

Sumulacja Rakiety w programie SimStructure

pliki:

- simstrucuture.zip
- ardugeek_rocket1.zip

Opis działania kodu sterującego

Ten dokument opisuje fragment 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. Początkowo ustawiona na 10, następnie dynamicznie zmieniana:

```
<code>desired_y = 15 + 5 * t;</code>
```

- **error_y** - błąd położenia:

```
<code>error_y = desired_y - p1.y;</code>
```

- Współczynniki regulatora PID:

- *Proporcjonalny: `kp_y = 1` Różniczkujący: `kd_y = 1` Całkujący: `ki_y = 0.3` - `i_y` - wartość całki błędu (integrator) ## Sterowanie siłą ciągu (thrust) Siła ciągu obliczana jest jako: `t1.thrust = t1.saturation * (kp_y * error_y - kd_y * p1.dy + ki_y * i_y); thrust = t1.thrust / 100000;` - `t1.saturation` to limit sygnału wyjściowego silnika. - Wynik dzielony jest przez 100 000, aby uzyskać wartość w [N]. ## Regulacja kąta nachylenia Sterowanie kątem realizowane jest osobnym regulatorem PID: - `a` - żądany kąt: `[a = pi - (pi/180) * b1.heading + 10 * (key2(„1”) - key2(„2”))];` - Współczynniki PID: Proporcjonalny: `kp_a = 1.0` Różniczkujący: `kd_a = 0.01` Całkujący: `ki_a = 0.001` - `angle_err` - przeskalowany błąd kąta: `angle_err = 50 * a;` - `i_a` - integrator kąta: `i_a = a / 10;` - Obliczenie sygnału sterującego kątem: `t1.angle = kp_a * a - kd_a * b1.spin + ki_a * i_a; t1_angle = 50 * t1.angle;` ## Część wizualizacyjna W bloku graficznym rysowane są: - Pozioma linia w pozycji `desired_y`**:

```
line(p1.x-1000, desired_y,  
p1.x+1000, desired_y,  
blue, #3)
```

- Dynamiczne teksty:

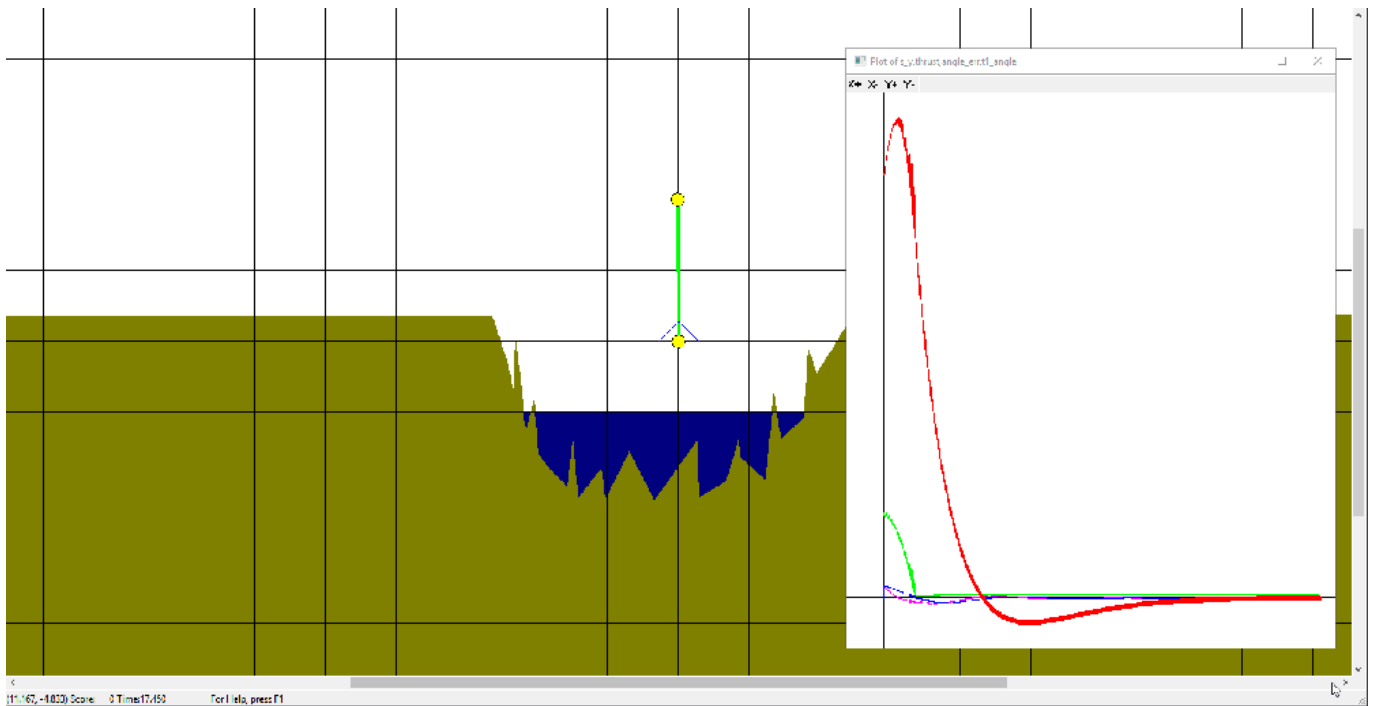
```
<code>text(#10,#50, "Thrust: %12.6f N", t1.thrust)
```

```
text(#10,#10, „Heading: %12.6f°”, b1.heading) text(#10,#30, „Angle: %12.6f rad”, a) text(#10,#70, „Integrator: %12.6f”, i_y)</code> - Wykresy sygnałów:
```

```
<code>plot s_y, thrust, angle_err, t1_angle</code>
```

Podsumowanie

- Sterowanie położeniem w osi Y realizowane jest przez regulator PID (P, I, D). - Sterowanie kątem nachylenia realizowane jest przez osobny regulator PID. - Blok wizualizacyjny umożliwia monitorowanie kluczowych parametrów w czasie rzeczywistym.



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;  
}
```