**Srdce jako pumpa**

**Milan Poláček**

# Zadání

Do řešení ze cvičení implementujete blok zdroje elasticity a blok srdečních intervalů.

Vyzkoušíte si:

* Tvorbu vlastního diskrétního konektoru
* Použití when
* Použití if
* Logicky přemýšlet

## Podotázky z instrukcí k řešení

* Jakého typu bude TPulse?
* Co vrací pre(TPulse)?
* Použijeme v bloku Heart Intervals if nebo when? Na co reaguje when? Jak bude vypadat podmínka? Dokážete popsat smysl operátorů pre()?
* Použijeme v implementaci vlastního bloku Rampa if nebo when?

## Simulace

* Pozorujte průběhy v Elastický kompartment (d), konkrétně elastanci. Čím je tento průběh omezen?
* Jak se mění proměnné ve zdrojích tlaku - Zdroj tlaku (a)?
* Jakým způsobem dochází k uzavírání a otevírání chlopní?
* Pozorujte vygenerovaný průtok a srovnejte tlaky a průtoky s realitou

# Řešení

Za pomocí instrukcí ze zadání jsem implementoval Heart elasticity (dále jen HE), který má nahrazovat elasticitu srdce. Implementace tohoto bloku zahrnovala jeden výstup a dva vstupy typu Real. Na vstupu jsem dal dle poskytnutého obrázku T0 (čas počátku systoly v sekundách) a Tsyst (délka systoly v sekundách). Na výstupu jsem dal Et. Dále jsem implementoval rovnice v podmínkách if podle instrukcí ze zadání. Blok HE jsem doplnil ještě od dva parametry konkrétně Systolickou a diastolickou elasticitu a přednastavil je dle zadání.

Poté jsem implementoval za pomocí instrukcí ze zadání blok Heart intervals (dále jen HI), který jsem dle doporučení ze zadání opatřil diskrétními výstupy (T0 a Tsyst) typu Real a vstupem HR (heart rate v úderech za minutu) také typu Real. Při deklaraci proměnných nám byla v instrukcích položena otázka, jakého typu bude proměnná TPulse, kdy jsem zvolil, že tato proměnná bude typu *discrete Real* (proč to je zdůvodněno dále v textu). Další otázkou, co bude v pre(TPulse). U řešení této otázky, ale není nutno modelovat, stačí využít znalost operátoru *pre*. Tento operátor nám zajišťuje rozlišení minulé a stávající hodnoty, ale má smysl jen u diskrétních proměných.

|  |
| --- |
| C:\Users\Milhouse\AppData\Local\Microsoft\Windows\INetCache\Content.Word\sys1.png |
| 1. Výsledný graf simulace závislosti pozice kuliček v čase dle 1. bodu zadání |

|  |
| --- |
|  |
| 1. Výsledný graf simulace závislosti pozice kuliček v čase dle 3. bodu zadání |

## Diskuze kauzality systému

Kauzalita systému, v grafu 1, nám názorně zobrazuje vzájemné působení závaží resp. kuliček. Kulička 1 je tedy vychýlena kuličkou 2 z rozvážné polohy. V důsledku toho kulička 1 ovlivňuje pozici kuličky 2 (a naopak).

## K jaké změně dochází při zapojení obou pružin?

U zapojení kuličky na jedné pružině (viz schéma 2) a při nastavení parametrů (k=2, m=3, výchozí výchylka 1m) je vidět z grafu 3, že systém je schopen oscilovat a výsledkem pozice kuličky v čase je tedy sínusovka (resp. cosínusovka). Při zapojení dvou pružin se dvěma závažími dochází k vzájemnému ovlivňování.

|  |
| --- |
| C:\Users\Milhouse\AppData\Local\Microsoft\Windows\INetCache\Content.Word\system_alone.png |
| 2. Schéma zapojení jedné kuličky na jedné pružině |

|  |
| --- |
| C:\Users\Milhouse\AppData\Local\Microsoft\Windows\INetCache\Content.Word\sysA.PNG |
| 1. Výsledný graf simulace závislosti pozice kuličky v čase |

## Popis napojení konektorů

Protože OpenModelica počítá veličiny jako toky a snaží se být, co nejblíže reálným předmětům (objektům) musí také přejímat základní fyzikální zákony. Zde konkrétně Zákon o zachování energie. A proto součet všech energií (vstupujících i vystupujících) v daném uzlu musí být roven nule. Z toho nám vyplývá, že některé veličiny musí být záporné a některé kladné.

# Závěr

V tomto domácím úkolu jsme si procvičili základní zacházení s programem OpenModelica. Zopakovali jsme si zde základy fyziky a rozšířili si znalosti o nový pohled na problematiku složitějších fyzikálních modelů (systémů).