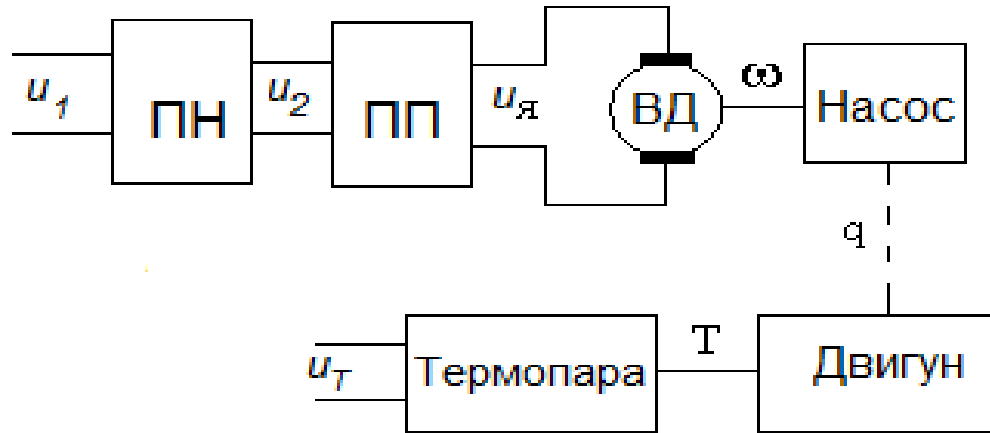


Алгоритм виконання курсової роботи

Етап 1. Побудова моделі об'єкта

Система автоматичного регулювання охолодження двигуна:



Для отримання передатної функції об'єкта необхідно перетворити кожне з диференціальних рівнянь елементів системи в операторну форму за нульових початкових умов. Підсилювачі:

$$u_2 = k_u \cdot u_1; \quad u_{я} = k_m \cdot u_2.$$

Здійснюємо перетворення Лапласа для першого підсилювача:

$$u_2(s) = k_u u_1(s);$$

Передатна функція:

$$W_1 = \frac{u_2(s)}{u_1(s)} = k_u.$$

Аналогічно для підсилювача потужності:

$$u_{я}(s) = k_m u_2(s);$$
$$W_2 = \frac{u_{я}(s)}{u_2(s)} = k_m.$$

Виконавчий двигун постійного струму ВД:

$$T_{ДВ} \frac{d\omega}{dt} + \omega = k_{ДВ} \cdot u_{я}$$
$$T_{ДВ} s \omega(s) + \omega(s) = k_{ДВ} u_{я}(s);$$
$$W_3 = \frac{\omega(s)}{u_{я}(s)} = \frac{k_{ДВ}}{T_{ДВ} s + 1}.$$

Насос :

$$T_h \frac{dq}{dt} + q = k_h \omega$$
$$T_h s q(s) + q(s) = k_h \omega(s);$$
$$W_4 = \frac{q(s)}{\omega(s)} = \frac{k_h}{T_h s + 1}.$$

Охолоджуюча сорочка двигуна:

$$T_P \frac{dT}{dt} + T = k_p q$$

$$T_p s T(s) + T(s) = k_p q(s);$$

$$W_5 = \frac{T(s)}{q(s)} = \frac{k_p}{T_p s + 1}.$$

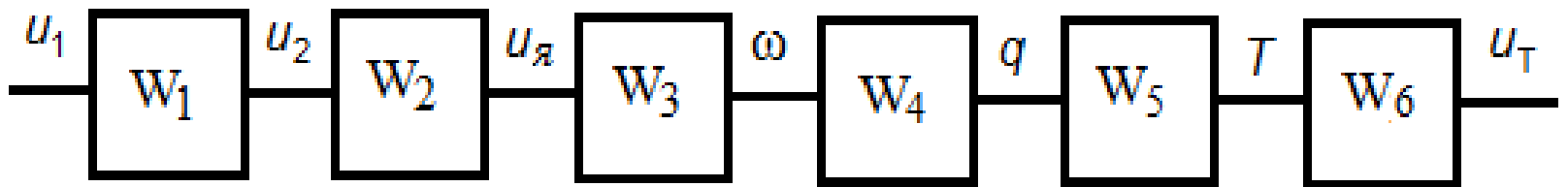
Термопара:

$$T_{tp} \frac{du_T}{dt} + u_T = k_{tp} T$$

$$T_{tp} s u_T(s) + u_T(s) = k_{tp} T(s);$$

$$W_6 = \frac{u_T(s)}{T(s)} = \frac{k_{tp}}{T_{tp} s + 1}.$$

ПФ об'єкта управління є послідовним з'єднанням шести ланок:



ПФ об'єкта в цілому отримаємо, використовуючи команди Matlab:

```
ku=30; km=2; kd=6; Td=0.4; kh=0.001; Th=1.2; kp=3; Tp=7; ktp=0.7;
```

```
Ttp=2;
```

```
W1=ku
```

```
W2=km
```

```
W3=tf(kd,[Td 1])
```

```
W4=tf(kh,[Th 1])
```

```
W5=tf(kp,[Tp 1])
```

```
W6=tf(ktp,[Ttp 1])
```

```
>> Wo=W1*W2*W3*W4*W5*W6
```

Wo =

0.756

6.72 s⁴ + 26.72 s³ + 28.88 s² + 10.6 s + 1

Таким чином, маємо динамічну систему 4-го порядку. Її полюси:

```
>> P=pole(Wo)'
```

```
P =  
-2.5000 -0.8333 -0.5000 -0.1429
```

Етап 2. Побудова графічного інтерфейсу

Скачуємо зразок графічного інтерфейсу у директорію Матлабу:

Мои документы\MATLAB

Посилання є у завданні на 6-й тиждень:

<http://citm.ho.ua/Dist/Disc/Source.rar>

Клацаємо по скачаному файлі правою клавішею миші і вибираємо опцію «Извлечь в Source\».

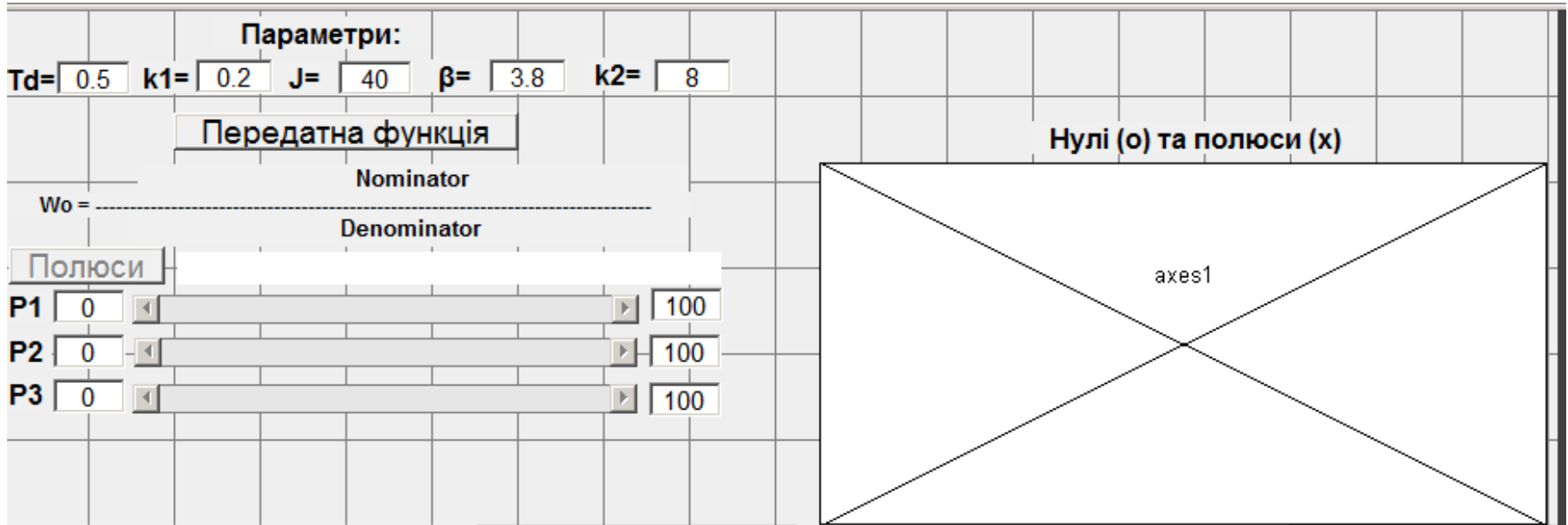
у вікні Матлабу Current Folder переходимо в новостворену директорію Source. Виконуємо команду:

```
>> guide Anten
```

Відкривається вікно графічного інтерфейсу. Вибираємо у меню File→Save As і зберігаємо під своїм ім'ям, наприклад, Cooling.

Коригуємо графічний інтерфейс під свій варіант.

По-перше, треба змінити вхідні параметри. якщо для цього варіанту їх більше, треба звільнити під них місце. Пересуваємо мишею нижче вікно axes1 і заголовок до нього.



Двічі клацаємо по заголовку $Td=$. Відкривається вікно Inspector. У рядку String змінюємо текст на ім'я першого параметру $k_u=$. Повертаємось у вікно cooling.fig. Двічі клацаємо по віконцю поруч зі скорегованим заголовком. У вікні Inspector у рядку String записуємо значення коефіцієнта k_u . У рядку Tag змінюємо ім'я edtTd на edtku.

Аналогічно коригуємо інші наявні параметри.

Додати нові параметри можна за допомогою копіювання. Виділяємо мишею групу елементів вікна `cooling.fig`, копіюємо комбінацією `Ctrl+C`, указуємо мишею, де хочемо розмістити копію, і виконуємо `Ctrl+V`.

По-друге, треба скорегувати функцію розрахунку передатної функції об'єкта.

У вікні `cooling.fig` клацаємо правою клавішею миші по кнопці «Передатна функція» і у контекстному меню вибираємо опції `View Callbacks`→`Callback`. У вікні редактора Editor відкривається функція `function btnTF_Callback`.

Після коментаря

% Зчитування параметрів ПФ об'єкта

ідуть оператори зчитування значень, заданих у віконечках введення параметрів. Їх треба скорегувати відповідно до нових тегів.

Наприклад, оператор

```
Td= str2double(get(handles.edtTd,'String'));
```

змінюємо на

```
ku= str2double(get(handles.edtku,'String'));
```

Такі оператори повинні бути для всіх вхідних даних.

Після зчитування даних іде розрахунок передатних функцій ланок і об'єкта в цілому. Його можна скопіювати з раніше зробленого розрахунку і вставити у функцію.

По-третє, треба зробити так, щоб кількість слайдерів під кнопкою «Полюси» дорівнювала порядку системи. Виділяємо один рядок, що включає заголовок **P3** та всі інші елементи праворуч, копіюємо і вставляємо нижче. Змінюємо заголовок на **P4** (або **P5**, **P6**), Для інших елементів праворуч змінюємо теги подібно до тегів у верхніх рядках: edtMinP4, sldP4, edtMaxP4.

У вікні `cooling.fig` клацаємо правою клавiшею миші по будь-якому з слайдерів P1-P3 і у контекстному меню вибираємо опції `View Callbacks`→`Callback`. У вікні редактора `Editor` відкривається функція `function sldP3_Callback`. Копіюємо усі рядки, що йдуть після першого рядка `function sldP3_Callback`.

У вікні `cooling.fig` клацаємо правою клавiшею миші по кожному з нових слайдерів P4 (або P5, P6) і у контекстному меню вибираємо опції `View Callbacks`→`Callback`. У вікні редактора `Editor` відкривається відповідна функція, наприклад, `function sldP4_Callback`. Вставляємо усі скопійовані рядки після першого рядка `function sldP4_Callback`. Змінюємо усюди P3 на P4 (або P5, P6).

По-четверте, треба скоректувати функцію визначення полюсів передатної функції.

У вікні `cooling.fig` клацаємо правою клавішею миші по кнопці «Полюси» і у контекстному меню вибираємо опції `View Callbacks`→`Callback`. У вікні редактора Editor відкривається функція `function btnPole_Callback`.

Після коментаря

% Задіювання необхідних слайдерів

ідуть оператори активізації слайдерів. Треба додати оператори до введених нових слайдерів, наприклад:

```
set(handles.sldP4, 'Enable', 'on')
```

Після коментаря

% Установлення границь для полюсів

треба зробити те ж саме, наприклад:

```
set(handles.edtMinP4, 'String', num2str(P(4)*2-1));  
set(handles.edtMaxP4, 'String', num2str(P(4)*0.5));
```


По-п'яте, треба скоректувати функцію синтезу спостерігача.

У вікні `cooling.fig` клацаємо правою клавішею миші по кнопці «Синтез системи зі спостерігачем» і у контекстному меню вибираємо опції `View Callbacks`→`Callback`. У вікні редактора Editor відкривається функція `function btnEst_Callback`.

Відшукуємо рядок

```
Pe=[-1,-2,-3];
```

і коригуємо вектор `Pe` так, щоб кількість елементів дорівнювала порядку системи.

На цьому коригування інтерфейсу закінчується, його треба зберегти і можна запустити на виконання кнопкою  .