

Ćwiczenie JCC-R01. Dwustrefowy regulator histerezy

JĘZYK C I C++ W PROGRAMACH STEROWANIA LABORATORIUM SYSTEMÓW STEROWNIA PRZEMYSŁOWEGO I AUTOMATYKI BUDYNKÓW

KATEDRA AUTOMATYKI NAPĘDU I URZĄDZEŃ PRZEMYSŁOWYCH
WWW.KANUP.AGH.EDU.PL

AKADEMIA GÓRNICZO-HUTNICZA
WWW.AGH.EDU.PL

Temat: Dwustrefowy regulator histerezy

Narzędzia: Kompilator, linker, debugger

Wstęp

Obsługa konsoli

W ćwiczeniu potrzebne są funkcje, które umożliwiają sterowanie położeniem kursora, kasowanie zawartości wiersza danego wiersza i czyszczenie ekranu.

Realizacja powyższych funkcji wymaga wysłania do konsoli odpowiedniego ciągu znaków. Do wysyłania znaków należy tutaj używać standardowej funkcji:

```
int putchar( int c )
```

Prototyp tej funkcji znajduje się w pliku nagłówkowym conio.h. Do sterowania konsolą nie należy używać funkcji printf() lub operatora >> w języku C++, ponieważ wysyłają one znaki na standardowe wyjście stdout.

W celu ustawienia kursora w zadanej pozycji (x, y) na ekranie należy wysłać na konsolę sekwencję kodów następujących znaków:

- *Esc*
- =
- y+' ' (współrzędna y plus kod znaku spacji)
- x+' ' (współrzędna x plus kod znaku spacji)

W celu oczyszczenia ekranu należy wysłać na konsolę sekwencję kodów następujących znaków:

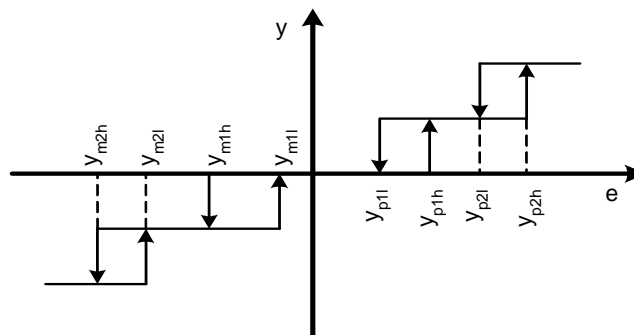
- *Esc*
- J

W celu oczyszczenia wiersza, w którym znajduje się kursor należy wysłać na konsolę sekwencję kodów następujących znaków:

- *Esc*
- K

Regulator histerezy

W ćwiczeniu wykorzystany będzie regulator histerezy dwukierunkowy i dwustrefowy. Regulator taki generuje sygnał wyjściowy przyjmujący kilka dyskretnych wartości (lub kilka sygnałów binarnych). Zależność sygnału wyjściowego takiego regulatora od wielkości błędu przedstawia rysunek 1.



Rys. 1. Charakterystyka regulatora histerezy dwukierunkowego, dwustrefowego

W ogólnym przypadku regulator taki może mieć ustawiane niezależnie wszystkie 8 progów przejść między stanami. Najczęściej regulatory tego typu mają progi ustawione symetrycznie względem zera, tj:

$$x_{p1h} = -x_{m1l}, x_{p1l} = -x_{m1h}, x_{p2h} = -x_{m2h}, x_{p2l} = -x_{m2l}$$

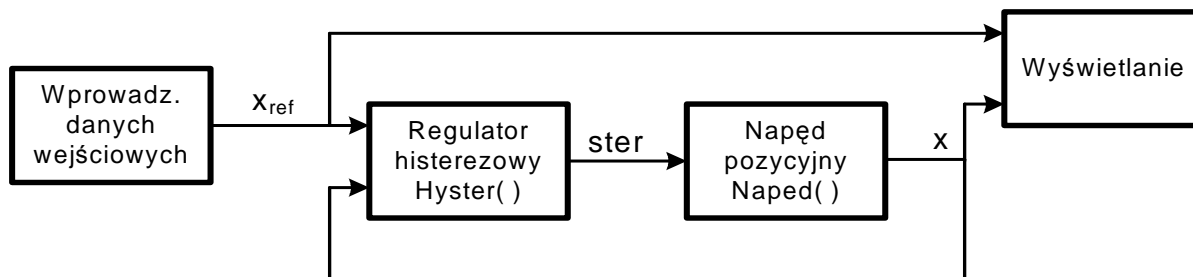
Regulator z symetrycznymi progami jest tematem niniejszego ćwiczenia.

Ćwiczenie JCC–R01. Dwustrefowy regulator histerezy

Program ćwiczenia

Celem ćwiczenia jest napisanie programu w języku C w środowisku systemu operacyjnego QNX, który realizuje funkcje dwustrefowego, dwukierunkowego regulatora histerezy z napędem pozycyjnym. Regulator histerezy ma pełnić funkcję regulatora położenia i tak sterować dwukierunkowym i dwubiegowym (szybko, wolno) napędem, aby ten osiągał zadane pozycje. Prędkość napędu ma zależeć od wielkości błędu położenia. W przypadku dużego błędu położenia napęd pracuje z dużą prędkością i po zmniejszeniu się błędu do pewnej wartości następuje przełączenie na małą prędkość. Po osiągnięciu zadanego położenia napęd zatrzymuje się.

Schemat blokowy układu będącego tematem ćwiczenia przedstawia rysunek 2.



Rys. 2. Schemat blokowy napędu

Na ekran konsoli mają być przedstawione wartości położenia zadanego i położenia rzeczywistego napędu w semigraficznej formie wykresu słupkowego (w poziomie) i w formie liczbowej. Program ma pozwalać na wprowadzanie wartości zadanej z klawiatury **bez zatrzymywania napędu** (w dowolnej chwili) oraz zakończenie programu po naciśnięciu klawisza *Esc*.

W celu realizacji zadania należy zdefiniować następujące funkcje, których prototypy są przedstawione poniżej:

- **void gotoxy(int ax, int ay);**
Funkcja ustawia kursor w pozycji (ax, ay) na ekranie konsoli.
- **void clrscreen(void);**
Funkcja czyści ekran konsoli i ustawia kursor w lewym górnym narożniku (współrzędne 0, 0)
- **void clrline(void);**
Funkcja kasuje wszystkie znaki w wierszu, w którym znajduje się kursor.
- **int Hyster(int xz, int x, int l1, int h1, int l2, int h2, int y1);**
Funkcja regulatora histerezy opisanego powyżej. Znaczenie argumentów funkcji jest następujące:
 - xz wartość zadana,
 - x wartość aktualna,
 - l1, h1 progi dolny i górny pierwszej histerezy,
 - l2, h2 progi dolny i górny drugiej histerezy,
 - y1 wartość wyjścia regulatora z poprzedniego kroku obliczeń (obiegu pętli programu).Funkcja zwraca wartości -2, -1, 0, 1 i 2 (szybko w tył, wolno w tył, stop, wolno w przód, szybko w przód)
- **int Naped(int ster, int krok, int kroks, int y1);**
Funkcja symulująca napęd pozycyjny. Znaczenie argumentów funkcji jest następujące:
 - ster sterowanie (wyjście z regulatora histerezy),
 - krok przyrost położenia dla prędkości małej (ster = -1 lub 1),
 - kroks przyrost położenia dla prędkości dużej (ster = -2 lub 2),
 - y1 położenie napędu z poprzedniego kroku obliczeń (obiegu pętli).Funkcja zwraca aktualne położenie (np. w zakresie 0–100).

W celu spowolnienia obiegu pętli programu można wykorzystać systemową funkcję `sleep()`, która wstrzymuje wykonywanie programu o zadaną liczbę sekund, albo `nanosleep()`, która pozwala wstrzymywać program na praktycznie dowolny.

Program należy napisać w dwóch wersjach:

- jednowątkowej, w której czytanie danych wejściowych, obliczenia i wyświetlanie wyników będą wykonywane w pętli w jednym wątku,
- dwuwątkowej, w której jeden wątek będzie służył do czytania danych wejściowych, a drugi wykonywał w pętli obliczenia i wyświetlanie wyników.