

# Sumulacja Rakiety w programie SimStructure

pliki:

- simstrucuture.zip
- ardugeek\_rocket1.zip

## Opis działania kodu sterującego

Ten dokument opisuje działanie fragmentu kodu sterującego dla pojazdu (np. drona lub rakiety) z wykorzystaniem regulatorów PID w dwóch osiach: pionowej (Y) oraz kąta nachylenia (Angle).

### Parametry PID dla osi Y

Kod definiuje następujące zmienne sterujące ruchem w osi pionowej:

- `desired_y` - zadana pozycja w osi Y. Na początku ustawiona na 10, ale potem dynamicznie zmieniana:

`desired_y = 15 + 5 * t;` - pozycja zależy od czasu `t`.

- `error_y` - błąd położenia: `error_y = desired_y - p1.y;`
- `kp_y`, `kd_y`, `ki_y` - współczynniki regulatora PID:
  - `kp_y = 1` (proporcjonalny)
  - `kd_y = 1` (różniczkujący)
  - `ki_y = 0.3` (całkujący)
- `i_y` - całka błędów (integrator) używana w członie całkującym

### Sterowanie siłą ciągu (thrust)

Wyznaczana jest siła ciągu na podstawie błędów:

```
't1.thrust = t1.saturation * (kp_y * error_y - kd_y * p1.dy + ki_y * i_y);'
```

Następnie przeliczana jest wartość siły ciągu do jednostek siły w Newtonach:

```
'thrust = t1.thrust / 100000;'
```

### Regulacja kąta nachylenia

Sterowanie kątem odbywa się przy użyciu osobnego zestawu PID:

- a - żądany kąt:

```
[]a = pi - pi/180 * b1.heading + 10 * (key2(„1”) - key2(„2”));
```

```
Żądany kąt jest zależny od kierunku (heading) oraz naciśnięcia klawiszy.  
* 'kp_a, kd_a, ki_a' – współczynniki PID dla kąta  
* 'angle_err' – przeskalowany błąd kąta do celów wizualizacji:  
  'angle_err = 50 * a;'  
* 'i_a' – całka z kąta: 'i_a = a / 10;'  
* 't1.angle' – wynik działania regulatora:  
  't1.angle = kp_a * a - kd_a * b1.spin + ki_a * i_a;'  
* 't1_angle' – przeskalowany sygnał sterujący kątem do wizualizacji:  
  't1_angle = 50 * t1.angle;'
```

## Część wizualizacyjna

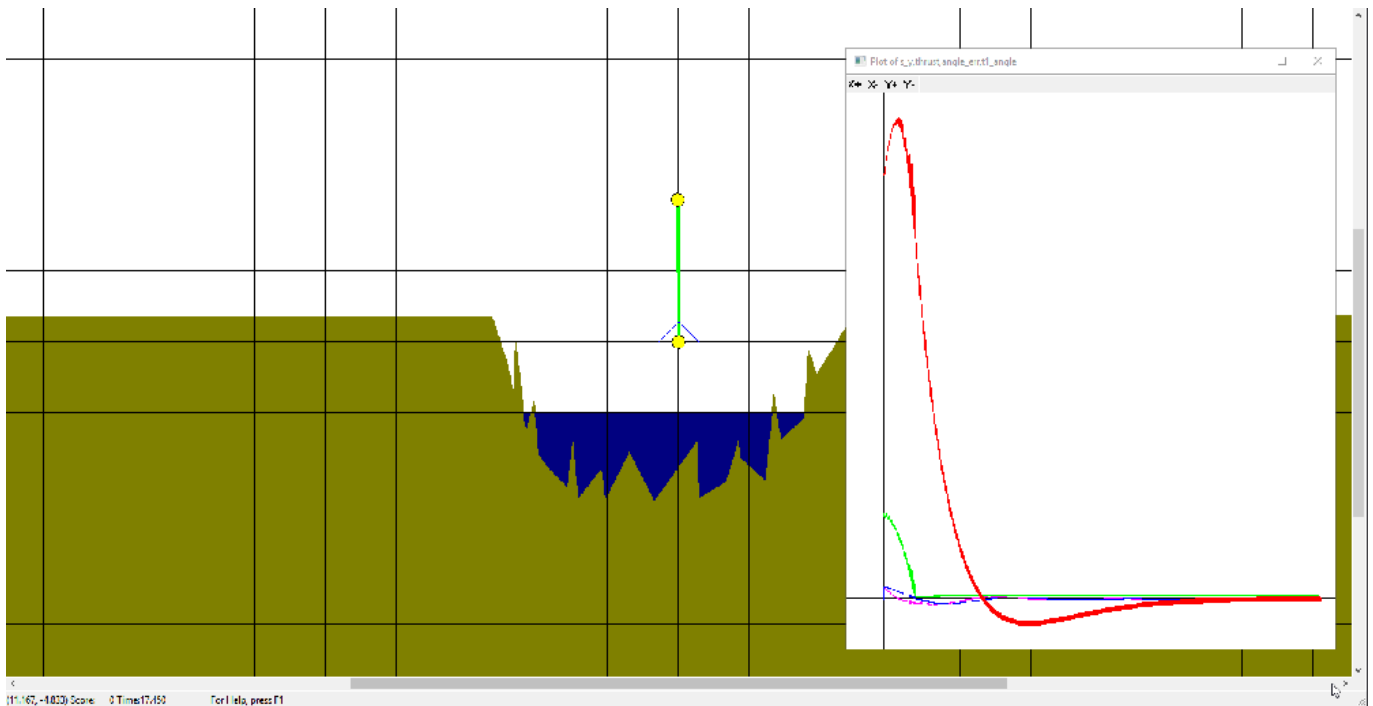
Na końcu kodu znajduje się blok graficzny służący do wyświetlania wyników:

- Rysowanie linii poziomej w pozycji `desired_y` (linia niebieska)
- Wyświetlanie tekstów:
  - Aktualna siła ciągu
  - Kierunek (heading)
  - Żądany kąt
  - Wartość całki z błędu w osi Y
- Wykresy (plot) dla:
  - `s_y` - błąd w osi Y
  - `thrust` - siła ciągu
  - `angle_err` - błąd kąta
  - `t1_angle` - sterowanie kątem

## Podsumowanie

Kod ten realizuje dwuwymiarowe sterowanie pojazdem:

- Stabilizacja położenia w osi Y z wykorzystaniem PID
- Sterowanie kątem nachylenia w celu orientacji
- Wizualizacja parametrów dla celów debugowania lub analizy



Kod:

```

double desired_y=10,error_y,kd_y=1,kp_y=1,ki_y=0.3;
signal s_y;
signal thrust;
signal t1_angle;
integrator i_y;

desired_y = 15+5*t;

error_y = desired_y-p1.y;
s_y = error_y;
i_y = error_y;
t1.thrust = t1.saturation*(kp_y*error_y-kd_y*p1.dy+ki_y*i_y);
thrust = t1.thrust/100000;
double a,kp_a=1.0,kd_a=0.01,ki_a=0.001,a_err;
[ ]a = pi-pi/180*b1.heading+10*(key2("1")-key2("2"));
signal angle_err;
angle_err = 50*a;
integrator i_a;
i_a = a/10;

t1.angle = kp_a*a-kd_a*b1.spin+ki_a*i_a;
t1_angle = 50*t1.angle;
@{
    line(p1.x-1000,desired_y,p1.x+1000,desired_y,blue,#3);
    text(#10,#50,"Thrust:  %12.6f N",t1.thrust);
    text(#10,#10,"Heading:%12.6f degree",b1.heading);
    text(#10,#30,"Angle:  %12.6f rad",a);
    text(#10,#70,"Integrator: %12.6f",i_y);
    plot s_y, thrust, angle_err, t1_angle;
}

```

}

From:

<https://wiki.ostrowski.net.pl/> - **Kacper's Wiki**

Permanent link:

<https://wiki.ostrowski.net.pl/doku.php?id=projekty:symulacjarakiety&rev=1746656323>

Last update: **2025/05/08 00:18**