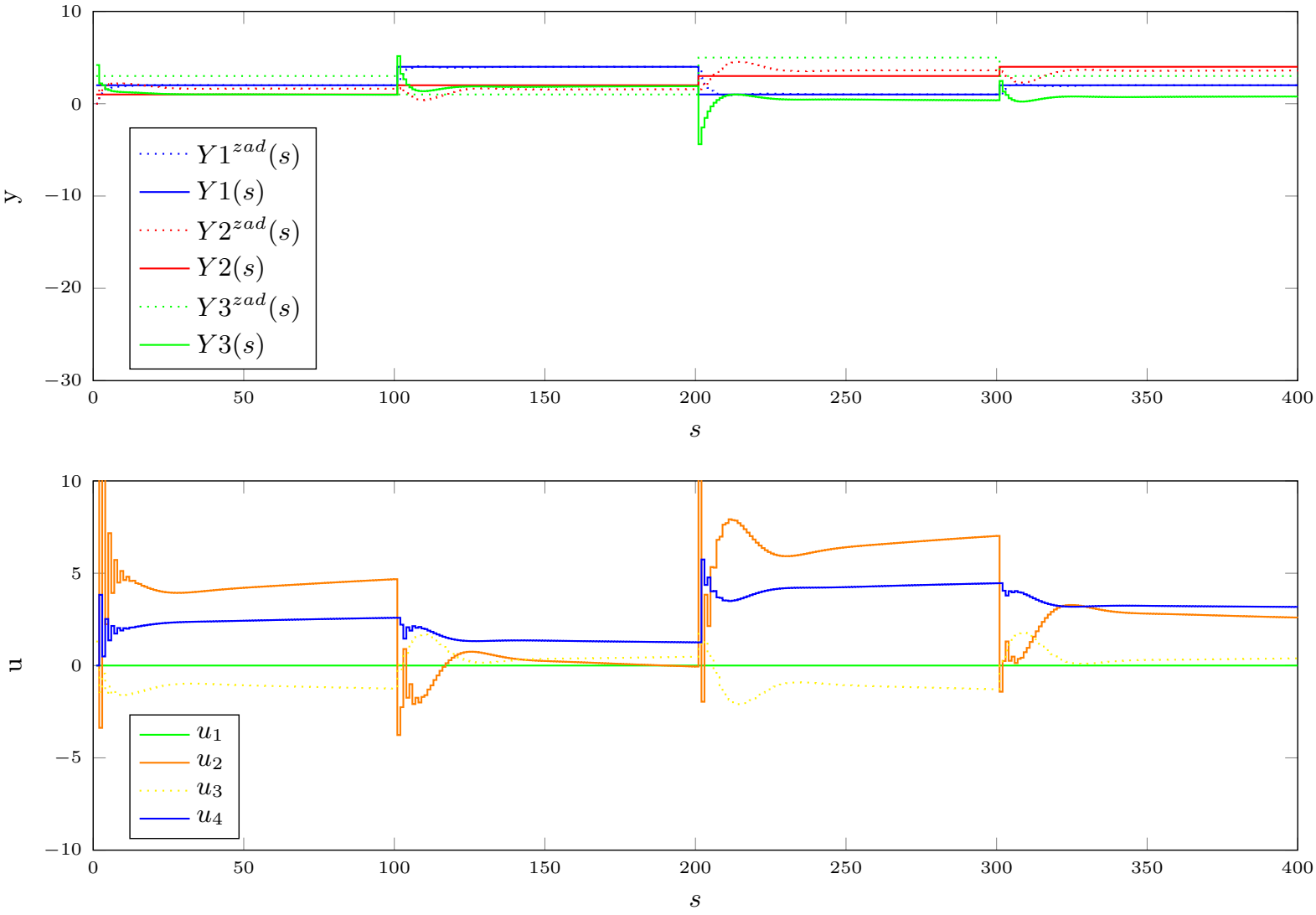


Rys. 4.27. PID1 K=4, Ti=10, Td=0, PID2 K=1, Ti=10000, Td=0, PID3 K=1, Ti=10000, Td=0, error=1088



Rys. 4.28. PID1 K=4, Ti=10, Td=0, PID2 K=2, Ti=10000, Td=0, PID3 K=2, Ti=10000, Td=0, error=732

Teraz dla porównania wynik algorytmu ewolucyjnego :



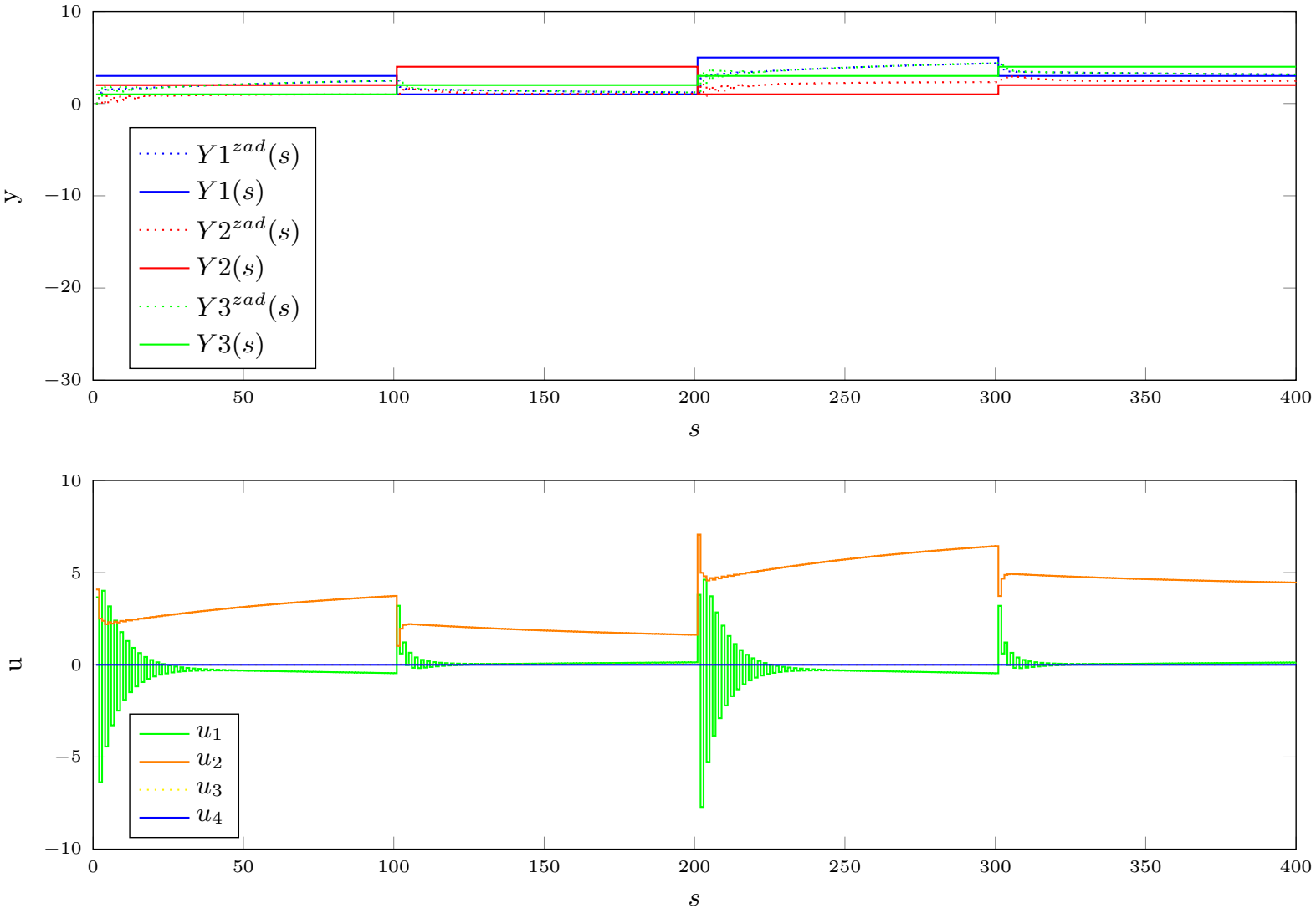
Rys. 4.29. PID1 K=2.1953, Ti=6.0433, Td=0.042969, PID2 K=4.2026, Ti=45.3851, Td=0.00012793, PID3 K=1.2957, Ti=99.9993, Td=0.00092811, error=405

Jak widać algorytm ewolucyjny znalazł lepsze rozwiązanie, ale otrzymana jakość regulacji metodą inżynierską jest porównywalna, ale nadal daleka od zadowalającej.

Postanowiliśmy też sprawdzić jakość regulacji dla innej konfiguracji par wejścia-wyjście regulatorów. W tej części ograniczymy się jedynie do wyników algorytmu ewolucyjnego, gdyż pokazywanie kolejnych wykresów z kalibracji metodą inżynierską mija się z celem.

Poniższy wynik jest dla przypadku :

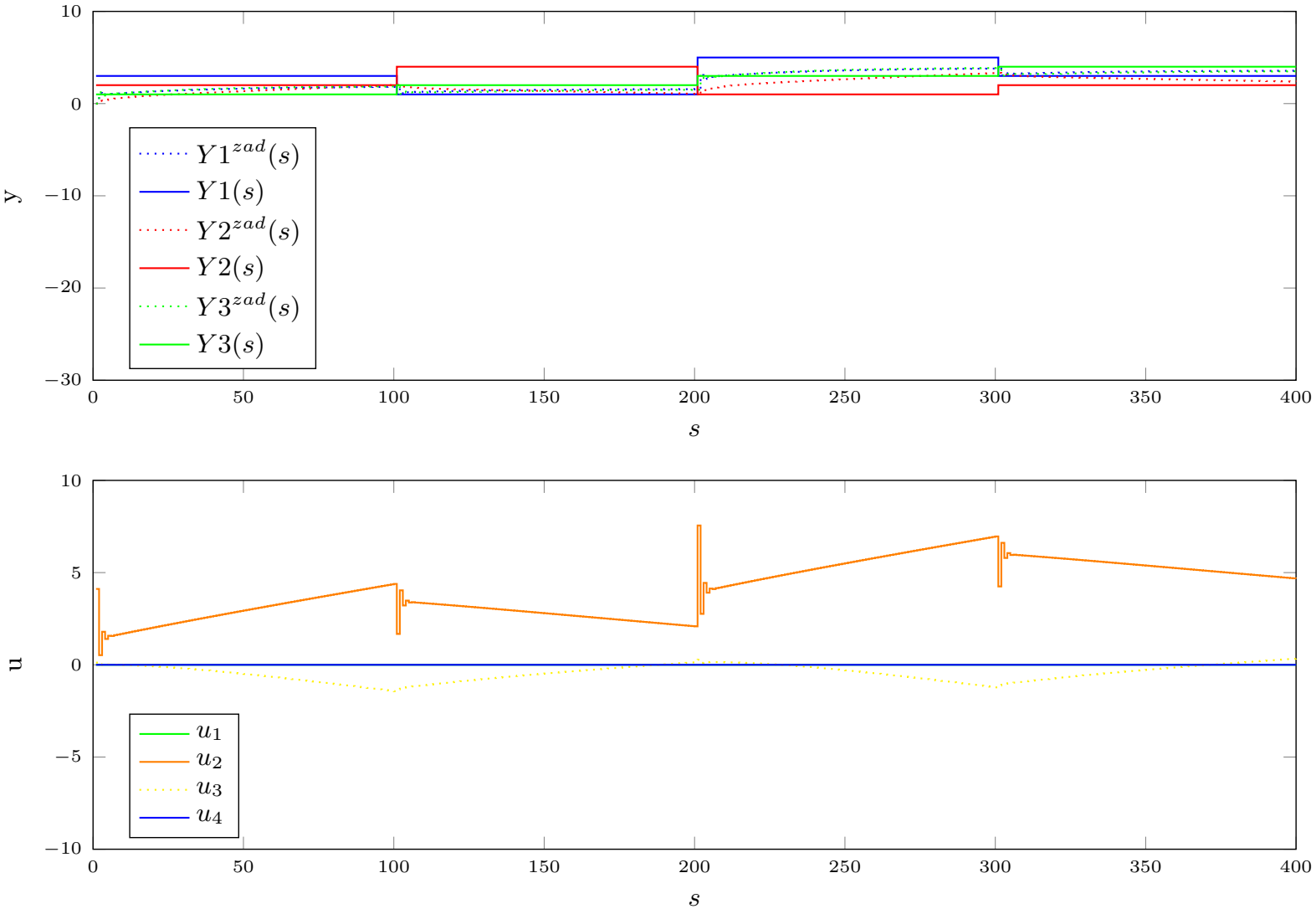
1. Pierwszy PID : wejście nr 3 , wyjście nr 2
2. Drugi PID : wejście nr 2 , wyjście nr 1
3. Trzeci PID : wejście nr 1 , wyjście nr 3



Rys. 4.30. PID1 K=0.004756, Ti=71.7394, Td=0.0035729, PID2 K=1.3234, Ti=20.7097, Td=0.008631, PID3 K=0.21697, Ti=99.9936, Td=7.9386, error=1715

Poniższy wynik jest dla przypadku :

1. Pierwszy PID : wejście nr 4 , wyjście nr 2
2. Drugi PID : wejście nr 2 , wyjście nr 1
3. Trzeci PID : wejście nr 3 , wyjście nr 3



Rys. 4.31. PID1 K=0, Ti=9.0362, Td=9.0012, PID2 K=0.66584, Ti=13.4519, Td=0.52067, PID3 K=0.10938, Ti=2.3822, Td=0, error=1758

Jak można łatwo zauważyć, że zły dobór par wejście-wyjście dla regulatorów powoduje, znacząco gorszą jakość regulacji, jeżeli w tych dwóch wypadkach w ogóle można mówić o regulacji (oczywiście jest to regulacja, ale jej jakość jest na tyle niska, że nie można mówić tu jakkolwiek zadowalających efektach).

5. DMC

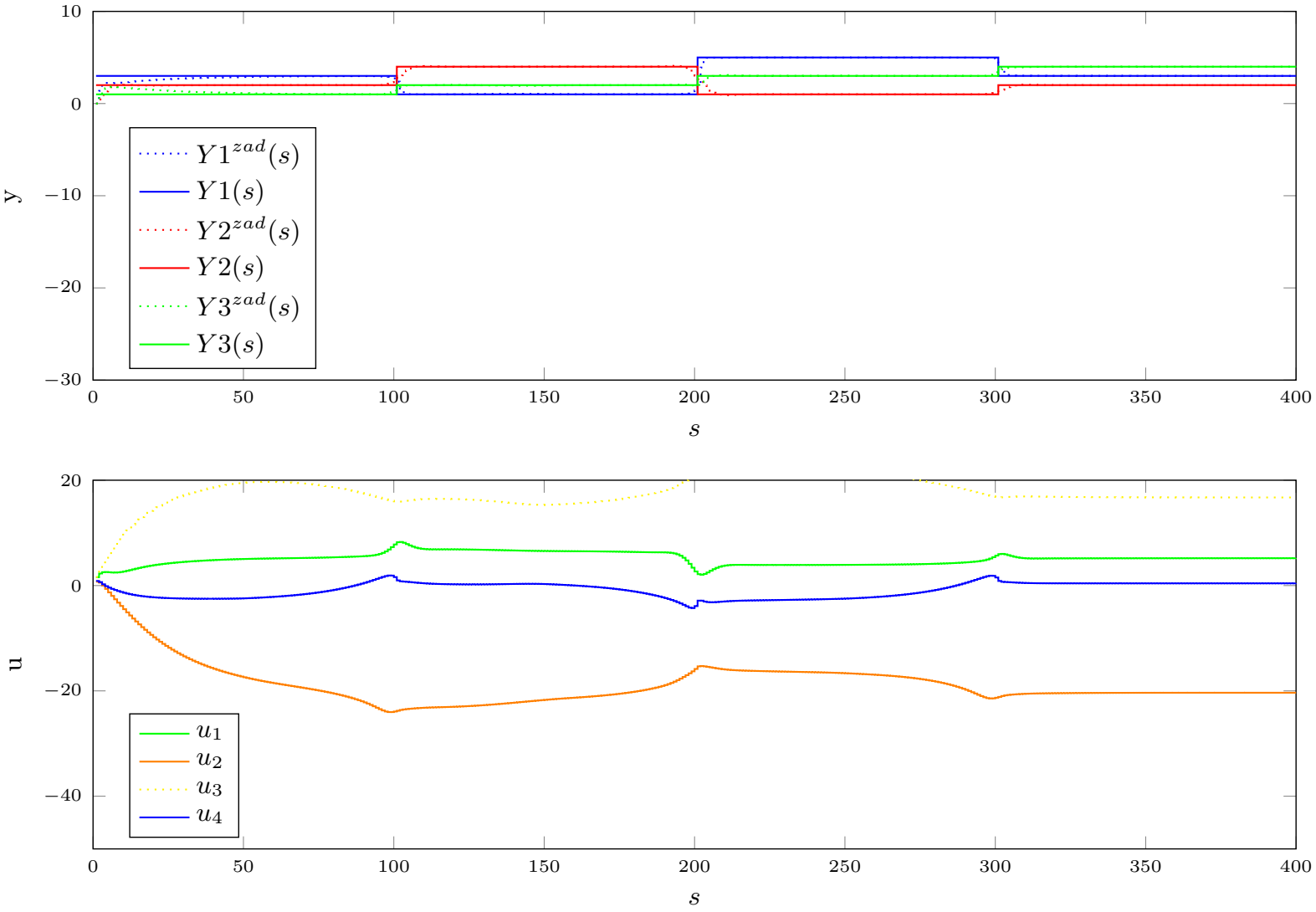
5.1. Algorytm DMC w wersji MIMO

Algorytm DMC w wersji MIMO działa na tej samej zasadzie co algorytm DMC w wersji SISO, czyli oblicza zachowanie wyjścia obiektu wprzód na podstawie poprzednich przyrostów sygnału sterowania. Podstawowa różnicą oczywiście jest to, że w naszym przypadku jest 12 torów sterowania (4 wejścia i 3 wyjścia), ale sama reguła działania pozostaje taka sama. Już można tutaj zauważyć przewagę DMC w wersji MIMO nad PID w wersji MIMO taką, że DMC wykorzystuje wszystkie wejścia i "jest świadome" wpływu każdego wejścia na każde wyjście. Pozwala to nam na otrzymanie znacząco lepszej jakości regulacji co potwierdzają nasze eksperymenty. Oczywiście nie ma nic za darmo co oznacza, że DMC w wersji MIMO wymaga znacząco ilość obliczeń niż PID w wersji MIMO, ale ten problem możemy zmniejszyć poprzez zaimplementowanie DMC w najprostszej wersji analitycznej (dzięki temu mamy mniej obliczeń do wykonania co obrót pętli).

5.2. Kalibracja algorytmu DMC w wersji MIMO

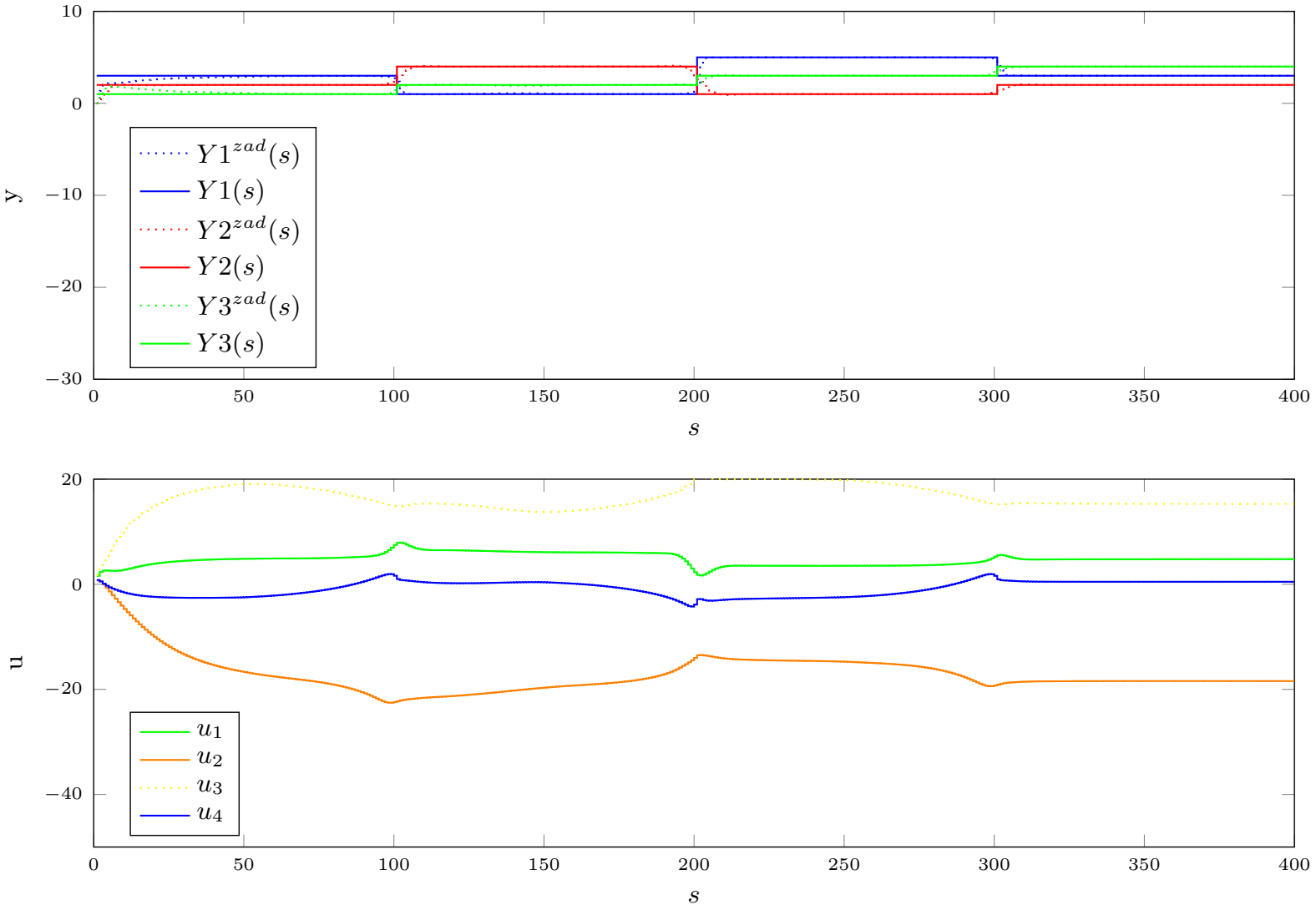
Kalibrację podzieliliśmy na dwie części. Pierwsza część składa się z doboru parametrów podstawowych takich jak horyzont dynamiki obiektu, horyzont predykcji i horyzont sterowania algorytmu DMC. Parametry Lambda i Psi są równe 1, żeby nie faworyzować żadnego toru sterowania. Druga część składa się z doboru parametrów Lambda i Psi dla ustalonych wcześniej wartości podstawowych. Dla otrzymania możliwie najlepszej jakości regulacji będziemy wykorzystawali wcześniej wspomniany algorytm ewolucyjny, by można porównać wyniki z wynikami regulacji za pomocą algorytmu PID w wersji MIMO.

Wyniki pierwszego etapu kalibracji możemy zobaczyć na wykresach poniżej:

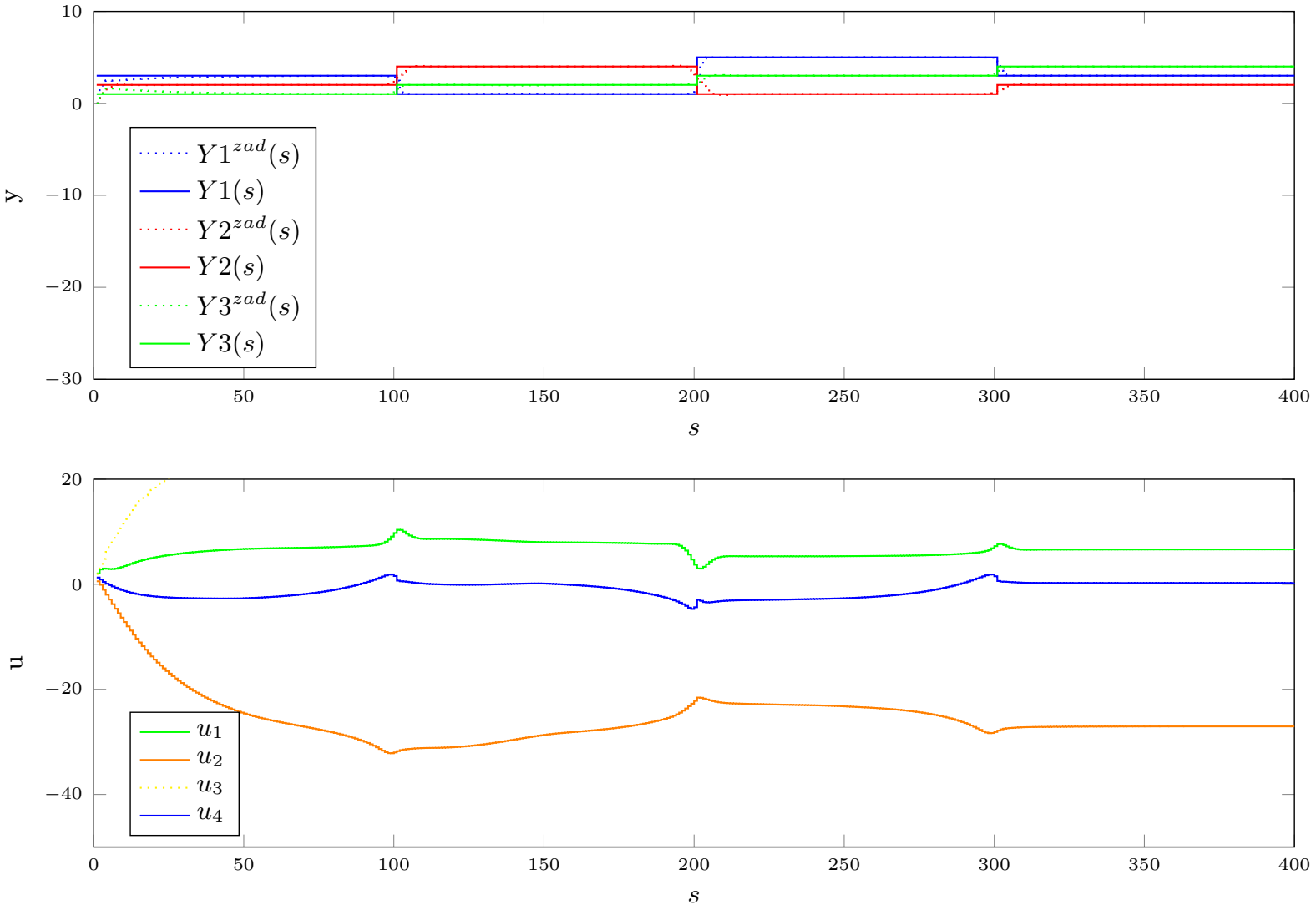


Rys. 5.1. DMC=D=204, N=107, Nu=54, Lambda=[1, 1, 1, 1], [Psi=1, 1, 1], error=69.8736

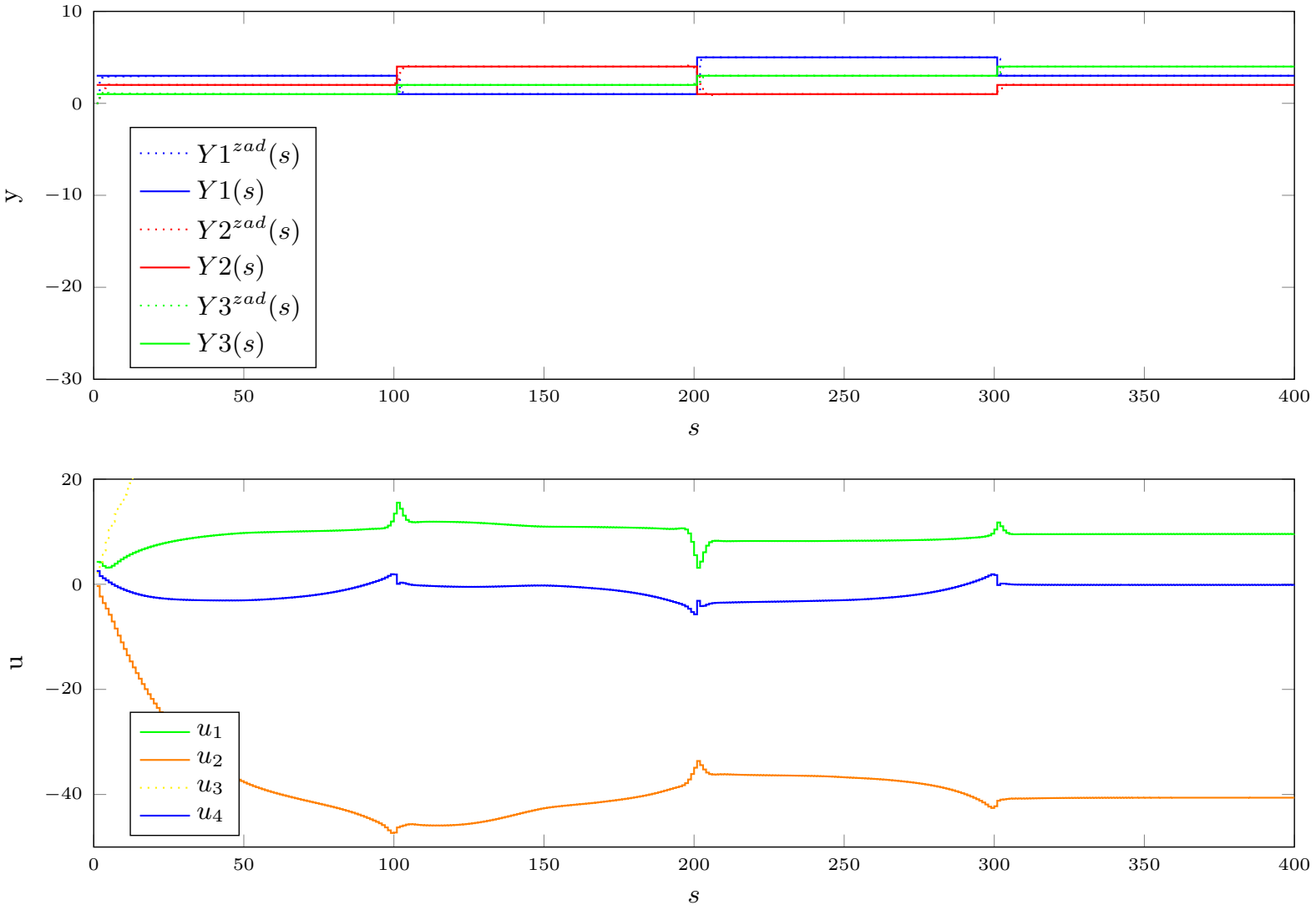
W drugim etapie kalibracji wykorzystamy zarówno metodę inżynierska jak i algorytm ewolucyjny, by móc również porównać, czy metoda inżynierska pozwala na otrzymanie porównywalnego wyniku. Najpierw pragniemy pokazać wyniki kalibracji metodą inżynierską:



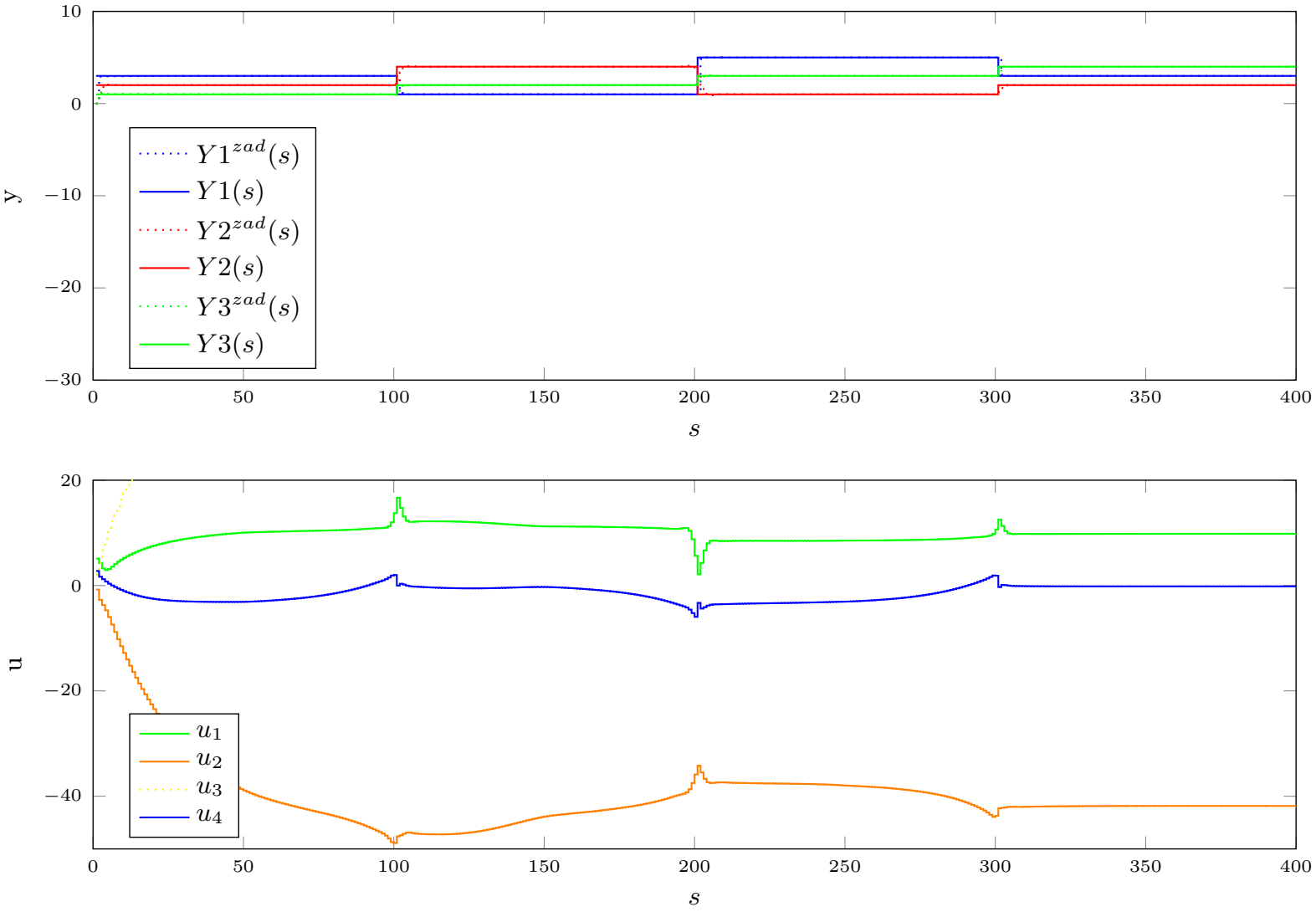
Rys. 5.2. DMC=D=204, N=107, Nu=54, Lambda=[1, 2, 3, 4], Psi=[1, 2, 3], error=69.2682



Rys. 5.3. DMC=D=204, N=107, Nu=54, Lambda=[0.5, 0.5, 0.5, 0.5], Psi=[1, 1, 1], error=55.4869

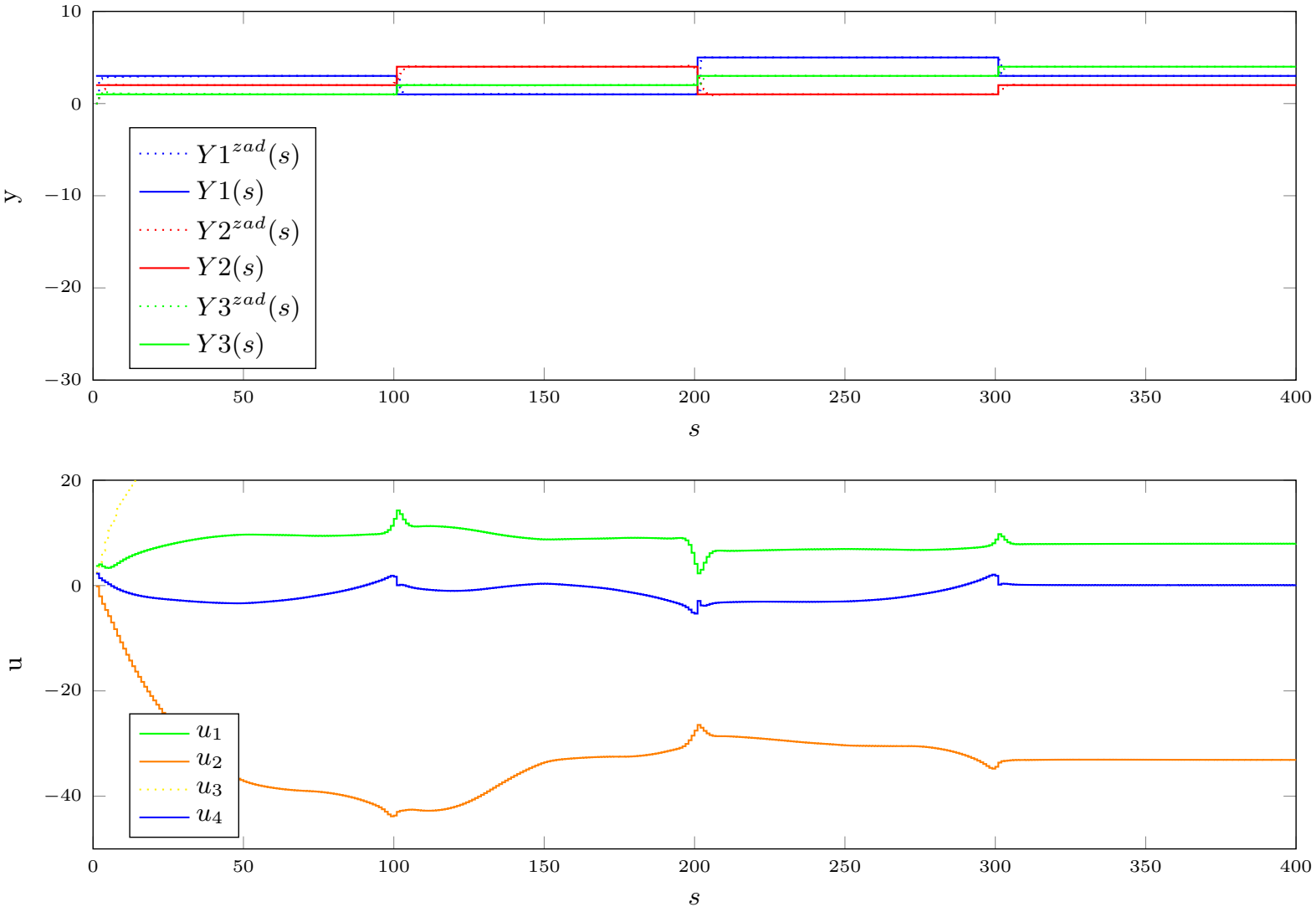


Rys. 5.4. DMC=D=204, N=107, Nu=54, Lambda=[0.5, 0.5, 0.5, 0.5], Psi=[10, 1, 1], error=45.7233



Rys. 5.5. DMC=D=204, N=107, Nu=54, Lambda=[0.5, 0.5, 0.5, 0.5], Psi=[20, 1, 1], error=46.6045

Jak widać jakość regulacji jest znacząco lepsza niż w przypadku algorytmu regulacji PID w wersji MIMO. Błąd obliczany dla przykładowego przebiegu jest prawie 10-krotnie mniejszy niż w przypadku PID. Teraz dla porównania wyniki algorytmu ewolucyjnego:



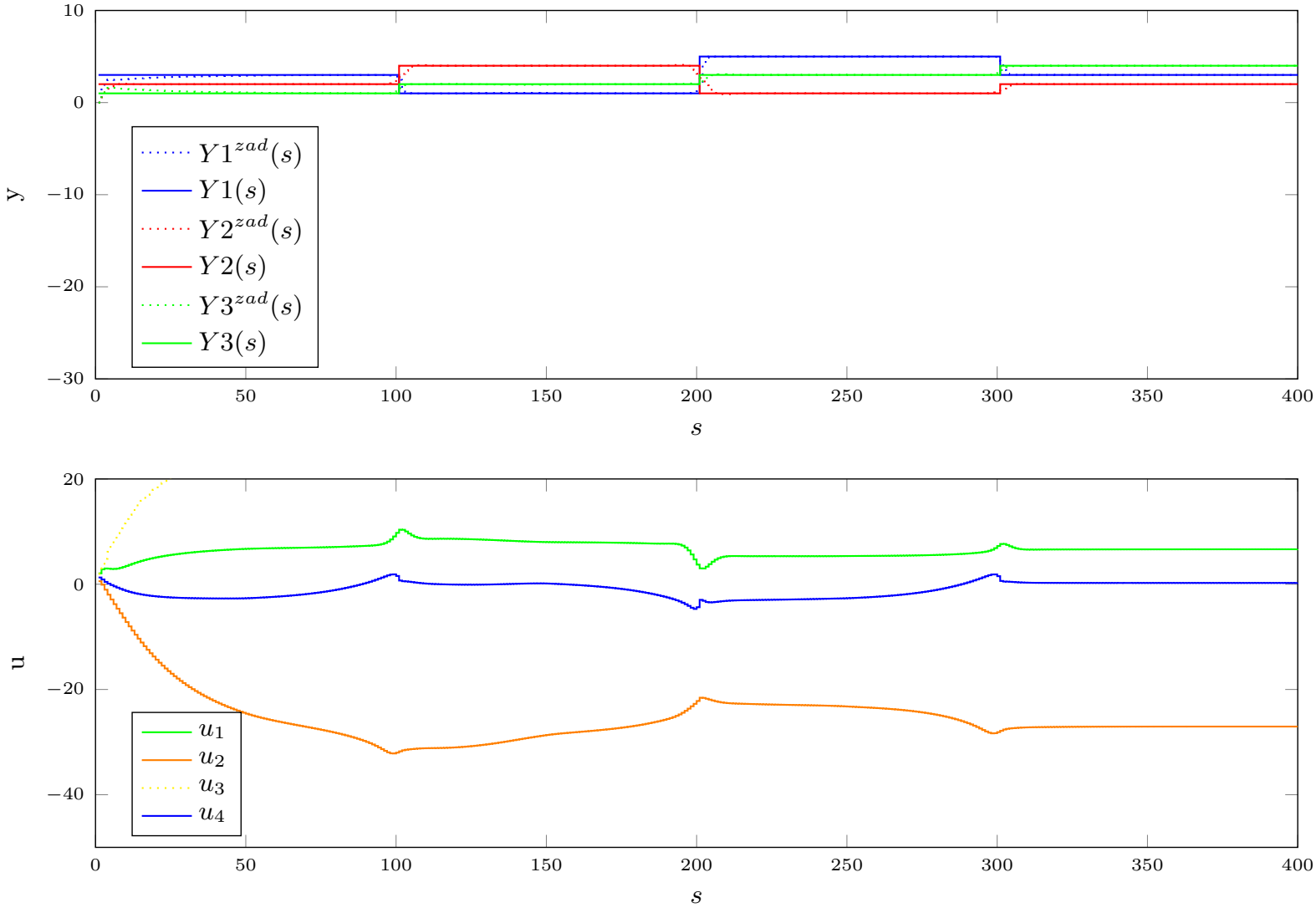
Rys. 5.6. DMC=D=204, N=107, Nu=54, Lambda=[42.7092, 55.6008, 357.2521, 284.4598], Psi=[503.0319, 516.5173, 980.2935], error=45.2539

Obliczony błąd na całej długości przebiegu jest prawie taki sam jak z metody inżynierskiej. Oznacza to, że pomimo 7 parametrów, które na raz kalibrujemy jesteśmy wstanie to zrobić z wykorzystaniem doświadczenia zdobytego podczas kalibrowania innych regulatorów.

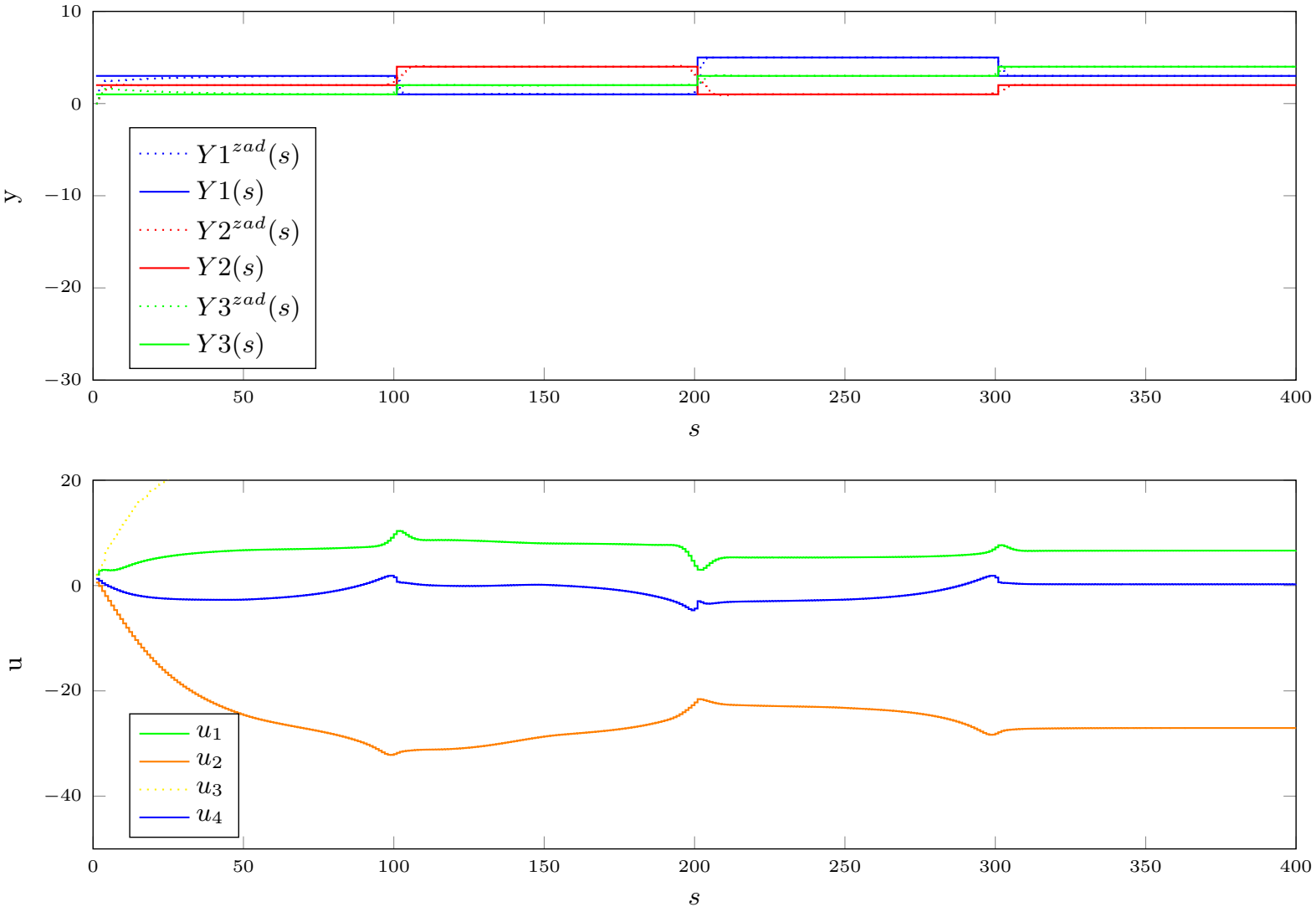
6. Porównanie i wnioski

6.1. Porównianie wyników sterowania algorytmami PID i DMC

Zgodnie z naszymi eksperymentami nie udało nam się znaleźć żadnych różnic pomiędzy dwoma wersjami algorytmu DMC w wersji MIMO jeśli chodzi o jakość regulacji. Zgodnie z logiką i wyczuciem nie powinno być żadnych różnic poza czasem wykonania, gdyż pierwszy element z wektora obliczonych przyszłych przyrostów sterowań jest obliczany w taki sam sposób.



Rys. 6.1. DMC D=204, N=107, Nu=54, Lambda=[0.5, 0.5, 0.5, 0.5], Psi=[1, 1, 1], error=55.4869



Rys. 6.2. DMC D=204, N=107, Nu=54, Lambda=[0.5, 0.5, 0.5, 0.5], Psi=[1, 1, 1], error=55.4869

6.2. Wnioski

Zgodnie z naszymi eksperymentami można wywnioskować , że w przypadku obiektu typu MIMO algorytm DMC pozwoli na znacząco lepszą jakość regulacji niż algorytm PID. Jednakże, jeżeli obiekt typu MIMO można w przybliżeniu podzielić na obiekty typu SISO (każde wejście oddziałuje znacząco na tylko jedno wyjście i ma w przybliżeniu zerowy wpływ na pozostałe wyjścia) to algorytm PID w wersji MIMO również powienien pozwolić na bardzo dobrą jakość regulacji, gdyż algorytm PID bardzo dobrze radzi sobie z regulacją obietów typu SISO.