

LABORATORIUM

(T. Żabiński, PRz 2011)

Modelowanie układów regulacji – pakiety Matlab i Simulink

1. Zamodeluj obiekty regulacji: silnik sterowany prądowo oraz napięciowo traktując je jako układy liniowe (liczbowe wartości parametrów transmitancji: $k_s/J=9$, $k=9$, $T=1$, model nie zawiera momentu obciążenia ani sił tarcia - patrz rys.1). Wykonaj eksperymenty w pakiecie Matlab oraz Simulink i porównaj uzyskane wyniki. Przedstaw na wykresach uzyskane przebiegi prędkości i pozycji dla odpowiedzi skokowej.
2. Stosując wzory podane w materiałach dobierz nastawy regulatorów dla serwomechanizmu prądowego (zadany czas regulacji 0.8 sek, parametry transmitancji silnika należy przyjąć jak w punkcie 1):
 - a) regulator PID o podwójnym zerze – sprzężenie pozycyjne (rys.8),
 - b) regulator PID o podwójnym zerze z filtrem wstępnym – sprzężenie pozycyjne (rys. 9),
 - c) regulator kaskadowy P(położenia) –PI (prędkości) – sprzężenie pozycyjne i prędkościowe (rys. 10),
 - d) regulator PD - sprzężenie pozycyjne (rys. 11).

W sprawozdaniu umieść odpowiedzi skokowe układów uzyskane w pakiecie Matlab oraz Simulink – porównaj wyniki.

3. Dla struktur regulacji z punktu 2 sprawdź parametry śledzenia dla wymuszenia liniowego (Matlab i Simulink). Porównaj uzyskane wyniki i przedstaw wnioski.
4. Zamodeluj w pakiecie Simulink zbiornik z wypływem pod ciśnieniem hydrostatycznym dla danych z wykładu. Rozszerz model o możliwość badania skokowej zmiany otwarcia zaworu na wypływie. Przeprowadź eksperymenty uwzględniające skokową zmianę dopływu i odpływu podczas jednego eksperymentu. Zmianę odpływu należy wprowadzić gdy poziom ustabilizuje się po skokowej zmianie dopływu (wykład). W sprawozdaniu zamieść wykresy i przeprowadź analizę uzyskanych wyników.
5. Przeprowadź symulacje (Simulink, wymuszenie skokowe i liniowe) dla modelu silnika zawierającego statyczną charakterystykę tarcia (model *TestFriction.mdl*). Porównaj wyniki z danymi uzyskanymi w punktach 2 i 3 oraz przedstaw wnioski.
6. Przeanalizuj schematy z rys. 16 i 17 oraz określ funkcje spełniane przez jego poszczególne elementy.

Przygotowanie do ćwiczenia:

- teoretyczne odpowiedzi skokowe obiektów z punktu 1 (*Teoria Sterowania – skrypt*, rozdział 2, punkt 2.4.),
- kody realizujące symulacje w pakiecie Matlab - *Teoria Sterowania – skrypt*,
- wpływ tarcia na zachowanie się serwomechanizmów (uchyby ustalone, efekt *Stribeck*, *stick-slip*, *limit cycles*),

Uwaga: wzory z wykładu 8 należy przekształcić do postaci regulatora PID stosowanego w

Simulink: $PID = k_p + \frac{k_i}{s} + k_d s$.

Literatura:

L. Trybus (2005). Teoria Sterowania – skrypt. *PRz*.

M. Szymkat (1993). Komputerowe wspomaganie w projektowaniu układów regulacji, *WNT, Warszawa*.

Literatura dotycząca zagadnień związanych z modelowaniem i kompensacją tarcia – dal zainteresowanych:

<http://www-lag.ensieg.inpg.fr/canudas/>

Armstrong-Hélouvry, Dupont B., P. and Canudas de Wit C. (1994). A survey of models, analysis tools and compensation methods for the control of machines with friction. *Automatica*, **30**, 1083-1138.

Tataryn, P.D., Sepehri N. and Strong D. (1996). Experimental comparison of some techniques for the control of manipulators with stick-slip friction. *Control Eng. Practice*, **4** (9), 1209-1219.

Li, Y.-F. (1999). Motion control subject to nonlinearities and flexibility. *Technical Report, Mechatronics Lab, Department of Machine Design, Royal Institute of Technology, Stockholm*, TRITA-MMK 1999:15.

Canudas de Wit C., Olsson H., Åström K. J., Lischinsky P. (1995). A New Model for Control of Systems with Friction. *IEEE Trans. on Automatic Control*, **40** (3), 419-425.

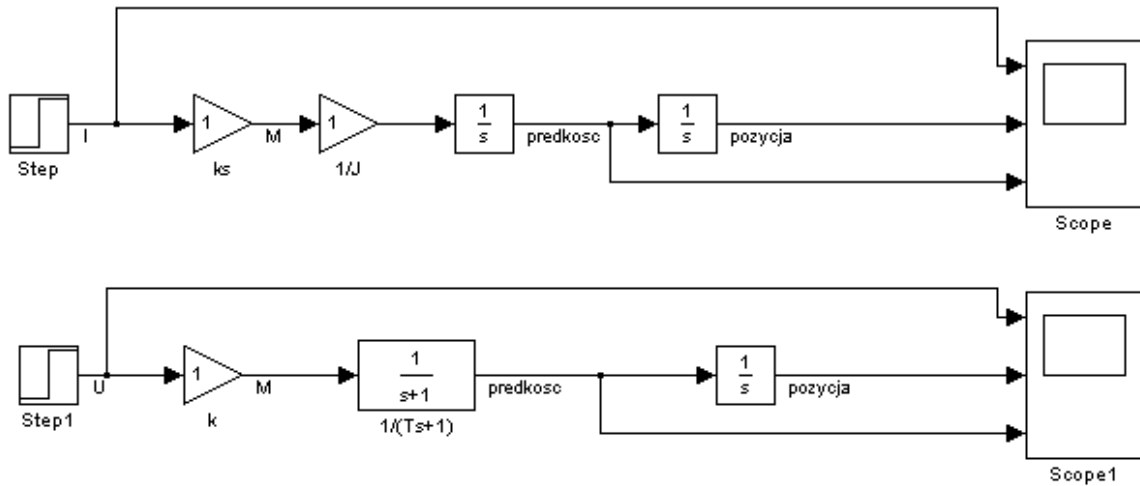
Hensen, Ronnie H.A. (2002). Controlled Mechanical Systems with Friction. *Ph.D. thesis Eindhoven University of Technology, Department of Mechanical Engineering, Systems and Control Group*.

Canudas de Wit, C. (2003). Modelling and Control of Systems with Dynamic Friction. *Mini-Course on: Control of Systems with Dynamic Friction*.

Bona B., Indri M., Smaldone N. (2003). Nonlinear friction estimation for digital control of direct-drive manipulators. In *European Control Conference, Cambridge*.

1. Tworzenie modeli i przeprowadzanie symulacji w pakiecie Simulink – wprowadzenie

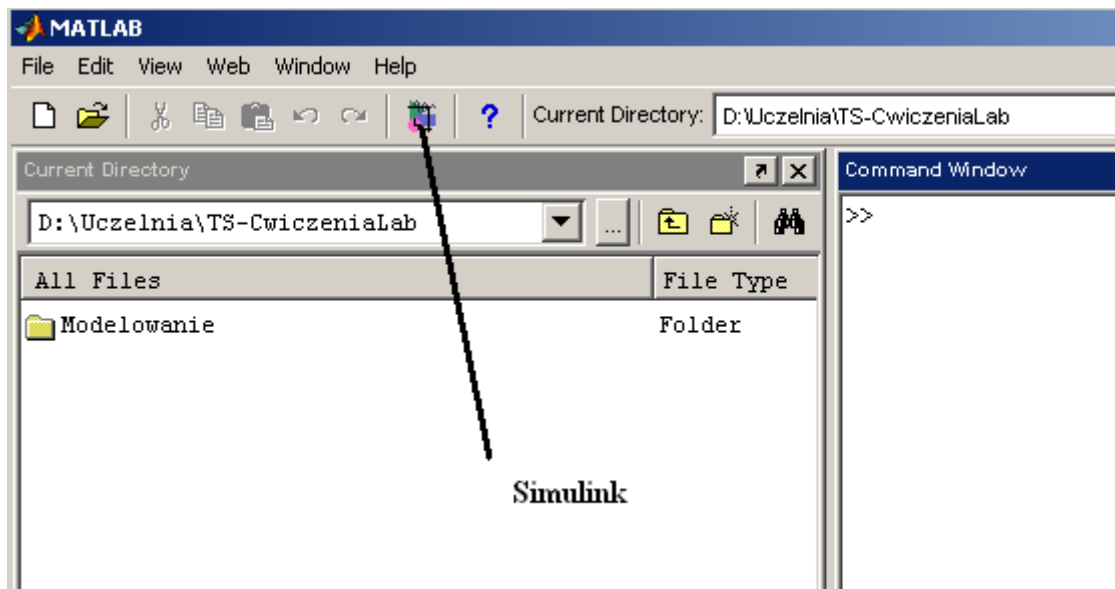
Poniżej przedstawiono modele, które należy utworzyć zgodnie z punktem 1 instrukcji.



Rys.1. Modelowanie obiektów regulacji

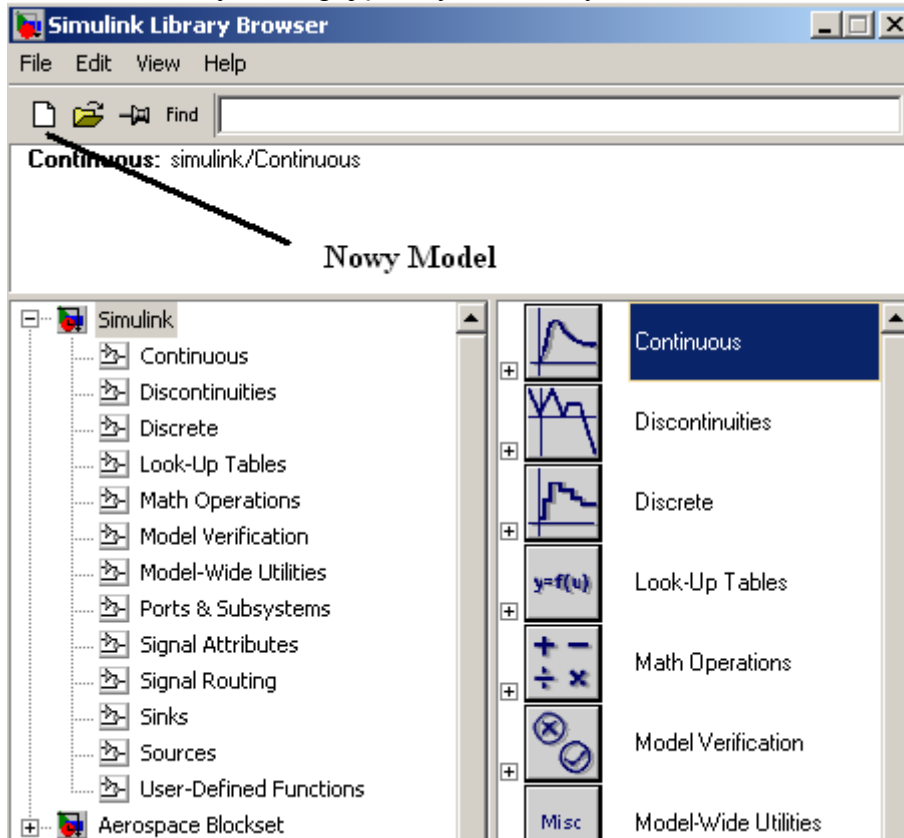
Tworzenie schematu:

a) Uruchom okno pakietu Simulink – rys 2.



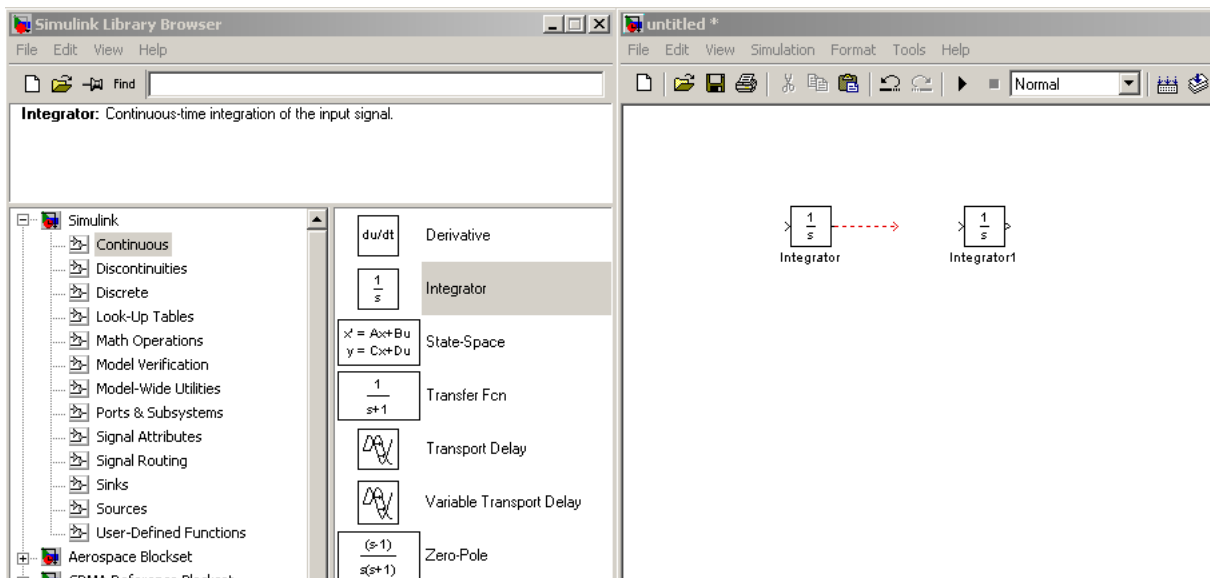
Rys.2. Uruchomienie pakietu Simulink

b) W oknie Simulink – wybierz opcję nowy model – rys. 3.



Rys.3. Wybór nowego modelu Simulink

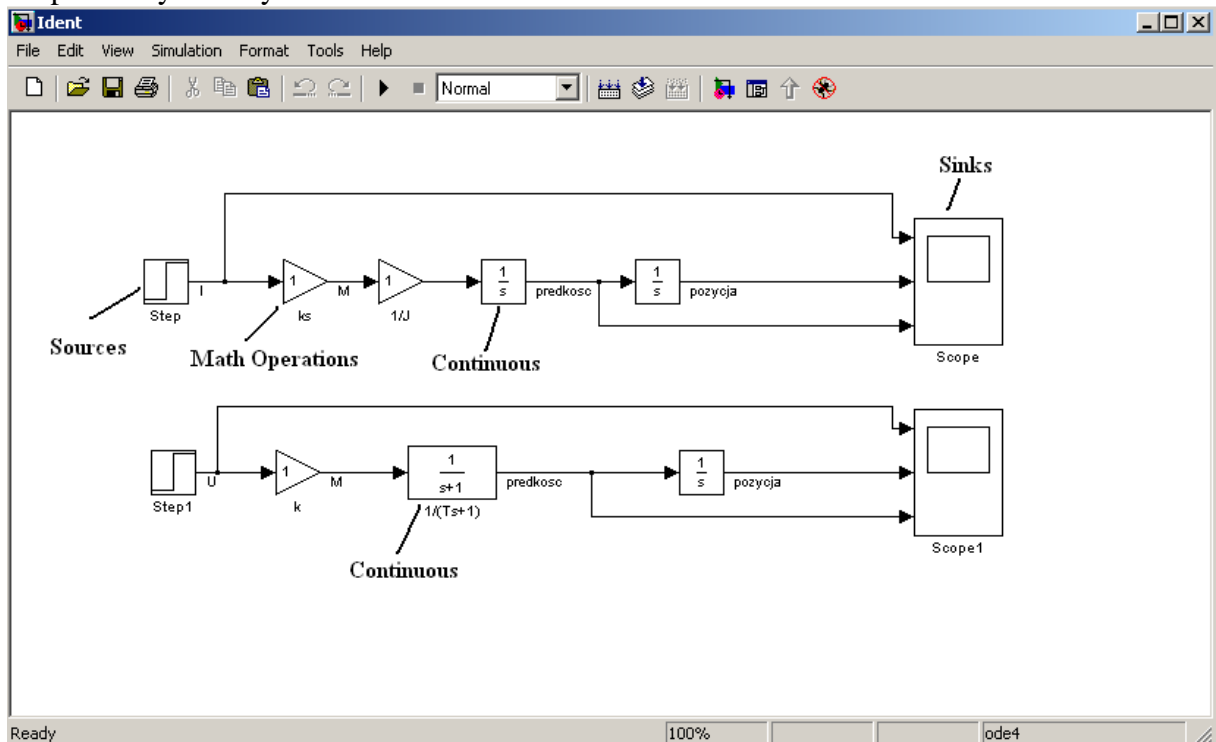
c) W nowym oknie utwórz schemat (rys.1) przeciągając bloki z biblioteki Simulink na okno modelu – rys.4.



Rys.4. Tworzenie nowego modelu Simulink

Wejścia oraz wyjścia bloków można połączyć przy pomocy myszki. Wyjście bloku wybiera się wskazując go myszką i naciskając lewy przycisk, następnie (z naciśniętym lewym przyciskiem myszki) należy doprowadzić sygnał do wejścia innego bloku.

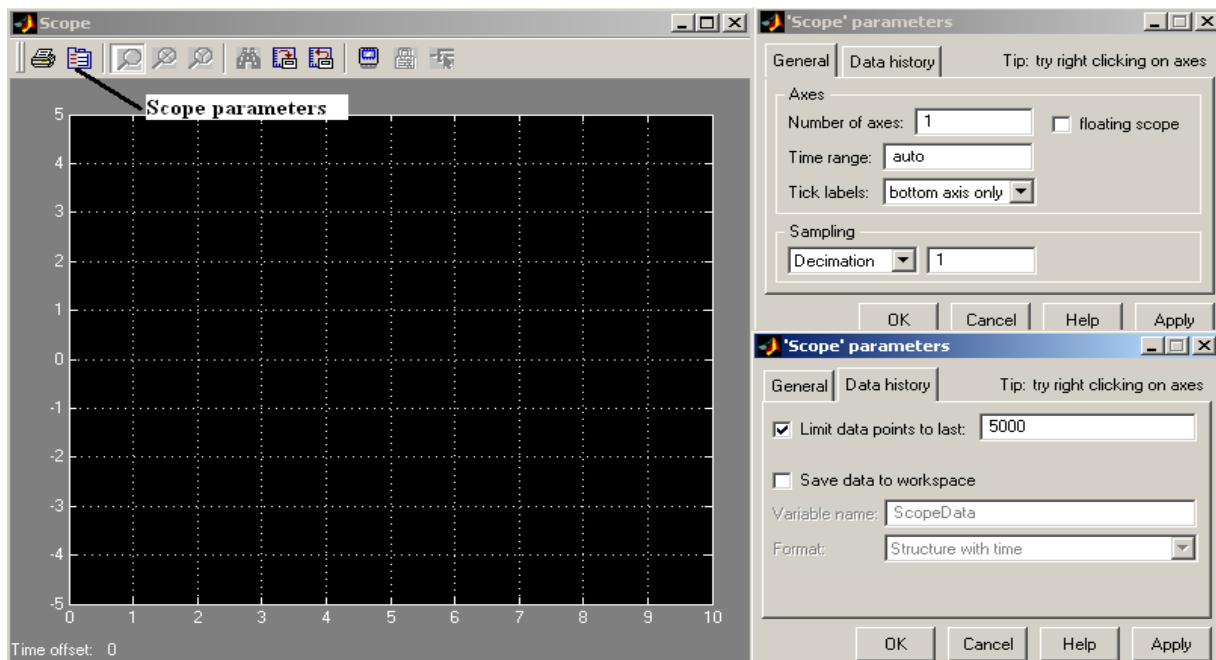
d) Bloki potrzebne do stworzenia schematu z rys. 1 znajdują się w bibliotekach Simulink pokazanych na rys.5.



Rys.5. Bloki z rys. 1 oraz ich przynależność do bibliotek Simulink

Dodatkowe ustawienia:

Blok *Scope* – w bibliotece znajduje się blok *Scope* o jednym wejściu. W celu zwiększenia ilości wejść należy wybrać opcję pokazaną na rys. 6.



Rys.6. Parametry bloku *Scope*

W *Data history* dostępna jest opcja *Save data to workspace* dzięki której dane z wykresów są automatycznie zapisywane (po wykonaniu symulacji) w zmiennej *ScopeData* dostępnej

w przestrzeni roboczej Matlab. Informacje o ilości oraz nazwach zmiennych dostępnych w danym momencie w przestrzeni roboczej można uzyskać przy pomocy komendy *whos*. Informacje o strukturze danych przechowywanych w zmiennej uzyskuje się wpisując jej nazwę w linii poleceń Matlab i naciskając *enter*.

Przykładowe informacje dla schematu z rys. 1:

```
>> whos
  Name          Size          Bytes Class
  ScopeData     1x1           321758 struct array
  ScopeData1    1x1           321760 struct array
  tout         1000x1          8000 double array

Grand total is 81143 elements using 651518 bytes

>> ScopeData

ScopeData =

    time: [10001x1 double]
   signals: [1x3 struct]
  blockName: 'Ident/Scope'

>> ScopeData.signals

ans =

1x3 struct array with fields:
    values
  dimensions
    label
    title
  plotStyle
```

Przy pomocy danych zapisanych w przestrzeni roboczej możliwe jest wykonywanie dodatkowych wykresów - funkcja *plot* jak pokazano poniżej.

```
>> plot(ScopeData.time,ScopeData.signals(2).values);grid
```

Możliwy jest również zapis (odczyt) danych do pliku w formacie *mat*:

Zapis zmiennej *ScopeData* do pliku o nazwie *ScopeData.mat*

```
>> save('ScopeData.mat','ScopeData');
```

Odczyt danych z pliku *mat* można wykonać przy pomocy komendy *load*:

```
>> load('ScopeData.mat');
```

bądź klikając dwukrotnie na nazwie pliku w oknie *Current Directory* pakietu Matlab. Po wykonaniu wczytania danych należy sprawdzić przy pomocy *whos* jakie dane pojawiły się w przestrzeni roboczej.

Pomocnicze komendy:

- usuwanie wszystkich zmiennych z przestrzeni roboczej

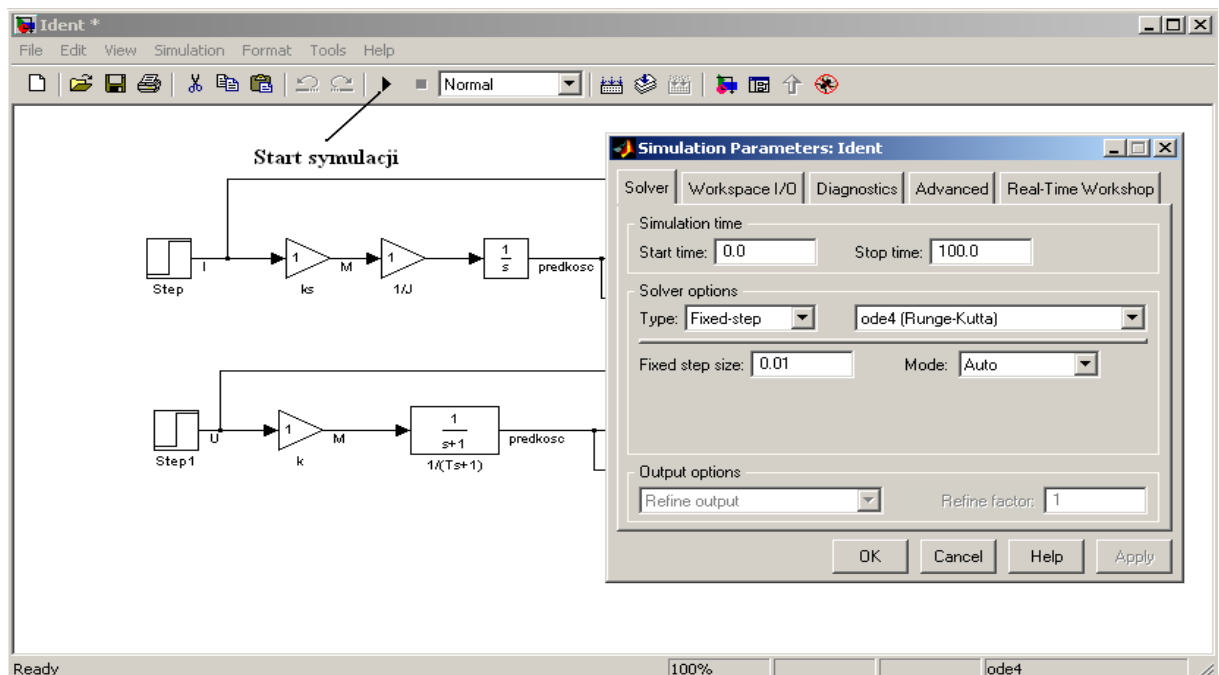
>> **clear all**

- czyszczenie okna komend

>> **clc.**

Przeprowadzenie symulacji:

Symulację uruchamia się naciskając przycisk *Start Simulation* – patrz rys. 7. Parametry symulacji określa się przy pomocy opcji *Simulation* -> *Simulation parameters*....

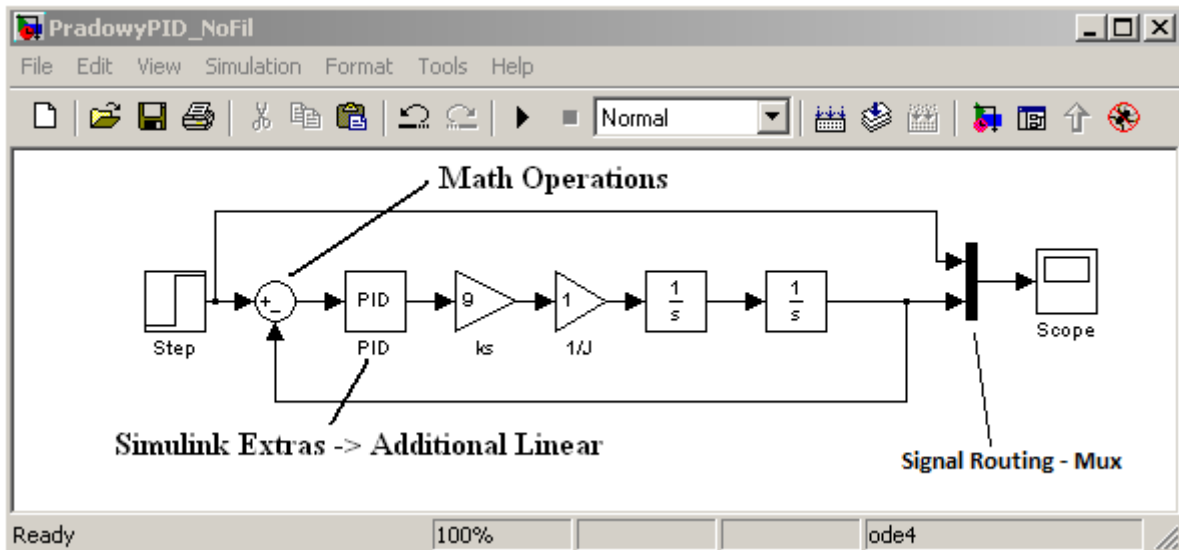


Rys.7. Parametry i start symulacji

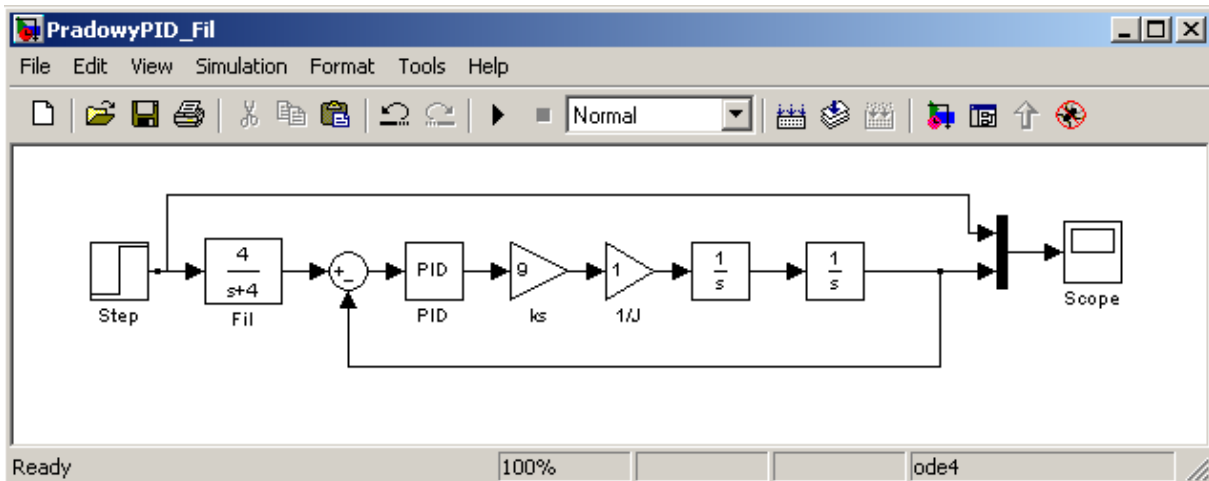
Jeżeli nie zostanie określone inaczej parametry symulacji *Solver options* należy ustawiać na *Fixed-step* oraz *ode4 (Runge-Kutta)*. Pozostałe parametry takie jak *Start time*, *Stop time* oraz *Fixed step size* należy dobierać w zależności od przeprowadzanej symulacji.

Aby obserwować przebiegi podczas symulacji należy otworzyć (dwukrotnie klikając na bloku *Scope*) okno wykresów przed symulacją.

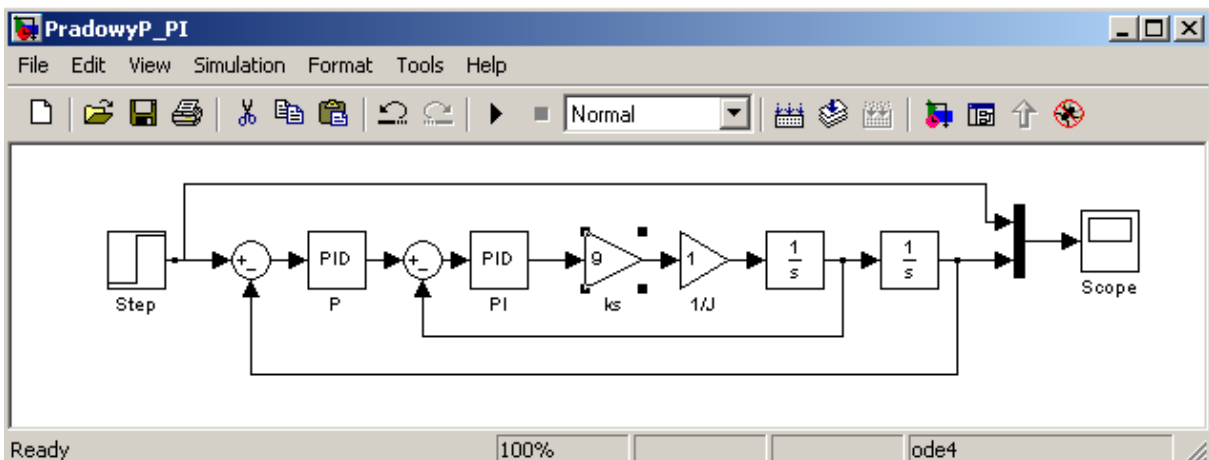
2. Struktury sterowania – serwomechanizmy – Simulink



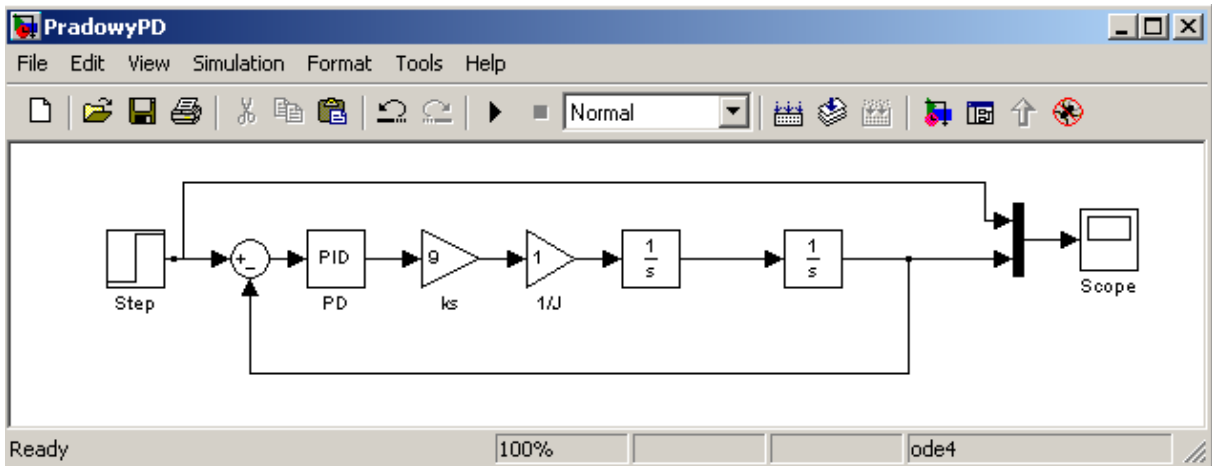
Rys.8. Układ z regulatorem PID o podwójnym zerze – sprzężenie pozycyjne



Rys. 9. Układ z regulatorem PID o podwójnym zerze z filtrem wstępnym – sprzężenie pozycyjne

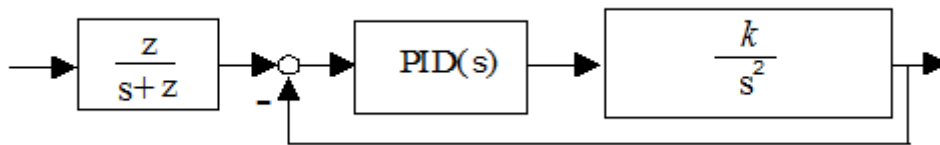


Rys. 10. Układ z regulatorem kaskadowym P(położenia) –PI (prędkości) – sprzężenie pozycyjne i prędkościowe



Rys. 11. Układ z regulatorem PD - sprzężenie pozycyjne

Strojenie regulatorów PID (rys. 8 i 9):



Rys. 12. Układ z regulatorem PID w sprzężeniu pozycyjnym

Ciągła transmitancja regulatora PID dana jest wzorem (jak w pakiecie Simulink):

$$PID(s) = k_p + \frac{k_i}{s} + k_d s = k_r \frac{(s+z)^2}{s}, \quad k_r = k_d, \quad z = \frac{2k_i}{k_p}, \quad k_d = \frac{k_p^2}{4k_i}, \quad k = \frac{k_s}{J}. \quad (1)$$

PROBLEM. Należy dobrać takie nastawy k_p , k_i , k_d , aby uzyskać przebiegi aperiodyczne krytyczne z zadaniem czasem regulacji t_r .

Transmitancja układu otwartego rozważanej pętli regulacji wyraża się zależnością

$$G_{ow}(s) = K \frac{(s+z)^2}{s^3}, \quad K = k \cdot k_r \quad (2)$$

Przeprowadzając metodę projektowania regulatora analogicznie do przedstawionej w wykładzie 8 otrzymuje się następujące zależności opisujące wartość zera z oraz wzmocnienia K_1 (dla punktu rozwidlenia linii pierwiastkowych):

$$z = \frac{4}{t_r}, \quad K_1 = \frac{27}{4} z \quad (3)$$

Ostatecznie uzyskuje się wzory dla nastaw regulatora:

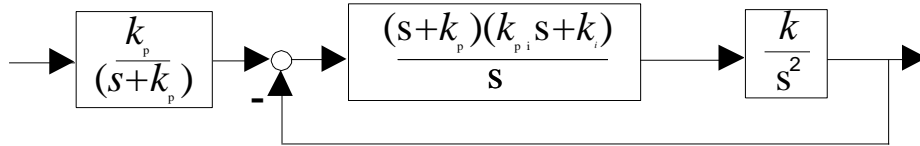
$$k_p = \frac{2 \cdot K_1 \cdot z}{k}, \quad k_i = \frac{K_1 \cdot z^2}{k}, \quad k_d = \frac{K_1}{k} \quad (4)$$

Strojenie struktury P-PI (rys. 10):

Metoda doboru nastaw jest realizowana identycznie jak powyżej. Regulatory oraz wzmocnienie obiektu określono następująco:

$$P = k_p, \quad PI = k_{pi} + \frac{k_i}{s}, \quad k = \frac{k_s}{J}. \quad (5)$$

Przekształcając układ z rys. 10 otrzymano:



Rys. 13. Układ z regulatorem P-PI po wstępnym przekształceniu

Stosując założenie o podwójnym zerze, regulator PID przekształca się do postaci:

$$PID(s) = k_r \frac{(s+z)^2}{s}, \quad k_r = k_{pi}, \quad z = k_p, \quad k_p = \frac{k_i}{k_{pi}}. \quad (6)$$

Na podstawie zależności 2, 3 oraz 6 nastawy struktury P-PI dane są wzorami:

$$k_p = z, \quad k_{pi} = \frac{K_1}{k}, \quad k_i = \frac{z \cdot K_1}{k} \quad (7)$$

Strojenie struktury PD (rys. 11):

Regulator PD oraz wzmocnienie obiektu określono następująco:

$$PD = k_p + k_d s, \quad k = \frac{k_s}{J}. \quad (8)$$

Stosując metodę linii pierwiastkowych Evansa otrzymano następujące zależności dla nastaw regulatora PD:

$$k_p = \frac{36}{k \cdot t_r^2}, \quad k_d = \frac{12}{k \cdot t_r} \quad (9)$$

Aby uzyskać przebiegi aperiodyczne krytyczne w strukturze pokazanej na rys. 11 układ należy uzupełnić o filtr wstępny postaci:

$$\frac{1}{\frac{k_d}{k_p} s + 1} \quad (10)$$

bądź wyposażyć układ w sprzężenie tachometryczne.

3. Śledzenie wymuszenia liniowego

Na schematach 8-11 blok *Step* należy zastąpić blokiem *Ramp* z biblioteki *Simulink* ->*Sources*.

4. Badanie odpowiedzi układu z modelem uwzględniającym tarcie

(Zaczerpnięto z: T. Żabiński, A. Turnau: COMPENSATION OF FRICTION IN ROBOTIC ARMS AND SLIDE TABLES, IFAC 2005)

Friction is a highly nonlinear phenomenon found in all mechanical systems. From the point of view of control, friction causes the following unfavourable effects: stick-slip motion, significant tracking errors, large settling time and limit cycles. Control methods used for precise and high-performance motion systems and also complex control algorithm (like optimal control) require efficient compensation of friction. Theoretically a rule compensating friction is simple: apply a force opposite to the instantaneous friction force. However, it is difficult to guarantee the robustness in both stability and performance criteria when fixed model friction compensation is used because friction is nonstationary and position-dependent phenomena. Thus methods that do not require a complete knowledge of the friction dynamics (Tataryn, *et al.* 1996; Li, 1999) have significant practical meaning. Many of such methods have been developed for example: stiff PD, dither, impulsive control, smooth robust nonlinear feedback, etc.

At the same time many intensive studies are being worked on to develop and identify friction models. Among those friction models already described in different sources the most popular and mainly accepted one is the LuGre (Canudas, *et al.* 1995) model with its modifications. The LuGre model links both steady-state friction curve and friction dynamic characteristic. This model, although relatively simple, reflects most of friction induced phenomena very significant for feedback control. Unfortunately, a good estimation of model parameters (especially parameters of dynamic part of the model) is often quite difficult to be achieved in practice, especially when typical industrial control environment and sensors are used. An excellent review of models, analysis and control tools for friction compensation can be found in the survey paper by Armstrong-Hélouvry, *et al.* (1994).

FRICTION MODELLING

There are static and dynamic friction models. Static models include the observed friction phenomena like: Coulomb, viscous, static friction and Stribeck effect and their possible combinations. They do not include frictional memory. Dynamic friction models are more complex. They describe such phenomena as: presliding displacement, or frictional lag. A number of the dynamic models are proposed: Dahl, Bliman, Sorine, LuGre and others. An excellent and brief review of models can be found in the Ph.D. project by Hensen (2002).

The LuGre model corresponds to: steady-state friction curve and the presliding phase by means of flexible bristels, representing the contact points of the moving surfaces. The basic model has the form (Canudas, 2003)

$$\frac{dz}{dt} = v - \frac{\sigma_0}{g(v)} z |v| \quad (1)$$

$$F = \sigma_0 z + \sigma_1 \frac{dz}{dt} + f(v)$$

where z denotes the average bristle deflection. The model behaves like a spring for small displacement where σ_0 is the stiffness of the bristles and σ_l the damping of the elastic bristles. The function $g(v)$ describes steady-state friction curve and $f(v)$ is the viscous friction. An equation of $g(v)$ that has been proposed (Canudas, *et al.* 1995) to describe the Stribeck effect is

$$g(v) = F_c + (F_s - F_c)e^{-(v/v_s)^2} \quad (2)$$

where F_c is the Coulomb friction, F_s is the static friction force and v_s is the Stribeck velocity. A modified description of the Stribeck curve has been proposed by Bona *et al.* (2003) in the form of

$$g(v) = \alpha_0 + \alpha_1 e^{-(v/v_1)\text{sgn}(\dot{\phi})} + \alpha_2 (1 - e^{-(v/v_2)\text{sgn}(\dot{\phi})}) \quad (3)$$

which gives possibility to achieve better data fitting ($F_s = \alpha_0 + \alpha_1$, $F_c = \alpha_0 + \alpha_2$). The steady-state part of the LuGre model (Canudas, *et al.* 1995)

$$F_{ss} = g(v)\text{sgn}(v) + f(v) \quad (4)$$

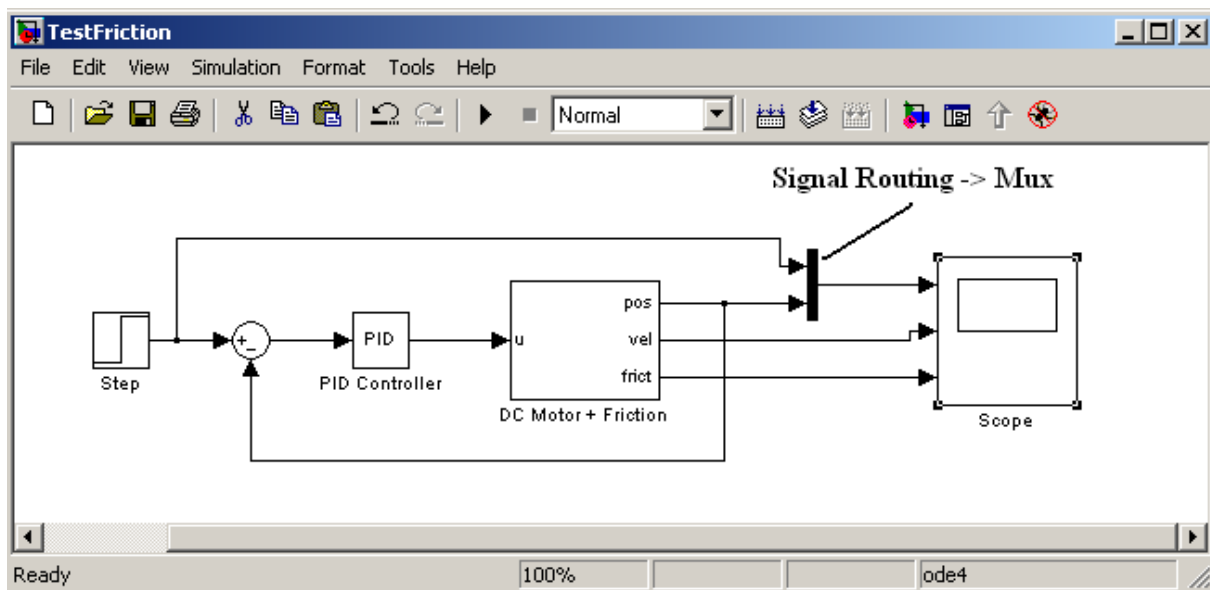
is used here to describe nonlinear friction torques. Viscous friction can be described as a linear function of velocity (Canudas, *et al.* 1995)

$$f(v) = F_v v \quad (5)$$

or like proposed by Bona *et al.* (2003) for direct-drive systems

$$f(v) = F_v v + F_{v2} v^2. \quad (6)$$

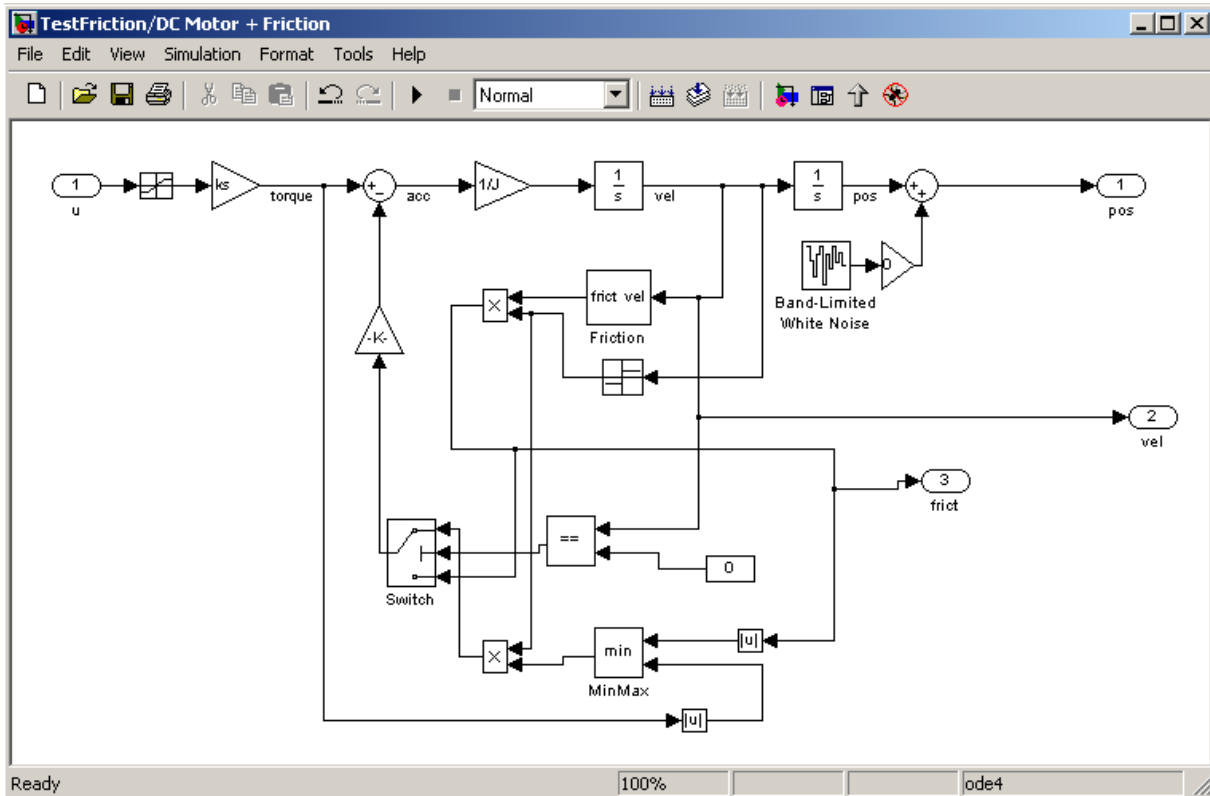
Schemat *TestFriction.mdl* zawiera liniowy model silnika sterowanego prądowo z uwzględnieniem statycznej charakterystyki tarcia.



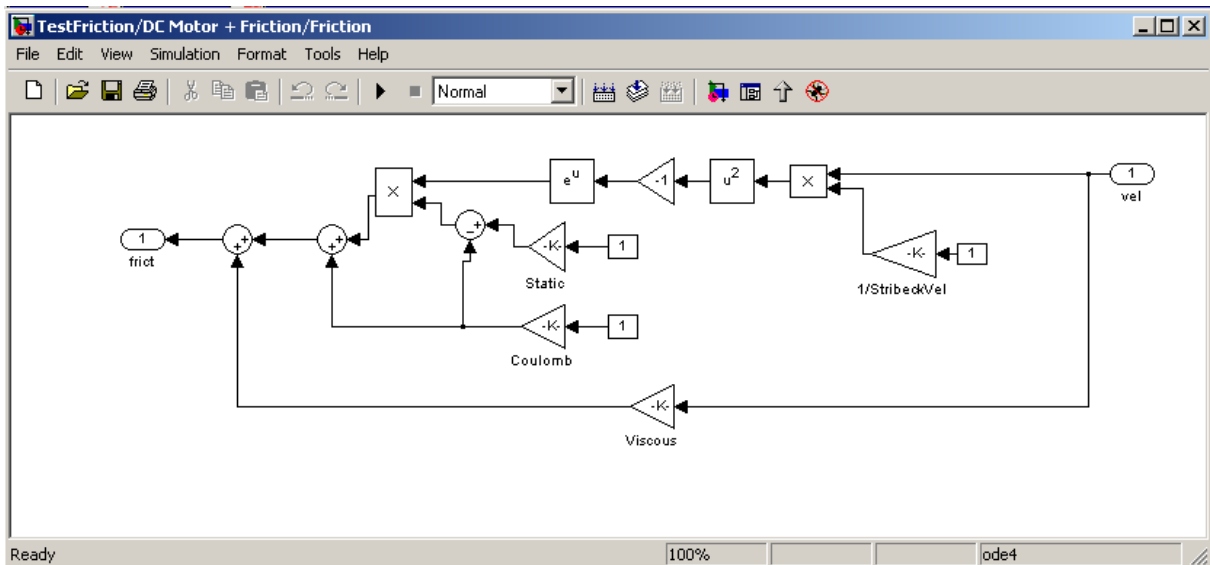
Rys. 14. Struktura modelu *TestFriction.mdl*

Parameter	Value
ks	1
J	1
Coulomb	0.5
Static	3
Viscous	0.1
StribeckVel	0.1
Friction ON	<input checked="" type="checkbox"/>

Rys. 15. Parametry statycznej charakterystyki tarcia



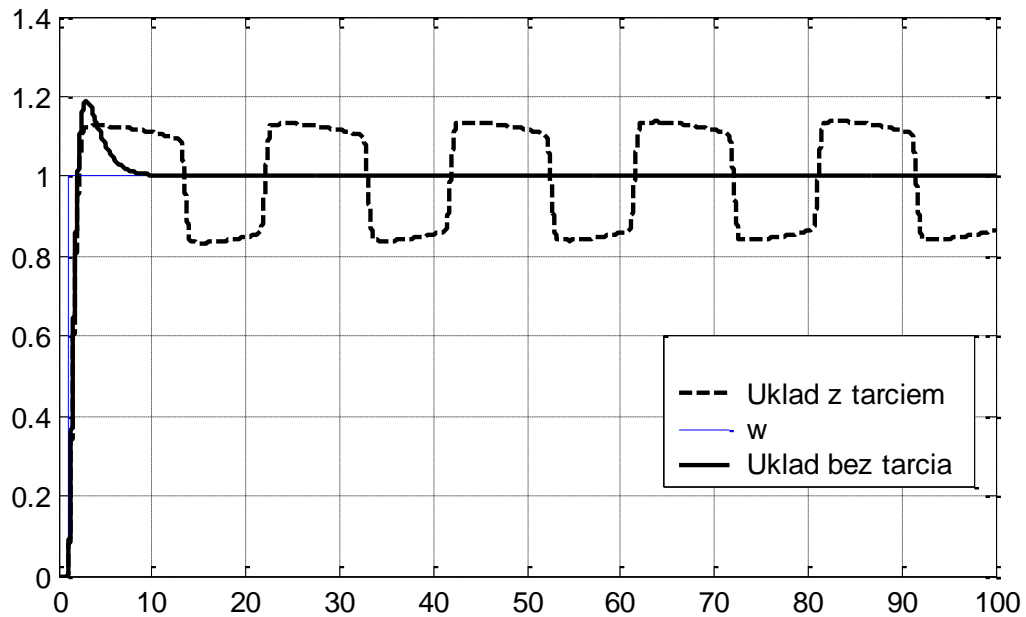
Rys. 16. Model silnika DC z statyczną charakterystyką tarcia



Rys. 17. Statyczna charakterystyka tarcia

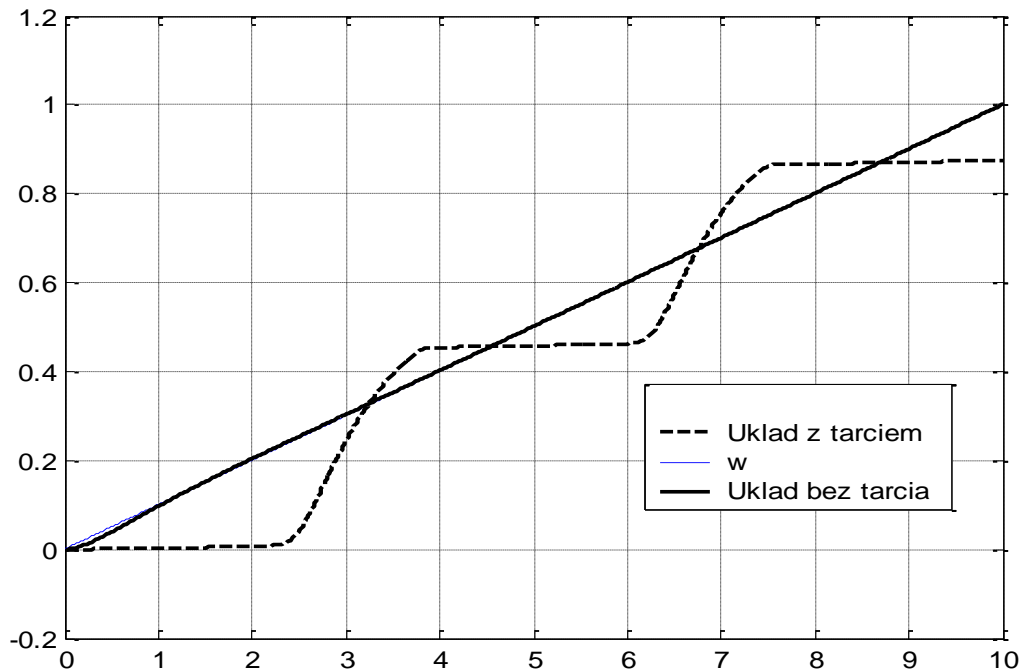
Przykładowe odpowiedzi układu z efektami wprowadzanymi przez tarcie:

- cykl graniczny (*limit cycle*) występuje w przypadku wymuszenia skokowego



Rys. 18. Odpowiedź (pos) układu z rys. 12 z uwzględnionym tarcie oraz bez tarcia

- drgania cierne (*stick-slip*) – występują gdy prędkość układu jest zbliżona do prędkości *Stribeck*



Rys. 19. Odpowiedź (pos) układu z rys. 12 dla śledzenia wymuszenia liniowego - z uwzględnionym tarcie oraz bez tarcia