

# **VŘS – PŘISTÁVÁNÍ RAKETY V GRAVITAČNÍM POLI ZEMĚ**

**Tomáš Dvořák**

**A05051**

**[tdvorak@students.zcu.cz](mailto:tdvorak@students.zcu.cz)**

**23.8.2009**

## Zadání

### Přistávání rakety v gravitačním poli země

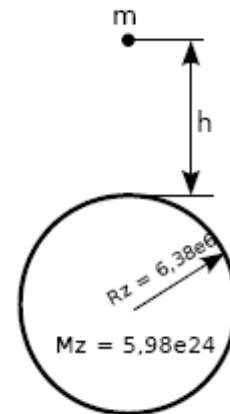
Gravitační síla působící na těleso o hmotnosti  $m$  ve výšce  $h$ :

$$F_g = \frac{\kappa m M_z}{(R_z + h)^2}$$

Poloměr země:  $R_z = 6,38 * 10^6 m$

Hmotnost země:  $M_z = 5,98 * 10^{24} kg$

Gravitační konstanta:  $\kappa = 6,67 * 10^{-11} N M^2 kg^{-2}$



### Úkoly:

- 1) Vizualizace modelu na PC
- 2) Ruční ovládání výkonu motorů
- 3) Vyzkoušejte použití regulátorů: reléový, P, PI, PID

*Ilustrace 1:  
Grafické znázornění  
konstant*

# Řešení

Pro popis pohybu rakety se použijeme 1. impulsový zákon

$$\frac{d}{dt}(p(t)) = F_c \quad \text{kde } [p(t)] - \text{hybnost tělesa v čase } t, [F_c] - \text{celková působící síla}$$

Hybnost tělesa je dána součinem hmotnosti a rychlosti, jež jsou obecně v čase proměnné

$$p(t) = m(t)v(t)$$

$$\frac{d}{dt}P = \frac{d}{dt}[m(t)v(t)] = m'(t)v(t) + m(t)v'(t)$$

Reaktivní síla motorů rakety se dá vyjádřit ze vzorce

$F_r = -W_{rel} * r$  kde  $W_{rel}$  je relativní rychlost spalín v m/s a  $r$  je reaktivita paliva, konstanta závislá na použitém palivu

$$F_p = -K \frac{M m(t)}{(Rz + h)^2} \quad \text{gravitační síla působící na těleso}$$

dále dosazením do vzorce 1. impulzového zákona

$$\frac{dp}{dt} = -F$$

$$m'(t)v(t) + m(t)v'(t) = -W_{rel} * r - K \frac{M m(t)}{(Rz + h)^2}$$

$$m'(t)v(t) + m(t)[v'(t) + K] = -W_{rel} * r \quad \text{kde } k \text{ je definováno jako } K = \kappa \frac{M}{(Rz + h)^2}$$

Obecné řešení diferenciální rovnice

$$m'(t)v(t) + m(t)v'(t) = 0$$

$$\frac{dv}{dt}m(t) = -m'(t)v(t)$$

$$\frac{dv}{v} = \frac{-m'(t)}{m(t)} dt$$

řešení rovnice pomocí metody variace konstant

$$\ln(v(t)) = -m'(t) \ln(m(t)) \frac{1}{m'(t)} + \ln C$$

$$\ln v(t) = \ln C m(t)^{-1}$$

$$v(t) = C m(t)^{-1}$$

$$\frac{dv}{dt} = C' m^{-1} C \frac{m'(t)}{m(t)^2}$$

$$C' = -F_r$$

$$C = \int -F_r$$

$$C = -F_r * t$$

Z předchozích vztahů lze vyjádřit dvě zásadní rovnice pro systém:

$$v'(t) = \frac{-F_r}{m(t)} + \frac{F_r * t * m'(t)}{m(t)^2}$$

$$m'(t) = -W_{rel} \frac{r}{v(t)} - \frac{m(t)}{v(t)} [v'(t) + K]$$

## Akční veličina

Jako akční veličinu použiji  $W_{rel}$ , ze vztahu pro reaktivní sílu motorů

$$F_p = -W_{rel} * r$$

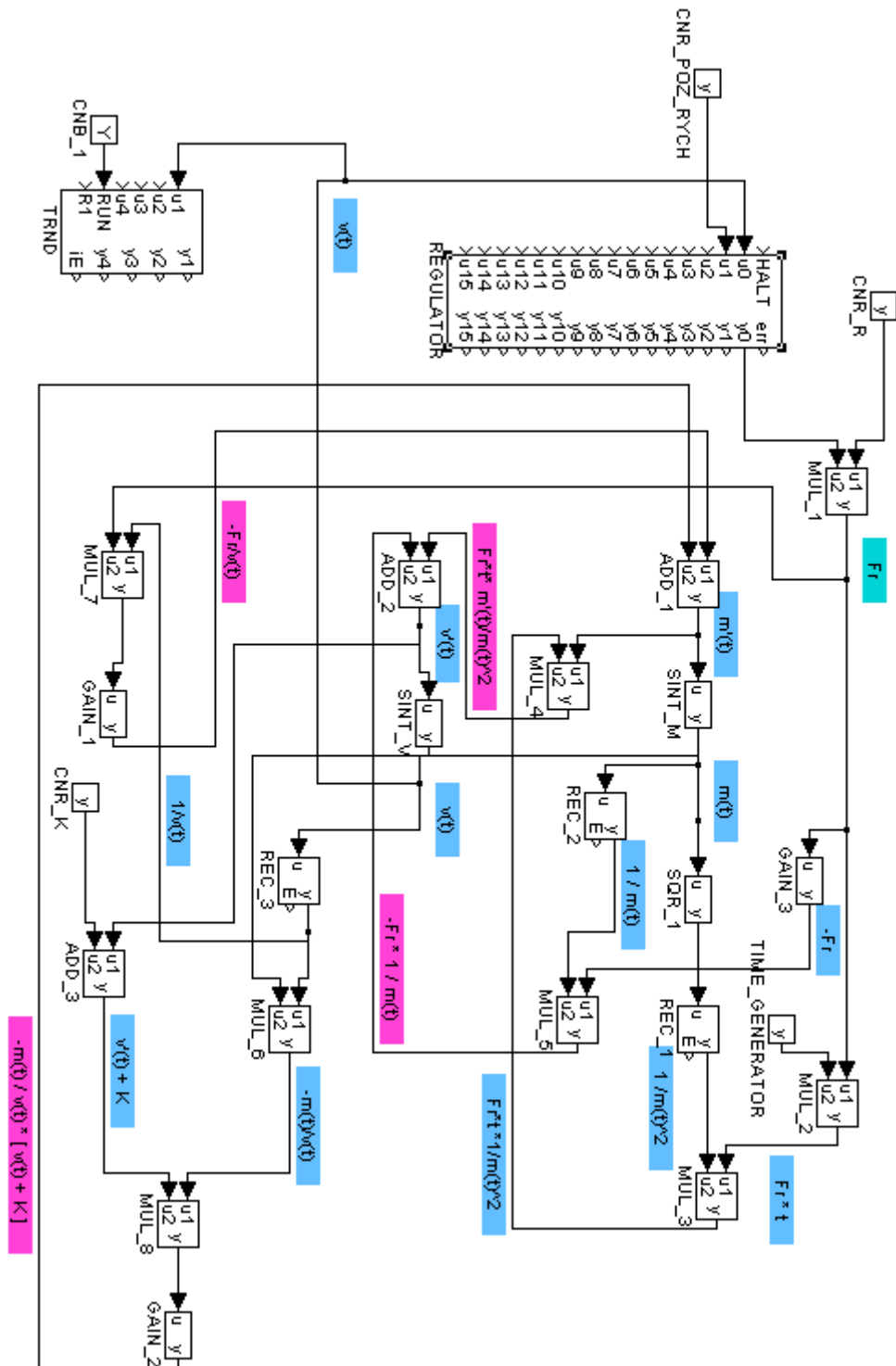
Kde  $W_{rel}$  je relativní rychlost spalin a  $r$  je konstanta závislá na použitém palivu.

Pomocí  $W_{rel}$  jsem tedy schopen regulovat rychlost přistávání rakety.

## Ruční regulace tahu motorů

Pokud bych chtěl sám regulovat tah motorů, namísto regulátoru bych do systému vložil jen spínač ovlivňující  $W_{rel}$ . Takto je systém autonomní a není možné do něj zasahovat manuálně.

# Model systému



Ilustrace 2: Model systému v REXu

# Návrh regulátorů

Fyzická implementace regulátoru je realizována pomocí funkčního bloku REXlang, který je standardním blokem prostředí REX. Tento blok se skládá ze vstupů které lze mapovat na výstupy. Dále je k dispozici možnost začlenit do bloku vlastní program ve formě zdrojového kódu založeného na jazyku C. Tímto jednoduchým spojením je dána možnost vytvářet bloky, které nejsou součástí prostředí REX. V mém konkrétním příkladě je realizován jediný vstup, aktuální rychlost rakety a jeden výstup akční veličiny Wrel. Regulátor je realizován ve formě reléového, P, PI a PID. Integrovaná složka je aproximována pomocí numerické metody lichoběžníkového pravidla a derivační složka pomocí diferencí (rozdíl reálné a chybové veličiny) Veškerá funkčnost je zřejmá ze zdrojových kódů.

<pre>/* ***** * releovy regulator pro regulaci rychlosti rakety * ***** */ double input(3) aktualni_rychlost; double input(4) pozadovana_rychlost; double output(0) Wrel; double parameter(0) Wmax; double parameter(1) Wmin; double e_t; int main(void) {     e_t = pozadovana_rychlost - aktualni_rychlost;     if (e_t &gt; 0) {         Wrel = Wmin     } else {         Wrel = Wmax     }     return 0; }</pre>	<pre>/* ***** * P regulator pro regulaci rychlosti rakety * * ***** */ double input(3) aktualni_rychlost; double input(4) pozadovana_rychlost; double output(0) Wrel; double parameter(0) P; double e_t; int main(void) {     e_t = pozadovana_rychlost - aktualni_rychlost;     Wrel += (P * e_t) ;     return 0; }</pre>
<pre>/* ***** * PI regulator pro regulaci rychlosti rakety * * ***** */ double input(3) aktualni_rychlost; double input(4) pozadovana_rychlost; double output(0) Wrel; double parameter(0) P; double parameter(1) I; double e_t; double sum_e = 0.0; int main(void) {     e_t = pozadovana_rychlost - aktualni_rychlost;     sum_e += e_t;     Wrel += (P * e_t) + (I * sum_e );     return 0; }</pre>	<pre>/* ***** * PID regulator pro regulaci rychlosti rakety* * ***** */ double input(3) cur_speed; double input(4) req_speed; double output(0) Wrel; double parameter(0) P; double parameter(1) I; double parameter(2) D; double e_t; double sum_e = 0.0; double last_speed = 0.0; int main(void) {     e_t = req_speed - cur_speed;     sum_e += e_t;     Wrel = (P * e_t) + (I * sum_e) - (D * (cur_speed - last_speed));     last_speed = cur_speed;     return 0; }</pre>