LABORATORIUM 5: Sterowanie rzeczywistym serwomechanizmem z modułem przemieszczenia liniowego

Uwagi (pominąć, jeśli nie ma problemów z wykonywaniem ćwiczenia)

- 1. Jeśli pojawiają się błędy przy próbie symulacji:
- spróbuj przebudować schemat naciskając przycisk (informacje z budowy wyświetlają się w programie matlab):
- 🗅 🕼 🖬 🚭 🕹 ங 🖻 🗢 수 수 🏳 으 으 🕨 🔻 🦻 99999 🛛 External 🛛 🖃 🔛 🔛 🐄 🖼 📾 🛞 🤤
- jeśli to nie pomaga wyłącz i włącz ponownie cały pakiet **Matlab.**
- **2.** Jeśli wózek zatrzymał się w wyniku znalezienia się w zasięgu czujników krańcowych to nastąpiło wyłączenie modułu mocy. W celu ponownego przeprowadzenia kolejnej symulacji należy:
- nacisnąć zielony przycisk na obudowie serwomechanizmu \rightarrow nastąpi ponowne włączenie modułu mocy,
- ustawić w bloku **Step** w polu **Final value** odpowiednią wartość, żeby oddalić ciężarek od czujnika.

<u>Czynności wstępne – inicjalizacja stanów karty sterującej oraz pierwsza</u> <u>symulacja serwomechznizmu</u>

Jeśli wózek znajdzie się poza czujnikami krańcowymi to należy niezwłocznie wyłączyć zasilanie i poprosić o pomoc prowadzącego.

- 1. Utwórz nowy folder na pulpicie, w którym będą umieszczane kolejne schematy uruchamiane w trakcie zajęć (po zakończeniu zajęć zajęć należy go usunąć).
- Ściągnij ze strony z instrukcjami plik ze schematem do przeprowadzenia identyfikacji: Ident.mdl oraz plik do przenoszenia danych z pakietu Simulink do programu Matlab: SaveData.m → kliknij prawym przyciskiem myszy na link do pobrania każdego z plików, dalej wybierz *zapisz jako...*
- 3. Uruchom plik Ident.mdl (może to chwilę potrwać) → ukaże się schemat, który posłuży do przeprowadzenia identyfikacji tzn. do znalezienia wzmocnienia serwomechanizmu i określenia typu sterowania.
- 4. Przed uruchomieniem zasilania należy przeprowadzić symulację schematu w celu zainicjalizowania stanów karty sterującej → naciśnij przycisk Connect To Target:

🗋 😂 🖬 🚭 🕹 🕹 📾 💼 😓 🔶 수 🏻 으 오 🕞 🖉 99999 🛛 External 💿 🗖 🔛 🛗 🖓 🍪 🛗 🖡 🔯 📠 🛞

następnie Start real-time code (po udostępnieniu):

🗅 | 😅 🖬 🚳 | 🕹 🖶 | 수 수 수 | 으 으 | 🕨 🗑 19999 - External 🛛 🖃 🔛 🐼 🍩 🔛 | 🚂 🔤 🛞 🤇

5. Zatrzymaj symulację: naciśnij Stop real-time, który pojawił się po uruchomieniu schematu

w miejscu przycisku **Start real-time code (**jeśli przed zatrzymaniem symulacji zostanie <u>naciśnięty przycisk Connect To Target grozi to błędem i koniecznością zrestartowania</u> <u>całego pakietu Matlab</u>**).**

- 6. Teraz dopiero możesz włączyć zasilanie urządzenia: naciśnij czarny przycisk na obudowie.
- **7.** Po włączeniu zasilania powinna świecić się czerwona dioda → naciśnij zielony przycisk → serwomechanizm powinien być gotowy do przeprowadzenia symulacji.
- 8. Przed przeprowadzeniem pierwszej symulacji należy na schemacie ustawić:
- w bloku **Step** pole **Final value** czyli wartość skoku podanego na wejście serwomechanizmu:
 - → kliknij dwa razy na blok Step,
 - → wartość dodatnia przesuwa wózek w prawo, ujemna w lewo,
 - → do zapisania liczby zmiennoprzecinkowej należy używać kropki,
 - → możesz użyć wartości np. 0.3,
 - → jeśli ciężarek jest blisko prawej strony tu ustaw wartość ujemną,
- w bloku **Switch** pole **Treshold**, czyli czas przez jaki sygnał skoku będzie podany na wejście serwomechanizmu:
 - \rightarrow kliknij dwa razy na blok Switch,
 - → ustaw wartość np. 0.4.
- **9.** Kliknij jeszcze dwa razy na blok **Scope**, aby widzieć przebiegi z eksperymentu → najlepiej ustaw okno ze schematem oraz z wykresami obok siebie. Na wykresach widać przebiegi do sygnałów:
 - → **u** przebieg czasowy sygnału wejściowego,
 - → **vel** prędkość wózka,
 - → **pos** pozycja wózka,
 - → **voltage** poziom napięcia zasilacza [V]
 - → **current** prąd płynący w obwodzie silnika [A],
- **10. Przeprowadź symulację:** nacisnij przyciski **Connect To Target** oraz **Start real-time code** → po upływie czasu większym niż pole **Treshold** w bloku **Switch** naciśnij przycisk **Stop real-time**.
- **11.** Ciężarek powinien się poruszyć w zależności od ustawienia pola **Final value** w bloku **Step.**
- **12.** Dla lepszego pozycjonowania wykresów w oknie **Scope** naciśnij **czarną lornetkę** w pasku narzędzi w tym oknie.
- **13.** Przed kontynuowaniem dalszej części ćwiczenia rozgrzej serwomechanizm do punktu pracy wykonując kolejne symulacje dla różnych wartości w bloku **Step** (ok. 5 symulacji).

Identyfikacja serwomechanizmu

Na podstawie kształtu odpowiedzi skokowych układu, określ typ sterowania (prądowy, napięciowy) serwomechanizmu oraz wyznacz parametry transmitancji obiektu:

- 1. Dla schematu Ident.mdl ustaw:
 - → wartość bloku step na 0.2 (**Final value** w **Step)**,
 - → czas sygnału wejściowego na 0.2 (**Treshold** w **Switch**).

- **2.** Przeprowadź symulację i zarejestruj przebiegi. Na podstawie kształtów przebiegów czasowych położenia, prędkości, prądu, napięcia określ typ sterowania....
- 3. Oblicz wzmocnienie serwomechanizmu:

→ w programie Matlab ustaw bieżącą ścieżkę na folder wcześniej utworzony, gdzie znajduje się **Ident.mdl** oraz **SaveDateIdent.m**.

- **4.** Uruchom plik **SaveDataIdent.m** → dwukrotnie kliknij, a następnie kliknij na zielony przycisk w pasu narzędzi uruchomionego programu → nastąpi przeniesienie wykresów z **Simulink** do pakietu **Matlab**.
- 5. Znajdź wykres, który przedstawia wyłącznie prędkość i pozycję serwomechanizmu:

→ do odczytywania współrzędnych punktów użyj **Data Cursor:**



 \rightarrow dla połowy zakresu, w którym wykres prędkości jest w przybliżeniu liniowy odczytaj współrzędne punktu **dla wykresu pozycji**:



 \rightarrow oblicz wzmocnienie dla współrzędnych powyższego punktu z wykresu pozycji (powinno mieścić się w przedziale [6,10]),

- → wskaż prowadzącemu obliczone wzmocnienie.
- 6. Zapisz screeny przebiegów do sprawozdania.

Sterowanie serwomechanizmem poprzez regulatory

- 1. Zapytaj prowadzącego o wartość czasu regulacji dla kolejnych eksperymentów (jeśli prowadzący nie poda wartości czasu regulacji przyjmij 0.8 s).
- **2.** Oblicz nastawy regulatora PID ze wzorów (uzyskane za pomocą metody linii pierwiastkowych):

$$k_p = \frac{216}{k \cdot t_r^2}, \quad k_i = \frac{432}{k \cdot t_r^3}, \quad k_d = \frac{27}{k \cdot t_r}, \quad \alpha = \frac{4}{t_r}$$

- **3.** Skonsultuj obliczone nastawy z prowadzącym.
- 4. Dla obliczonych nastaw wykonaj następujące czynności przed wykonaniem eksperymentu z

wykorzystaniem schematu z regulatorem PID:

→ ściągnij pliki PID.mdl oraz SaveDataPID.m do wcześniej utworzonego folderu,

 \rightarrow uruchom schemat **PID.mdl**,

 \rightarrow wartość z bloku step ma być podana na wejście regulatora, jeśli tak nie jest kliknij dwa razy na przełącznik:

 \rightarrow ustaw przełącznik tak, aby filtr wstępny był wyłączony,

 \rightarrow w bloku Step ustaw wartość na 2 cm, wartość ma być podana w metrach czyli należy ustawić 0.02,

→ w bloku regulatora PID wpisz wcześniej obliczone nastawy,

- 5. Przeprowadź symulację z regulatorem PID:
 - \rightarrow uruchom okno z przebiegami klikając dwukrotnie na blok **Scope**,

 \rightarrow w trakcie symulacji jedna z osób grupy powinna dokładnie obserwować ruch ciężarka (czy ruch jest płynny? czy ciężarek przesuwa się w dobrym kierunku? czy się nie cofa? i czy w ogóle się przesuwa?),

→ wciśnij Connect To Targen oraz Start real-time code,

- \rightarrow symulację należy zatrzymać po ok. 7 s (Stop real-time).
- 6. Przenieś otrzymane przebiegi z programu Simulink do pakietu Matlab poprzez plik SaveDataPID.mdl.
- 7. Przeanalizuj wykres z wartością zadaną w oraz z odpowiedzą układu, czyli wykresem pozycji:

→ sprawdź czy występuje przeregulowanie,

→ jeśli tak to oblicz przeregulowanie bezwzględne w sprawozdaniu z ćwiczenia (użyj Data Cursor do zrobienia screena).

- 8. Powtórz eksperyment dla schematu z regulatorem PID z filtrem:
 - → ustaw przełącznik tak, aby filtr wstępny był włączony,
 - → wykonaj te same czynności co w przypadku symulacji układu bez filtra,
 - \rightarrow porównaj oba wykresy.
- **9.** Powtórz powyższe eksperymenty bez filtra wstępnego dla pozostałych regulatorów (ich schematy oraz pliki eksportujące wykresy z Simulink do pakietu Matlab należy pobrać ze strony z instrukcjami):

 \rightarrow wzory na nastawy PPI:

$$k_p = \frac{4}{t_r}, \quad k_{pi} = \frac{27}{k \cdot t_r}, \quad k_i = \frac{108}{k \cdot t_r^2}, \quad \alpha = \frac{4}{t_r}$$

 \rightarrow wzory na nastawy PD:

$$k_p = \frac{36}{k \cdot t_r^2}, \quad k_d = \frac{12}{k \cdot t_r}, \quad \alpha = \frac{3}{t_r}$$

W sprawozdaniu należy zamieścić

- 1. Screen z otrzymanych przebiegów z identyfikacji serwomechanizmu w programie Simulink.
- **2.** Screen dla wykresu pozycji i prędkości z programu Matlab z zaznaczonym punktem, na podstawie którego liczone jest wzmocnienie.
- 3. Obliczenia wzmocnienia serwomechanizmu.
- 4. Obliczenia nastaw poszczególnych regulatorów (PID, PPI, PD).
- **5.** Screen z przebiegami (<u>wystarczy wykres z pozycją i wartością zadaną</u>) dla poszczególnych regulatorów dla <u>wymuszenia skokowego</u> z filtrem oraz bez filtra.
- 6. Obliczenia przeregulowania jeśli wystąpią.
- 7. Jeśli uda się zdążyć to jw. ale dla <u>wymuszenia trapezoidalnego</u>.
- 8. Wnioski końcowe w tym:
- porównanie jakości regulacji poszczególnych regulatorów,
- efekt zastosowania filtru dla poszczególnych regulatorów (dlaczego dla PD i PPI nie jest potrzebne użycie filtra?),
- jeśli dany regulator nie nadaje się do sterowania serwomechanizmem, to podać przyczynę (związaną z transmitancją serwomechanizmu oraz brakującymi członami regulatora),
- inne wnioski, które nasuną się w trakcie wykonywania ćwiczenia...