Sterowanie Dykretne

Patryk Lorenc

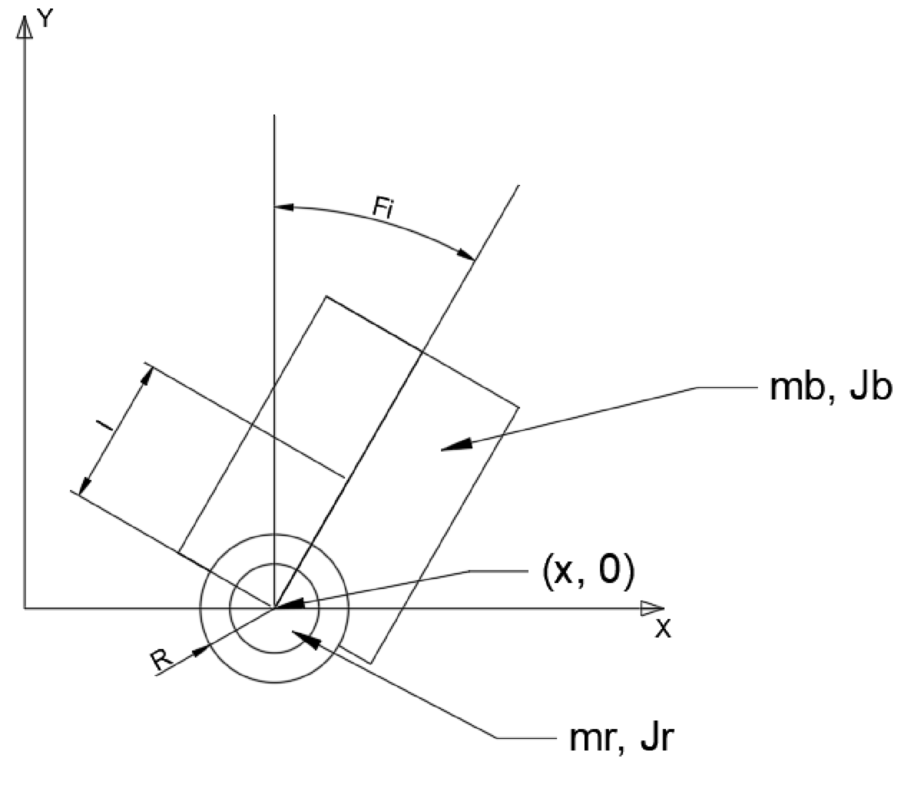
APiR IMiR

Temat: Problem wachadła odwróconego na przykładzie samobalsujacego się dwu kołowego robota

Celem niniejszego projektu jest opracowanie efektywnego układu regulacji dla problemu wahadła odwróconego, zastosowanego w kontekście dwukołowego robota mobilnego. Wahadło odwrócone jest jednym z fundamentalnych problemów w dziedzinie sterowania i automatyki, znacząco wpływając na stabilność i zachowanie dynamiczne systemów mechanicznych.

W przypadku dwukołowego robota, dynamiczne właściwości wahadła odwróconego stają się kluczowym wyzwaniem, które należy skutecznie zarządzać w celu zapewnienia stabilnego i precyzyjnego poruszania się. Projekt ten skupia się na modelowaniu matematycznym wahadła odwróconego, identyfikacji jego parametrów oraz opracowaniu układu regulacji, który umożliwi efektywne utrzymanie równowagi dwukołowego robota w różnych warunkach.

Model matematyczny:



Wyprowadzenie transmitancji

r – promień koła

mr – masa kola

Jr – bezwładność koła

fr – opór toczny

l – odległość centrum układu spółrzędnych od środka masy

mb – masa robota bez kół

Jb – bezwładność kół robota

fb – opór powietrza

g – siła grawitacji

Środek ciężkości robota

$$x_i = x + l \cdot sin(\theta)$$

$$y_i = y + l \cdot cos(\theta)$$

Obliczamy prędkość różniczkując stronami

$$V_{xi} = \dot{x} + l \cdot cos(\theta)$$

$$V_{yi} = -l \cdot sin(\theta)$$

Obliczmy wektor prędkości

$$v^2 = v_x^2 + v_y^2 = \dot{x}^2 + 2 \cdot L \cdot \dot{x} \cdot \dot{\theta}\, cos(\theta) + l^2 \cdot \dot{\theta}^2\, cos(\theta) + l^2 \dot{\theta}^2\, sin(\theta)$$

Upraszamy używając „jedynki trygonometrycznej”

$$E_k = \frac{1}{2} m_B \cdot \dot{x}^2 + m_B \cdot l \cdot \dot{x} \cdot \dot{\theta}_B \cdot \cos(\theta) + \frac{1}{2} m_B \cdot l^2 \cdot \dot{\theta}^2$$

Uogólnione równie Lagranga dla prędkości obrotowego układu

$$\frac{d}{dt}\left(\frac{\partial E_T}{\partial \dot{x}}\right) - \frac{\partial E_T}{\partial \dot{x}} = Q_R$$

$$\frac{d}{dt}\left(\frac{\partial E_\Gamma}{\partial \dot{\theta}}\right) - \frac{\partial E_T}{\partial \theta} = Q_B$$

Obliczając dla prędkości uogólnionej znajdujemy Q_R

$$\frac{d}{dt}\left(\frac{\partial E_T}{\partial \dot{x}}\right) = (m_B + m_R)\ddot{x} + m_B l \dot{\theta} \cos(\theta)$$

$$\left(\frac{\partial E_T}{\partial \dot{x}}\right) = 0 => Q_R = \ Fw - f_R \dot{x}$$

Analogicznie dla prędkości kątowej uogólnionej

$$\frac{d}{dt}\left(\frac{\partial E_\Gamma}{\partial \dot{\theta}}\right) = m_B l \ddot{x} \cos(\theta) + (J_B + m_B l^2)\ddot{\theta} - m_B l \dot{x} \dot{\theta}^2 \sin(\theta)$$

$$\frac{\partial E_T}{\partial \theta} = -m_B l \dot{x} \cdot \dot{\theta} \sin(\theta)$$

$$Q_B = -m_B g l sin(\dot{\theta}) - f_B \dot{\theta}$$

Podstawiając wszystko do równania Lagranga otrzymujemy równanie opisujące ruch robota

$$\ddot{x} = \ -\frac{m_B l}{m_B + m_R}\left(\ddot{\theta}\cos(\theta) + \dot{\theta}^2 \sin(\theta)\right) + \frac{1}{m_B + m_R}(Fw - f_R \dot{x})$$

$$\ddot{\theta} = -\frac{m_B l}{(J + m_B l^2)}\left(\ddot{x}\cos(\theta) + g sin(\theta)\right) - \frac{f_B}{(J + m_B l^2)}(\dot{\theta})$$

Dokonujemy linearyzacji w punkcie pracy z wykorzystaniem wielomianu Taylora. Dodatkowo zakładamy ze koła są nie odkształcalne.

Punkt pracy taki że:

$$\theta = \ \dot{\theta} = 0$$

Linearyzacja:

$\cos(\theta) = -1$

$\sin(\theta) = \ \theta$

$$\ddot{x} = \frac{m_B l}{m_B + m_R}(\ddot{\theta}) + \frac{1}{m_B + m_R}(Fw - f_R \dot{x})$$

$$\ddot{\theta} = -\frac{m_B l}{(J + m_B l^2)}(\ddot{x}) - \frac{f_B}{(J + m_B l^2)}(\dot{\theta})$$

Wykonując podstawienie otrzymujemy:

$$\ddot{x} = \frac{(Fw - f_R\dot{x})(J + m_B l^2) + m_B^2 l^2 g\theta}{J(m_B + m_R) + m_B m_R l^2}$$

$$\ddot{\theta} = \frac{m_B l(Fw - f_R\dot{x}) + m_B g(m_B + m_R)\theta}{J(m_B + m_R) + m_B m_R l^2}$$

Wykonujemy transformatę Laplaca i obliczmy transmitancje operatorowa układu ciągłego gdzie wejściem układu jest moment obrotowy kola a wyjściem odchylenie w pionie robota

$m = J(m_B + m_R) + m_B m_R l^2$ – podstawienie za całkowity moment bezwładności

$$\frac{\theta(s)}{Fw(s)} = \frac{m_B l s}{m s^3 + f_R(J + m_B l)s^2 - m_b \lg(m_B + m_R)\, s - f_R m_B l g}$$

Podstawiając

$$Fw(s) = rs\,\theta_W(s)$$

Gdzie

$\theta_W(s)$ – transformata drogi kątowej przebytej przez koło

Ostatecznie:

$$\frac{\theta(s)}{\theta_W(s)} = \frac{m_B l s^2}{p s^3 + f_R(J + m_B l)s^2 - m_b \lg(m_B + m_R)\, s - f_R m_B l g}$$

Dane:

r = 0.04 m

mr = 0.18 kg

Jr = 0.000025 kg*m^2

fb = 0.02 $\frac{N}{rad*s}$

l = 0.1 m

mb = 1.2 kg

Jb = 0.0015 kg*m^2

fr = 0.02 $\frac{N}{rad*s}$

g = 9.8 m*s^2

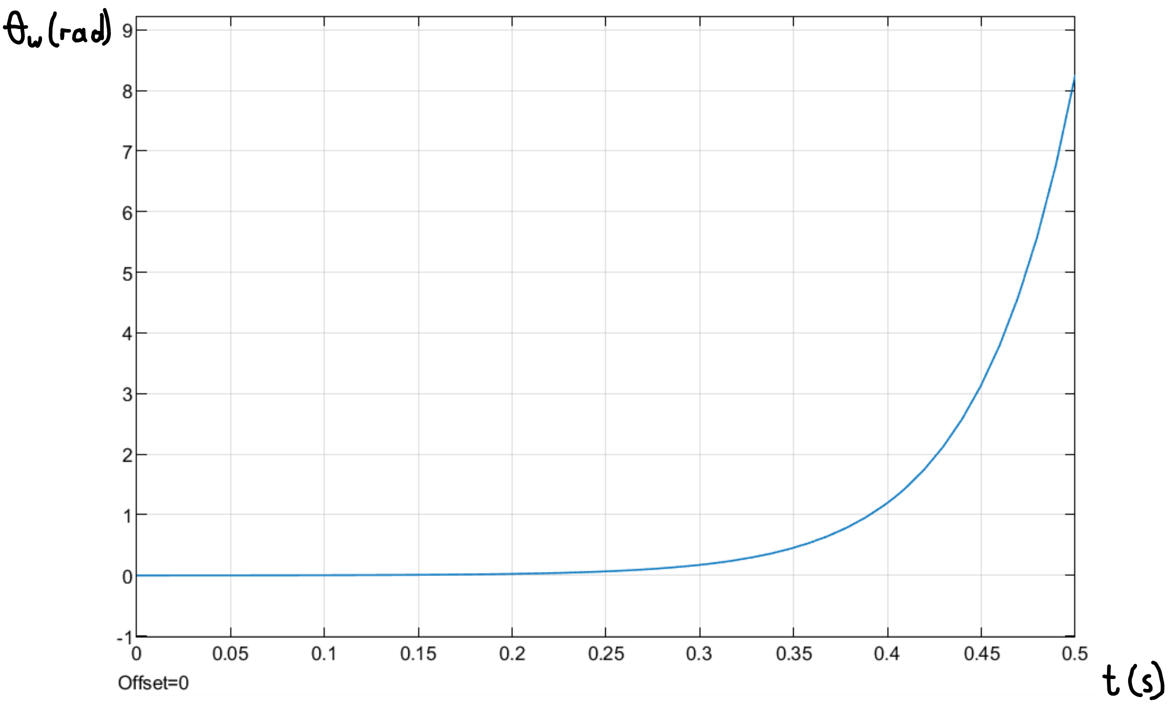
Otrzymana transmitancja układu ciągłego:

```
Gs =

                     0.0048 s^2
  ---------------------------------------------
  0.00423 s^3 + 0.00243 s^2 - 1.623 s - 0.02352
```

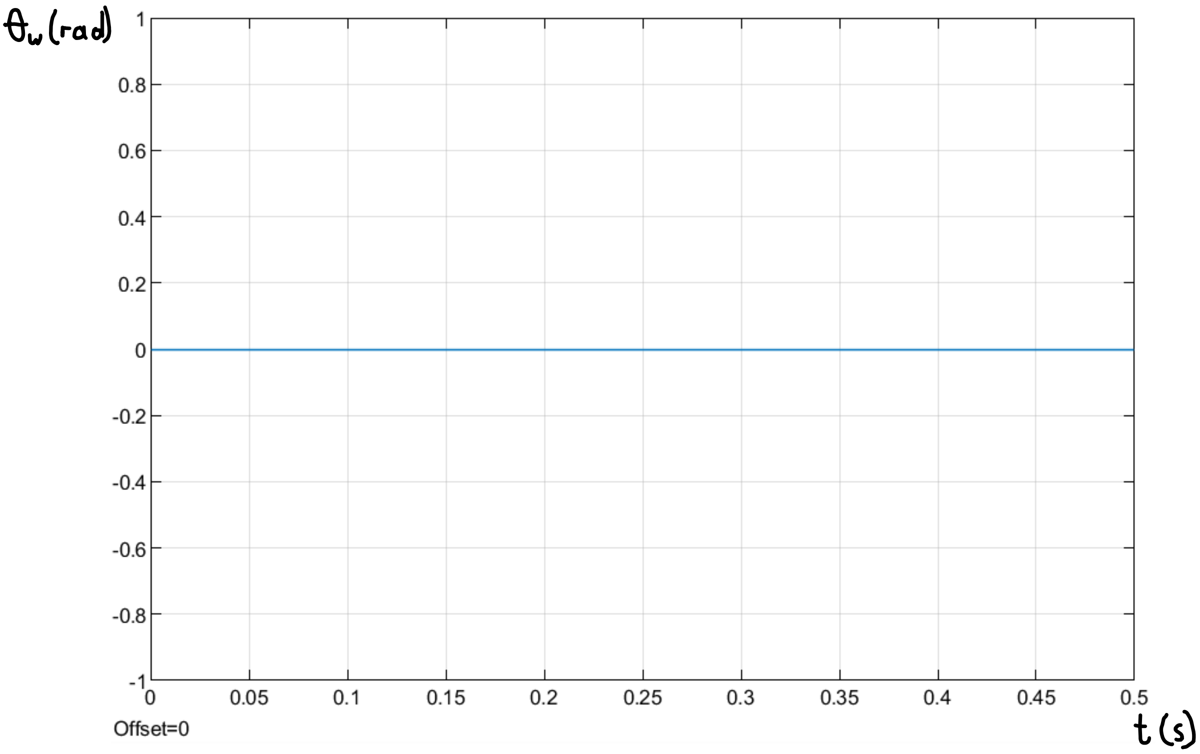
Dyskretyzacja przy Ts = 0.01

```
Gz =

   0.0111 z^2 - 0.0222 z + 0.0111
  ---------------------------------
  z^3 - 2.982 z^2 + 2.926 z - 0.9442
```

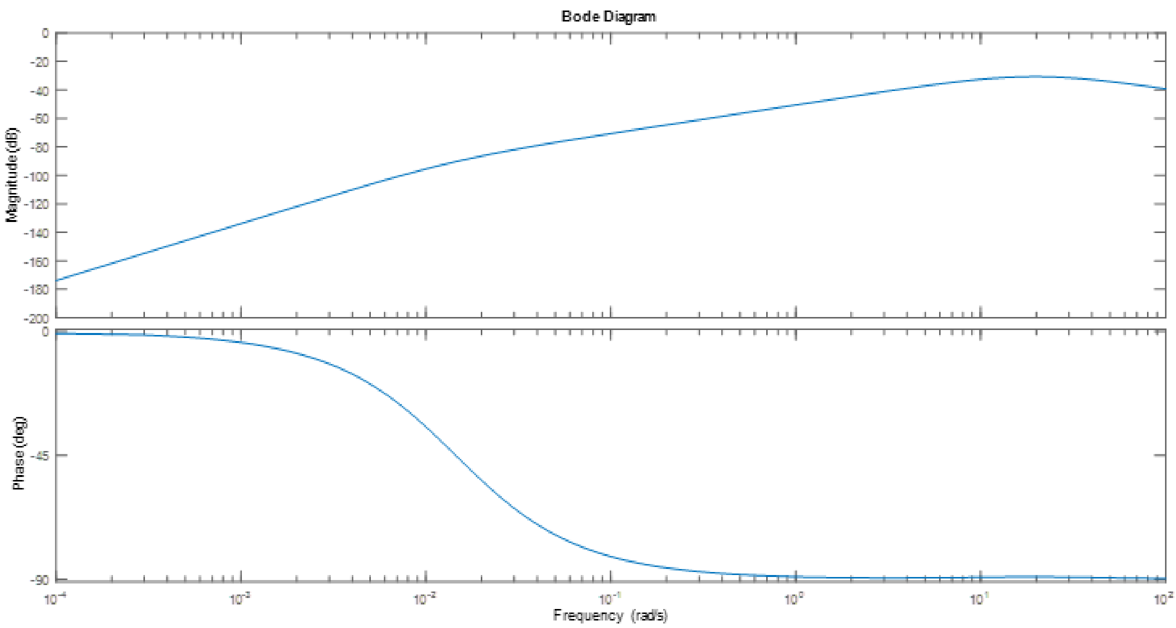
Odpowiedz skokowa na układ ciągłego:



Odpowiedz skokowa układu dyskretnego:



Wykres Bodego układu ciągłego



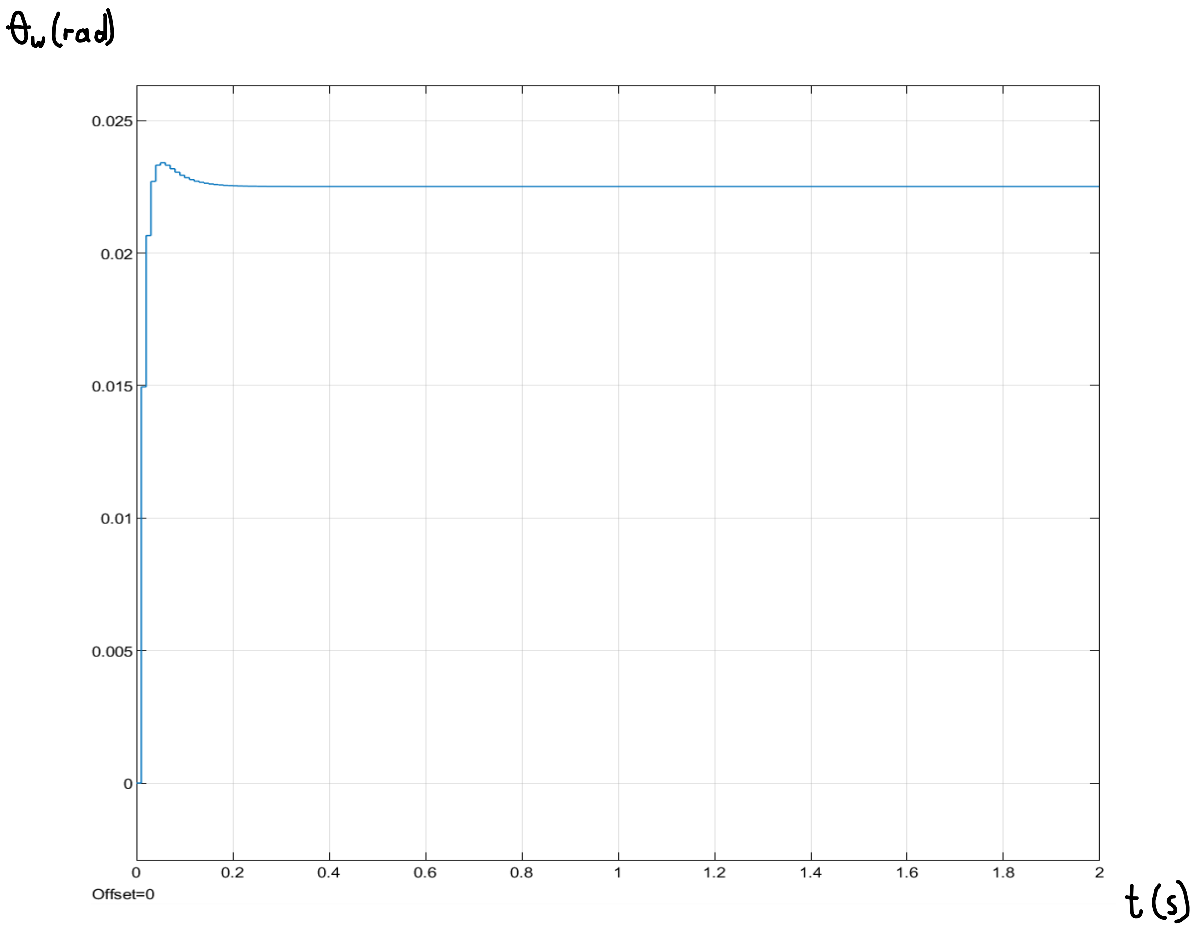
Na wykresie Bodego widać że układ nie ma zapasu wzmocnienia przez co nie możliwe jest wprawienie go w oscylacje, przez co nie jest możliwe uzyskiem nastaw regulatora oscylacyjną metodą Ziglera-Nicolsa. Ponadto odpowiedz skokowa pokazuje ze układ jest sam z siebie niestabilny przez co nie można użyć metody odpowiedzi skokowej do obliczenia nastaw regulatora.

Metoda heurystyczna

Na podstawie odpowiedzi skokowej widać że w układzie czas całkowania Ti musi być dużo większy od czasu różniczkowania Td. Ze względu na tą zależność pomijam element różniczkujący i zakładam ze Td = 0. Czas całkowania przyjmuje bardzo duży i zakładam ze powinien on wynosić Ti = 1500. Natomiast wzmocnienie dobieram tak żeby uzyskać możliwie najmniejsze przeregulowanie. Ja przyjąłem Kp = 75.

Po zastosowaniu moich nastaw i zaimplantowaniu ich do regulatora PID(z) otrzymuje następującą odpowiedz układu.

Wykres położenia od czasu



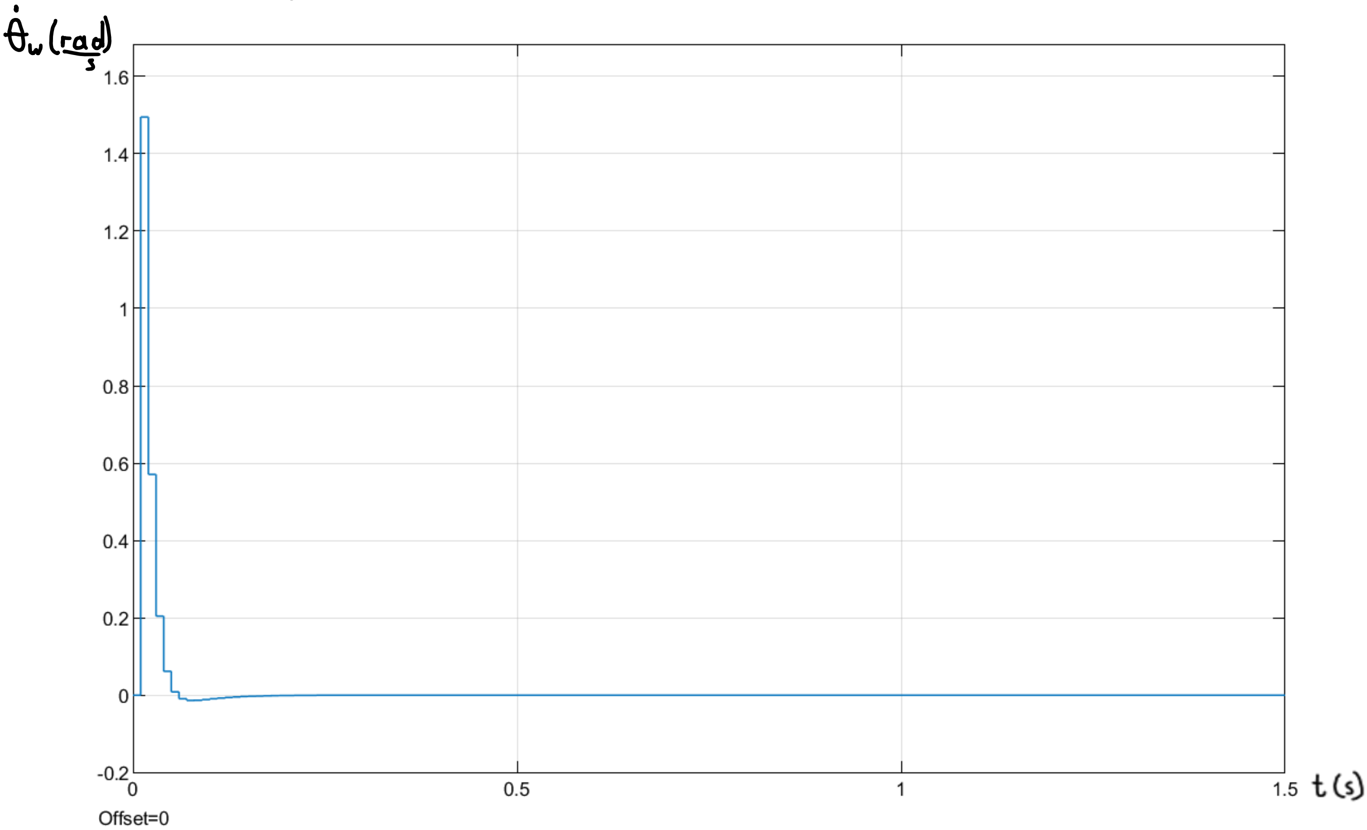
Przeregulowanie w układzie wynosi:

k = $\frac{2.386 - 2.299}{2.299} * 100\% = 3.7\%$

Czas regulacji wynosi:

Tr = 0.2 s

Wykres prędkości obrotowej od czasu



Użycie metody Tune z pakietu Matlab użyta w celu porównania jej z metodą heurystyczną.

Matlab zaproponował nastawy:

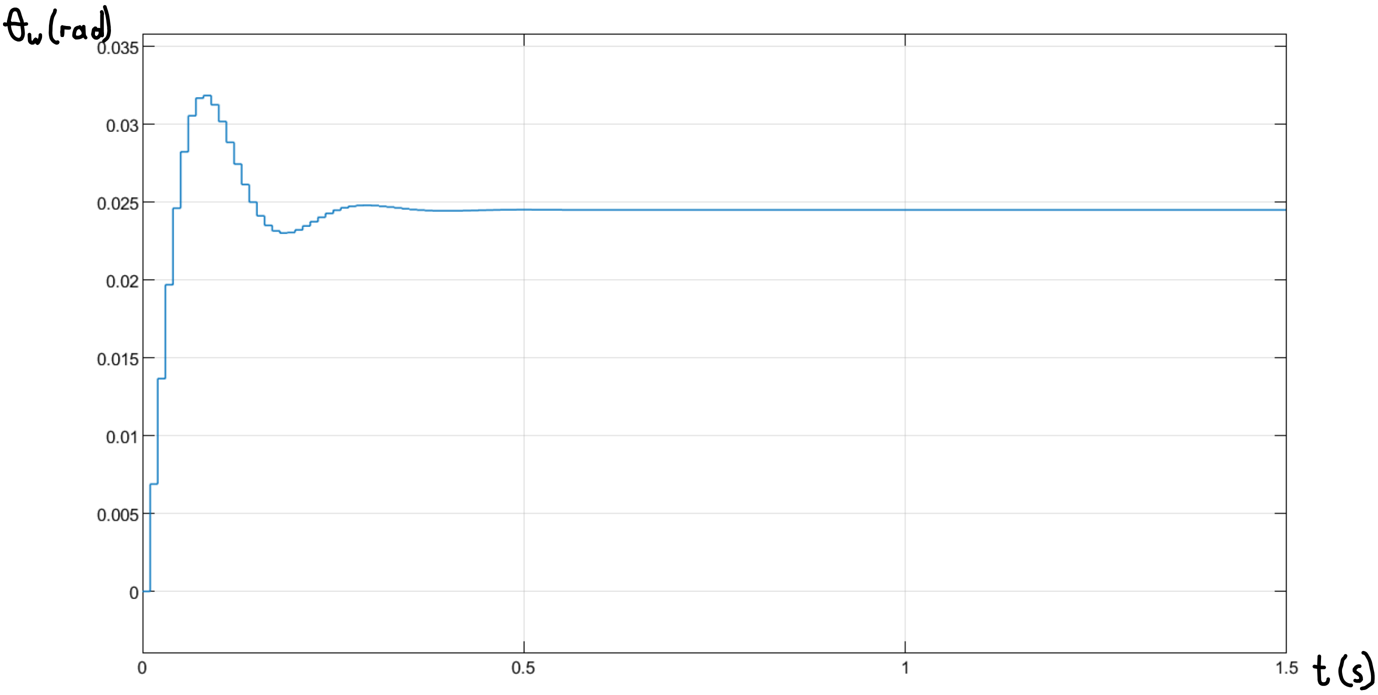
Kp = 34.5896183101414

Ti = 1171.26424695968 s

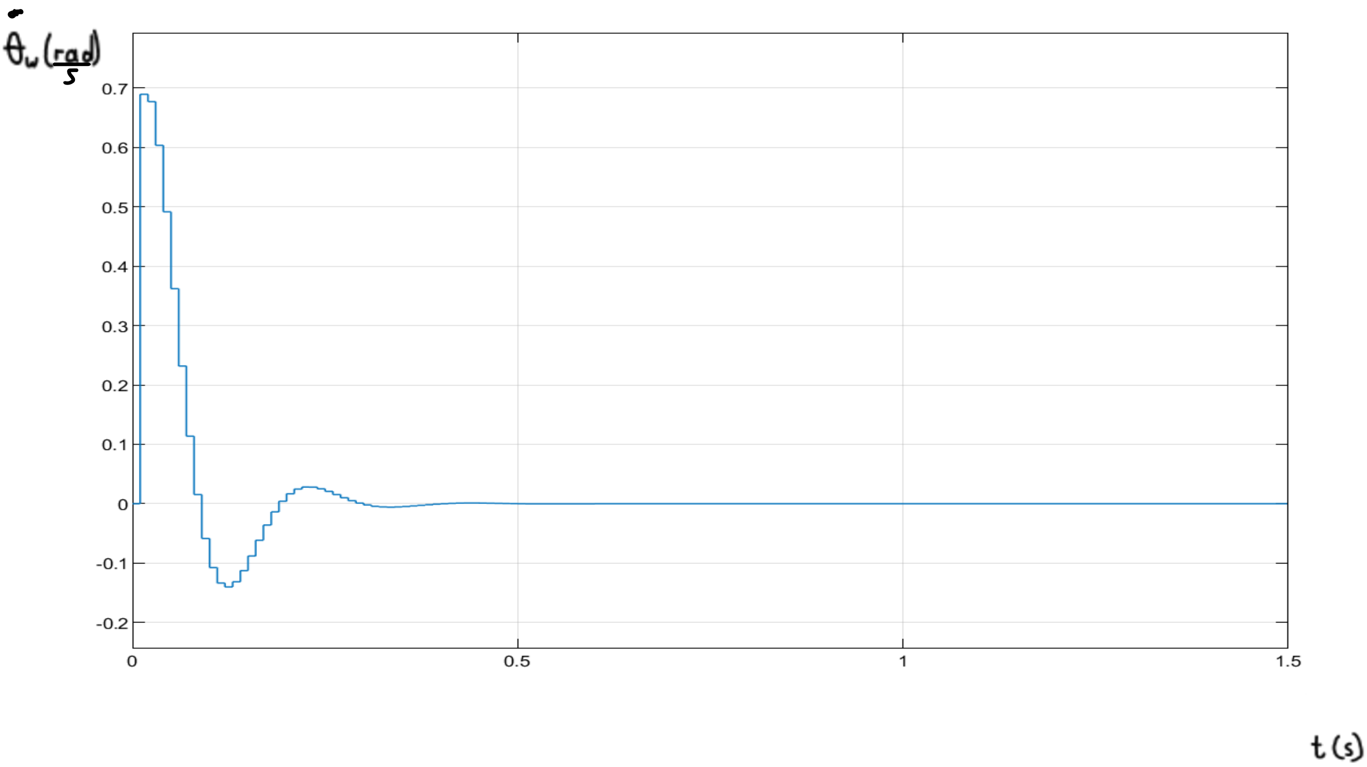
Td = 0 s

Odpowiedz układu wygląd na następująco.

Wykres położenia od czasu



Wykres prędkości kątowej od czasu



Przeregulowanie:

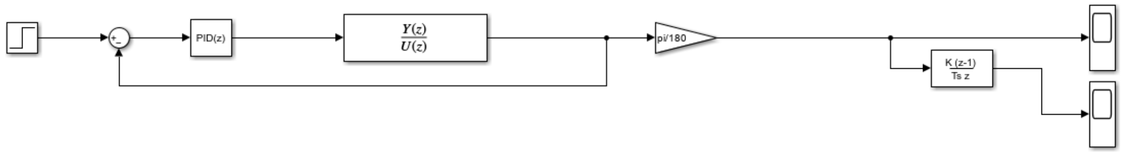
K = $\frac{3.219 - 2.455}{2.455} * 100\% = 31\%$

Czas regulacji:

Tr = 0.3 s

Metoda heurystyczna dała lepszy efekt ze względu na mniejsze przeregulowanie układu i krótszy czas regulacji.

Schemat układu UAR przy obydwu metodach jest taki sam.



Regulator LQR

Regulator Liniowo-Kwadratowy (LQR), stanowi kluczowy element w rozwiązaniu problemu LQ, gdzie układ dynamiczny jest opisany zestawem liniowych równań różniczkowych, a celem jest minimalizacja kosztu zgodnie z zasadami teorii sterowania optymalnego. W kontekście regulacji liniowo-kwadratowej, dąży się do określenia optymalnych nastaw regulatora, sterującego na przykład maszyną czy procesem. Algorytm matematyczny wykorzystywany w regulatorze LQR ma na celu minimalizację funkcji kosztów, gdzie wagi przypisane przez inżyniera wpływają na parametry tej funkcji.

Kod w matlabie umożliwiający obliczenie nastaw regulatora LQR

```
% regulator LQR
Q = 2*C'*C

Q1 = [0.5, 0, 0;
    0, 1, 0;
    0, 0, 40]
R = 1;                          % Control effort weighting

[K1,S1,P1] = lqrd(A,B,Q1,R,Ts);

sys1 = ss(A-B*K1,B,C,D);
% step(sys1)

C_lqr = [1 0 0;
         0 1 0;
         0 0 1]

D_lqr = [0; 0; 0]

sys_lqr=ss(A,B,C_lqr,D_lqr)
d_sys_lqr=c2d(sys_lqr,0.1)
```

Dobranie nastawów regulatora LQR wymaga policzanie równania Riccatiego. Pakiet Matlab posiada gotową funkcje która to oblicza. Macierz bias-ów R = 1. Wynika to z tego ze mój układ to SISO, a macierz wag Q dobrałem w sposób eksperymentalny. Obrazuje ona jakie wpływ dane równania w przestrzeni stanów mają wpływ

Macierz wag:

```
Q1 =

        0.5         0         0
          0         1         0
          0         0        40
```

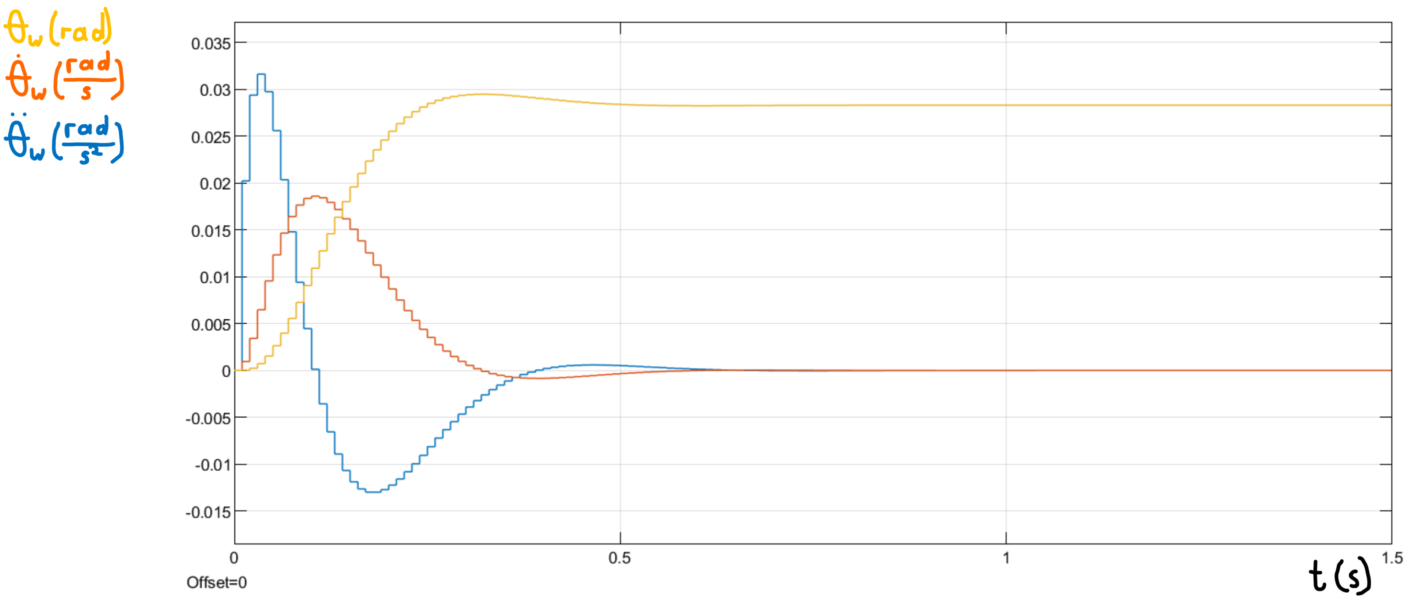
Macierz wzmocnień:

```
K1 =

   13.0109761253033    4.3593489814667    6.50627580630869
```

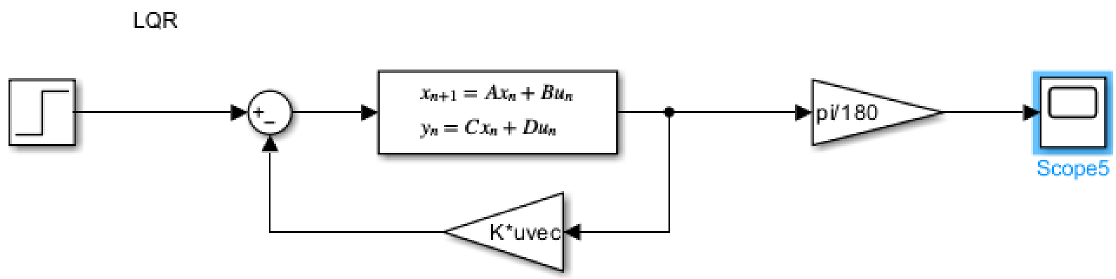
Pozostałe macierze otrzymane jako wyniki równania Riccatiego

```
S1 =

  13.4564137858297    5.04192812025366    6.92405428834472
  5.04192812025366    107.553177140546    41.6037340519001
  6.92405428834472    41.6037340519001    72.4184999518176


P1 =

  0.990871541795987 +   0.00597257359035619i
  0.990871541795987 -   0.00597257359035619i
  0.948497345465233 +                     0i
```

Odpowiedzi układu:



Schemat z Simulinka:

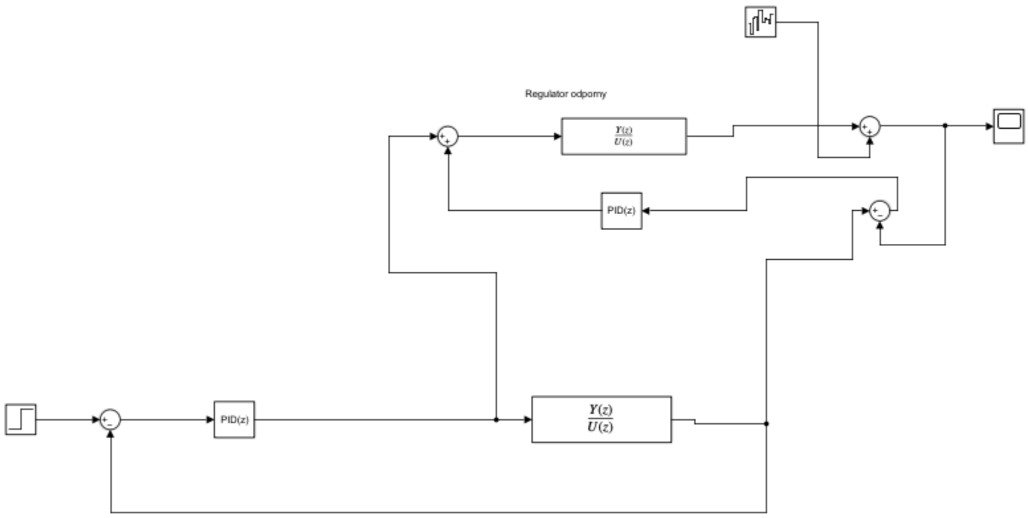


Regulator odporny

Regulator odporny, znany również jako regulator krzepki (ang. robust control), to podejście do projektowania systemów regulacji, które ma na celu zapewnienie stabilności i efektywnego działania układu regulacji nawet w obliczu niepewności, zakłóceń oraz błędów modelowania obiektu. Kluczowym celem jest tolerancja na różnice między rzeczywistym obiektem regulacji a jego matematycznym modelem. Regulatory odporne są projektowane tak, aby utrzymywały stabilność układu nawet w przypadku zmian parametrów obiektu czy nieznanych zakłóceń. Działają w określonym zbiorze parametrów, zapewniając skuteczność regulacji w warunkach rzeczywistych, pomimo potencjalnych niepewności i błędów modelowania.

Jako regulator w torze głównym układu zakładam regulator dobrany metoda heurystyczna, ze względu na to że dał on lepsze efekty niż PID dobrany za pomocą narzędzia tune. Drugi regulator rozniez został dobrany metoda heurystyczna.

Układ sterownia w Simulinku



Transmitancja członu z opóźnieniem

```
Guzd =

              0.0001148 z - 0.000116
  z^(-300) * ----------------------
               z^2 - 2.02 z + 1.021
```

Skrypt w Matlabie który policzył tą transmitancje

```
% opoznienie ukladu

Gucd = tf(L, M, 'InputDelay', 3)
Guzd = c2d(Gucd, Ts, 'ZOH')
```

Nastawy regulatora w torze głównym

P = 75

Ti = 1500 s

Td = 0 s

Nastawy regulatora układu z opóźnieniem

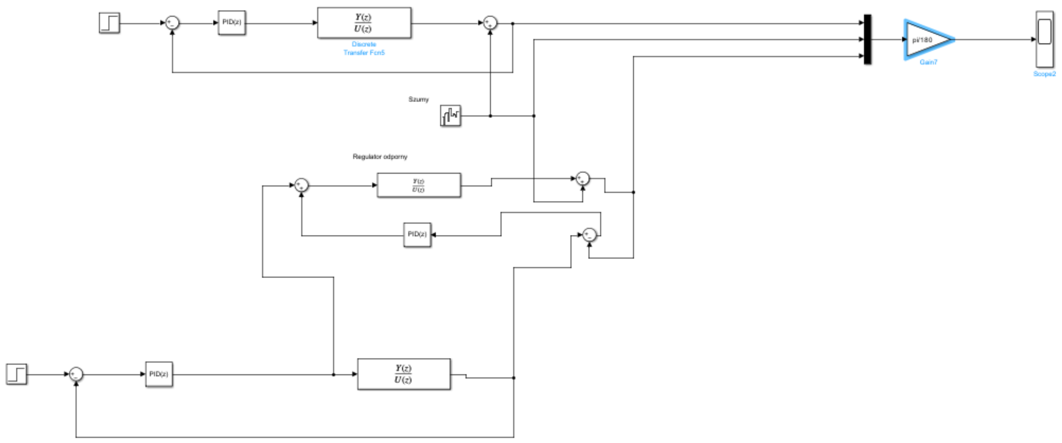
P = 100

Ti = 5000 s

Td = 0 s

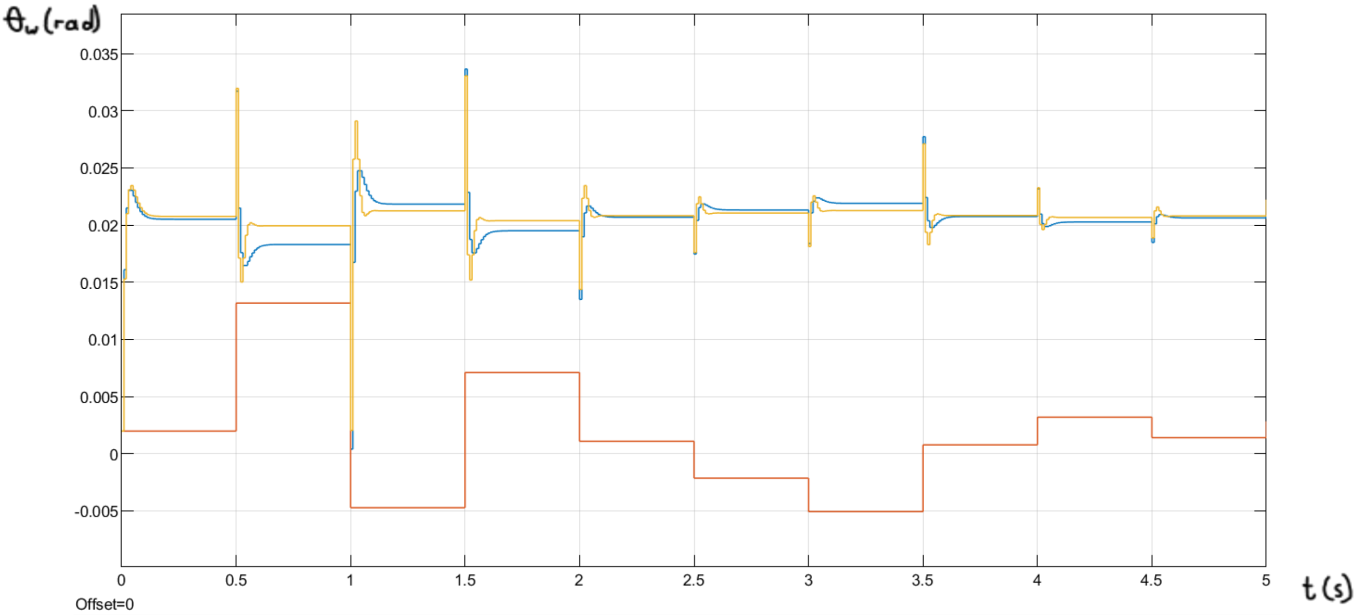
Porównanie układu odpornego ze zwykłem układem posiadającym w torze głównym zwykły regulator PID o nastawach takich samych jak regulator w torze głównym regulatora odpornego.

Układ w Simulinku



Blok generujący zakłócenia jest ustawiany tak żeby generował losowe liczby miedzy 0 a 0.05 co 0.5 sekundy.

Wykres położenia od czasu



Żółty wykres – odpowiedz układu z regulatorem PID

Niebieski wykres – odpowiedz układu odpornego

Pomarańczowy wykres – szumy generowane przez bloczek

Na wykresie widać że peaky zakłóceń są bardziej tumanie przez układ z regulatorem odpornym.

Regulator czasooptymalny

regulator czasooptymalny jest to rodzaj regulatora, którego głównym celem jest minimalizacja czasu potrzebnego do osiągnięcia zadanej wartości wyjścia systemu regulowanego. W przeciwdziedzinie regulacji czasooptymalnej, nacisk kładziony jest na jak najszybsze dostosowanie się systemu do pożądanego stanu, co może być kluczowe w niektórych zastosowaniach.

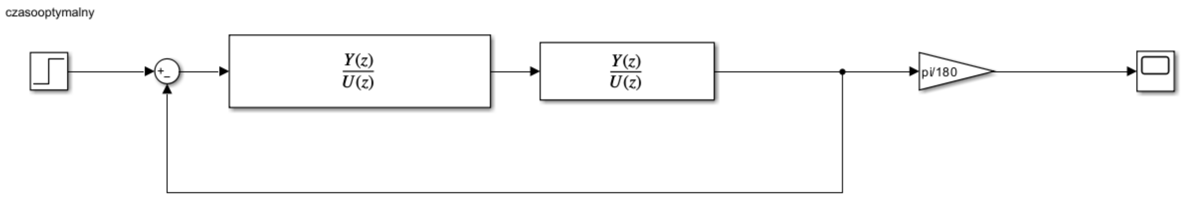
Transmitancja regulatora:

```
Grczopt =

          z^3 - 2.982 z^2 + 2.926 z - 0.9442
  ----------------------------------------------
  1.735e-18 z^3 + 0.0111 z^2 - 0.0222 z + 0.0111
```

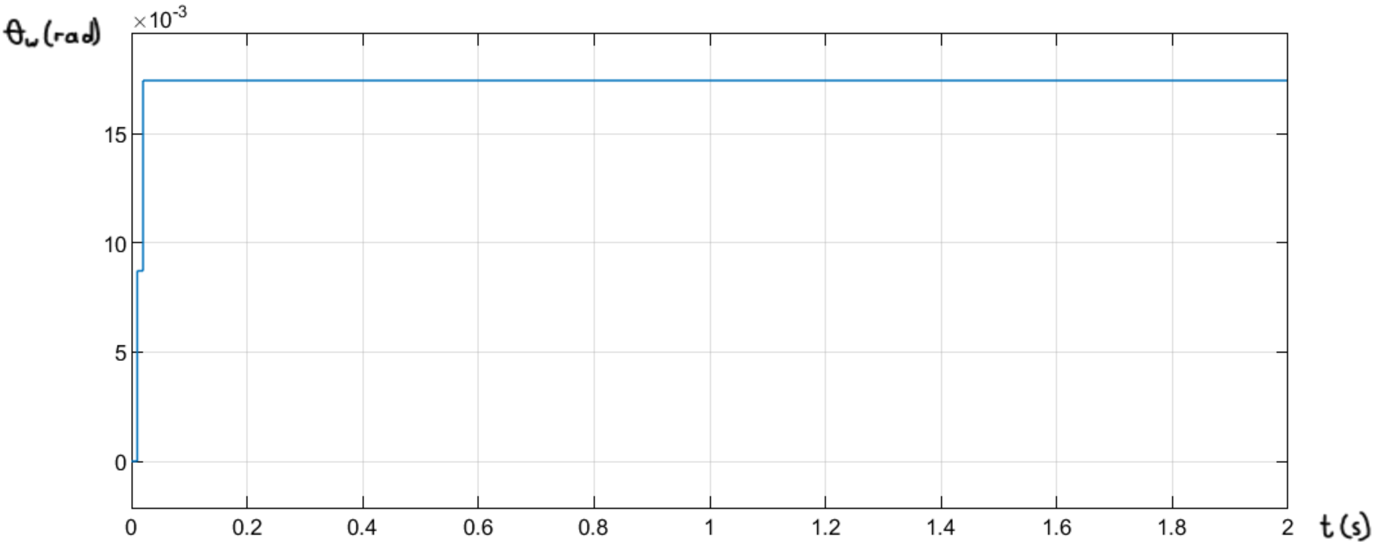
Skrypt w Matlab który pozwolił mi otrzymać transmitancje regulatora.

```
% czasooptymalny
Lz = Gz.num{1}
Mz = Gz.den{1}

order_of_tf = length(Mz) - 1

hig = sum(Lz)

Mr = [hig, zeros(0, order_of_tf)] - Lz
Lr = Mz
Grczopt = tf(Lr, Mr, 1)
```

Schemat w Simulinku



Wykres położenia od czasu



K = 0

Tr = 0.02 s = 2*Ts

Regulator czasooptymalny charakteryzuje się przede wszystkim maksymalnie szybkim czasem regulacji (zwykle równe dwa okresy czasu próbkowania). Wadą tego regulatora jest to że tworzy on bardzo duży sygnał sterujący który nie zawsze jest możliwy do osiągniecia. Dodatkowym problem jest też to że wytwarza on w układzie bardzo duże przyspieszenia. W przypadku tego układu jest on nie użyteczny ze względu na to że układ ten wymaga niskich czasów próbkowania przez co regulator czasooptymalny będzie regulował bardzo szybko układ a co za tym idzie będzie tworzył duży sygnał sterujący który jest nie możliwy do osiągnięcia.

Podsumowanie:

Obiekt jakim jest samobalansujący się dwu kołowy roboty jest układem ciężkim do regulacji ze względu na to ze sam z siebie jest układem niestabilnym. Dodatkowym problem jest też to że wymaga on dosyć niskiego czasu próbkowania (założenie ze Ts > 0.01) powoduje że nie możliwe jest dobranie regulatora PID.

Regulator PID daje dobre warunki regulacji jednak jego problem jest to że nie da się wydłużyć czasu regulacji przez co układ generuje dosyć spore prędkości kątowe które wymagają zaimplementowania serwonapędów będąc w stanie wytrzymać te obciążenia.

Regulator LQR daje najlepsze efekty regulacji. Przyśpieszenia i prędkości w układzie są dosyć niskie przez co nie ma w nim dużych przeciążeń. Czas regulacji jest odpowiednio szybki a przeregulowanie praktycznie nie występuje

Regulator odporny jest lepsza wersją regulatora PID ze względu na to że posiada on dokładnie te same problemy jednak ma on własności tłumienia zewnętrznych zakłóceń które w przypadku tego modelu są dużym problem.

Regulator czasooptymalny daje maksymalnie szybki czas regulacji i zerowe przeregulowanie jednak jest on najgorszym regulatorem ze względu na to ze generuje on nie osiągalny do otrzymania sygnał sterowania i gigantyczne obciążenie elementów które najprawdopodobniej od razu uległy by zniszczeniu.