

Pružina – 1. Úkol z předmětu Modelování a simulace (A6M33MOS)

Cílem této části je získat praktický podklad pro tvorbu domácí úlohy. Budeme se zabývat pružinou.



Vzorec pro pružinu na obrázku by vypadal takto:

$$F_1 = -k \cdot x$$

Kde k je tuhost pružiny, x výchylka a F je síla.

Pokud dáme na pružinu závaží, lze ji popsat tímto vztahem

$$F_2 = m \cdot a$$

Kde m je hmotnost závaží, a zrychlení a F je síla. Pro systém spojení pružiny a závaží pak samozřejmě platí

$$F_2 = F_1$$

Konektory

Lehce se seznámíme s konektory. V elektrotechnické doméně jste je nepřímo poznali – např. vstup a výstup rezistoru, obsahoval flow a non-flow proměnnou, tedy proud a napětí. K dispozici máte přichystaný konektor pro pružinu – joint, který obsahuje proměnné **sílu** (flow) a **výchylku** (non-flow).

Přístup k proměnným v kódu:

- `joint.f` (síla)
- `joint.y` (výchylka)
- Např.: `joint1.y = y;`

Úkol na cvičení

- Vytvoříme si jednotlivé bločky, s jejichž pomocí budeme modelovat chování naší pružiny. Vytvořte si tedy tyto bloky (`model`), které budou řízeny těmito rovnicemi (`equation`).

model	Equation
fix	$y=0$
spring	$f_1=f_2;$ $f=-k \cdot dy;$ $dy=y_2-y_1$
mass	$F=m \cdot a;$ $a = \frac{d^2y}{dt^2}$

Pozn.: Nezapomeňte jednotlivé proměnné deklarovat.

- Vytvořte connector `joint` s dvěma proměnnými
 - `Real y;`
 - `flow Real f;`
- Napojte tyto proměnné na konektory
 - `Joint1.y = ...`
 - Zkontrolujte si správný název instancí (opravdu to spojujete se správným konektorem?)
- Všimněte si počáteční podmínky (initial equation) u bloku *mass*. Tím nastavujeme počáteční rozkmit. Pokud je 0, model bude statický.
- Propojte tyto bloky skrze konektory (povšimněte si, že součet sil = 0)
- Provedte simulaci chování
 - Pokud vám vyjde exponenciála, někde jste přehodili znaménko u sil. Stačí místo $-kx$ dát $+kx$.
- Kdo má hotovo může jít domů ☺

Domácí úkol - do 2. 10. 2012 23.59

Tato úloha je pro všechny povinná!

1. Zkuste si zapojit dvě pružiny do serie a paralelně.
2. Zkuste zapojit dvě závaží na jeden konektor. Jaký je s tím spojený problém? Jak byste ho řešili?
3. Vytvořte si nový blok, který pojmenujte třeba jako *source*. Bude stejný jako blok *fix* s tím rozdílem, že bude obsahovat `RealInput` (najdete v `modelka library – Modelica.blocks.interfaces`) a ten bude v sobě obsahovat zdroj sinusového signálu (`Modelica.Blocks.Sources.Sine`). Ten bude napojen na výchylku. Alternativně můžete přímo v kódu *source* přiřadit funkci `sine`.
4. Sestavte nucené kmitání jednoho oscilátoru. Dejte pozor, aby vám systém kmital i sám o sobě – nejprve vyzkoušejte `sin` s amplitudou 0. Nastavení `sinu` dvakrát klepejte na jeho ikonu v diagramu
5. Sestavte sériově dvě pružiny se závažím uprostřed (`fix – spring – mass – spring – mass`) – v podstatě kmitání s vlastní hmotností pružiny
6. Nuceně rozkmitajte na volném konci (`fix -- spring -- mass -- spring -- source`)
7. Nuceně rozkmitajte na pevném konci (`source -- spring -- mass -- spring -- mass`)
8. Popište pozorované změny kauzality a obecné vlastnosti systémů.

Bonus (1b)

1. Přidejte parametr tlumení
2. Přidejte parametr gravitace
3. Prozkoumejte rezonanci v závislosti na tlumení modelu z 2).

Bonus bonusu – tentokrát opravdu na bonus ☺

1. Implementace v OpenModelice a report případných bugů (++)0,5b)
2. Z minula: implementujte systém ventilace v simulinku a to jak pro zadané *P*, tak pro *Q*. Popište postup. (++)0,5b)