

Tvorba modelů s Physiobrary – cirkulace a přenos krevních plynů

Úvod

Na tomto modelu si vyzkoušíme práci s rozsáhlou knihovnou, která nám umožní rychle poskládat složitý model. Zároveň si ukážeme základy regulace zpětné vazby a expandable konektory.

Úloha se zdá být roztáhlou, ale je jen detailním návodem jak přetahat pár bločků. Sestavování včetně kontroly parametrů by vám nemělo zabrat více než 30m. Pokud budete mít se sestavováním nějaký problém, diskutujte ho co nejdříve na fóru.

Z důvodu pozdního zadání a také, abyste si ještě nahrabali body, je termín odevzdání úlohy přesně za týden, tj sobota 16.1. do 23:59. **Bonus bod (+1b) pro každého**, kdo to ale stihne ve standardním termínu, tj. **úterý 12.1.**

Zadání

K tomuto zadání úlohy naleznete níže ke každému bodu detailnější informace.

1. Rozšiřte model cirkulace, který jsme dělali na cvičení o regulaci průtoku (Frank-Starlingův zákon). Ověřte a demonstруйте funkci (například nastavte počáteční hodnotu pulmonálních vén o 1l více a systémových vén o 1l více) ve srovnání s neregulovaným modelem.
2. Vytvořte model transportu CO₂ analogicky k modelu O₂, který jsme dělali na cvičení a rozšiřte model transportu CO₂ o regulaci dýchání.
Demonstруйте vliv regulace při cvičení (místo TissuesFlowRate použijte Step) a diskutujte výsledky.
3. Spojte všechny tři subsystemy do jednoho modelu pomocí BusConnectoru (expandable connector) a propojte je i uvnitř subsystemů. Alespoň jeden submodel s buskonektorem otestujte zvlášť .
4. **Bonus (+1b):** Rozšiřte si knihovnu děděním tak, abyste **zobrazili** jednotky toku na běžněji používané – ml/min a tlaku z Pa na mmHg. Zároveň bychom chtěli do takto modifikovaného modelu zadat parametry v běžnějších jednotkách (například právě toho toku v jednotkách ml/min a nikoli v m³/s). Popište zvolené řešení.
5. **Bonus (+1.5b):** Demonstруйте funkci celého modelu na 3 možných jevech (například cvičením zvýšíme spotřebu CO₂, šokový stav (pokles systémového odporu), omezení dechu etc.) dle vlastního výběru. Diskutujte platnost modelu pro každý jev.
6. **Bonus (+1b) za včasné odevzdání.**
7. **Bonus (+1b) za první report chyby v zadání na fóru¹.**

Physiobrary

Physiobrary je malá knihovna určená k vývoji modelů zejména fyziologických regulací. Její poslední verzi naleznete na <https://github.com/MarekMatejak/Physiobrary>. Prostředí Dymola umí využít její pokročilé funkce, jako třeba automatické přepočítávání jednotek. To bohužel OpenModelica neumí, ani nám nepomůže s Expandable konektory.

¹ Negarantuji, že je přítomna :)

Jednotky

Většina jednotek používaných ve Physiobrary je ze základní soustavy SI. K automatickému přepočtu používáme `displayUnits`, můžeme tak zadávat parametry přímo v cílových jednotkách a zároveň v nich sledovat průběhy. Pozor, počítá se ale stále v základních!!!

Například u konektoru `HydraulicPort` je uveden typ `Types.Pressure pressure`, který když si otevřeme (`Physiolibrary.Types.Pressure`), tak říká:

```
type Pressure = Modelica.SIunits.Pressure(displayUnit="mmHg", nominal=133.322387415);
```

a najdeme-li si i tu, tak:

```
type Pressure = Real ( final quantity="Pressure", final unit="Pa", displayUnit="bar");
```

Tedy jedná stále o typ `Real`, jen s nějakými dalšími vlastnostmi, které ale na samotný výpočet nemají vliv. Počítám stále v Pa, Dymola to ale umí zobrazovat i jako bar (pokud použijeme `Modelica.SIunits.Pressure`), respektive mmHg (pokud použijeme redefinovaný typ z Physiobrary). `OpenModelica` tento přepočet bohužel neumí a `displayUnit` i `Unit` si klidně ignoruje. Na samotný výpočet to vliv nemá, ten přeci probíhá v základních jednotkách.

Kromě přepočítávání je tento přístup vhodný i kvůli grafickým překlepům. Mám-li konektor `Real Output`, připojím ho na jakýkoli `Real input`. Ale konektor `Pressure Output` lze připojit zas a pouze na `Pressure Input`. Proto není dobré používat `Real` proměnné, ale vždy jen jednotky odvozené z těch základních. Balíček `Physiolibrary.Types.Examples.Units.ParametricClass` ukazuje jejich použití. K tomu je potom potřeba si dodefinovat některé základní konstanty – nemůžeme již používat normální `Real` k propojení těchto konektorů. Je to kvůli tomu, aby se nemátly přepočty jednotek.

Expandable konektor

Je konektor, který obsahuje právě tolik proměnných, kolik do něj zapojíme – rozšiřujeme ho tedy prostým propojením (a do něj nemusíme už nic dalšího psát, většinou je tedy prázdný. Podívejte se na `Physiolibrary.Types.BusConnector`). Všechny spojené expandable konektory potom slouží jako propojení těchto konektorů. Aby to bylo univerzálnější, je to propojené jménem, které si vybereme.

```
connect(busConnector.mu_j_nazev, arteries.volume)
      a třeba v úplně jiné části modelu pak najdeme
connect(arteries.solutionVolume, busConnector.mu_j_nazev)
```

Expandable konektory se pak mezi sebou propojují normálně jako každé jiné slušné konektory.

```
connect(o2MinimalBus.busConnector, cVSMinimalBus.busConnector)
```

V tom velmi pomůže prostředí, které si propojení „pamatuje“, jako například právě Dymola. Ta nám při napojení na `BusConnector` nabídne seznam již použitých napojení (případně až po kliknutí na `<AddVariable>`). To bohužel OM neumí, dokonce nás mate, že při připojení na konektor vygeneruje špatnou rovnici `connect`. Musíme ji proto manuálně upravit.

Počítání v ekvilibriu

Physiolibrary je teoreticky schopná počítat modely přímo v ustáleném stavu. Tuto vlastnost pro její složitost teď nebudeme využívat, jen aby vás nemátla celková komplexnost prvků knihovny.

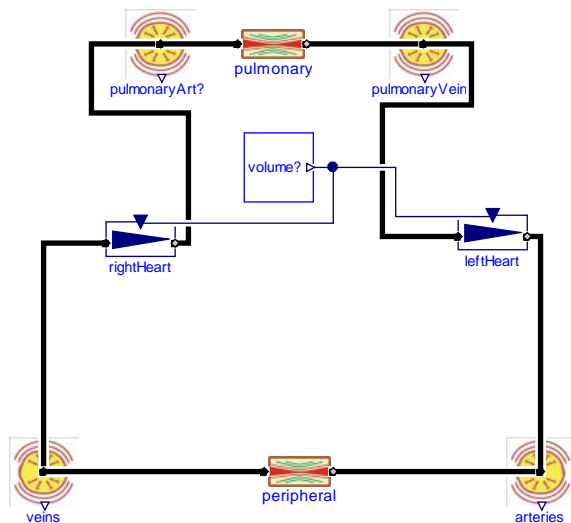
Subsystémy

Budeme pracovat s několika subsystémy. Bloky cirkulace a přenosu kyslíku jsme se zabývali na cvičení, v této úloze budeme vytvářet systém přenosu CO₂ a doplníme některé regulace.

Jednotlivé bloky si nejdřív sestavíme a otestujeme zvlášť, poté je teprv připojujeme na BusConnector.

Cirkulace

Jednoduchá cirkulace s pouze minutovými průtoky. Sledujte tlaky v jednotlivých částech řečiště a celkový minutový průtok.



Obrázek 1: Jednoduchá cirkulace

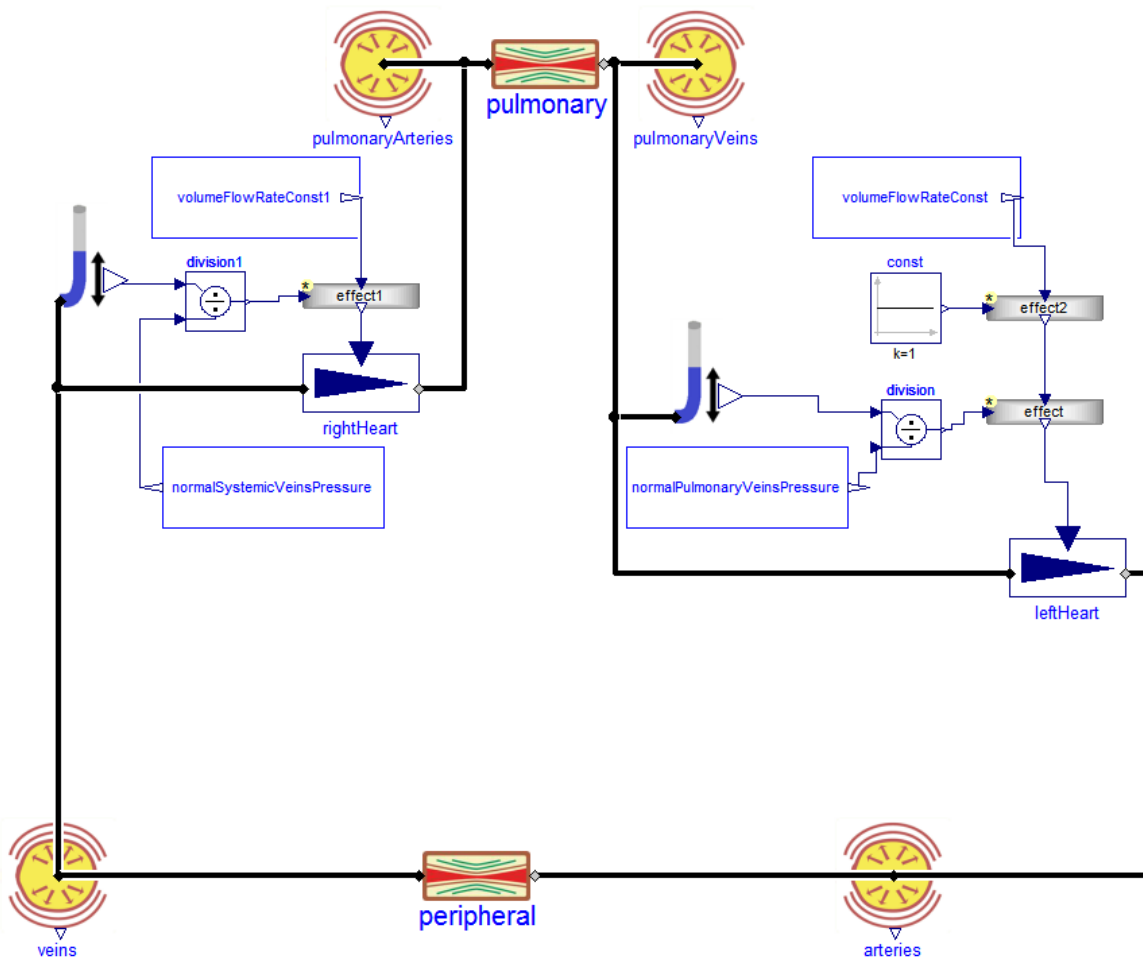
Tabulka hodnot

Hydraulic.ElasticBaloon	PulmonaryArteries	ZeroPressVol 0 ml, 6 ml/mmHg (5e-08 m ³ /Pa), volume_start 100ml
Hydraulic.ElasticBaloon	PulmonaryVeins	ZeroPressVol 0 ml, 80 ml/mmHg (6e-07 m ³ /Pa), volume_start 400ml
Hydraulic.ElasticBaloon	Arteries	ZeroPressVol 0 ml, 10 ml/mmHg (7.5e-08 m ³ /Pa), volume_start 1000ml
Hydraulic.ElasticBaloon	Veins	ZeroPressVol 0 ml, 1750 ml/mmHg (1.3e-05 m ³ /Pa), volume_start 3500ml
Hydraulic.Resistor	Pulmonary	558 ml/mmHg/min (6.9e-08 m ³ /Pa/s)
Hydraulic.Resistor	Peripheral	57 ml/mmHg/min (7.1e-09 m ³ /Pa/s)
Hydraulic.Pump	rightHeart	
Hydraulic.Pump	leftHeart	
Types.Constants. VolumeFlowRateConstant		5500 ml/min (9.1e-05 m ³ /s)

Regulace průtoku krve v oběhu

Duplikujte předchozí model (right-click – new – duplicate class) a upravte model cirkulace tak, aby se srdeční pumpa regulovala pomocí *preload* (tlaku na vstupu) a to jak u systémového, tak pulmonálního okruhu. Zároveň si připravíme další stupeň regulace v levém srdci pro pozdější využití.

Types.Constants.VolumeFlowRateConst	VolumeFlowRateConst1	5500 ml (? m ³)
Types.Constants.PressureConst	normalSystemicVeinsPressure	2 mmHg ² (? Pa)
Hydraulic.PressureMeasure	pulmPressure	
Blocks.Factors.Effect	effect	
Modelica.Blocks.Math.Division	Division	
<i>To samé i pro regulaci levého srdce, akorát jiný parametr a další stupeň:</i>		
Types.Constants.PressureConst	normalPulmonaryVeinsPressure	5 mmHg ³ (? Pa)
Blocks.Factors.Effect	effect2	
Modelica.Blocks.Sources.Constant	constant1	1 [-]

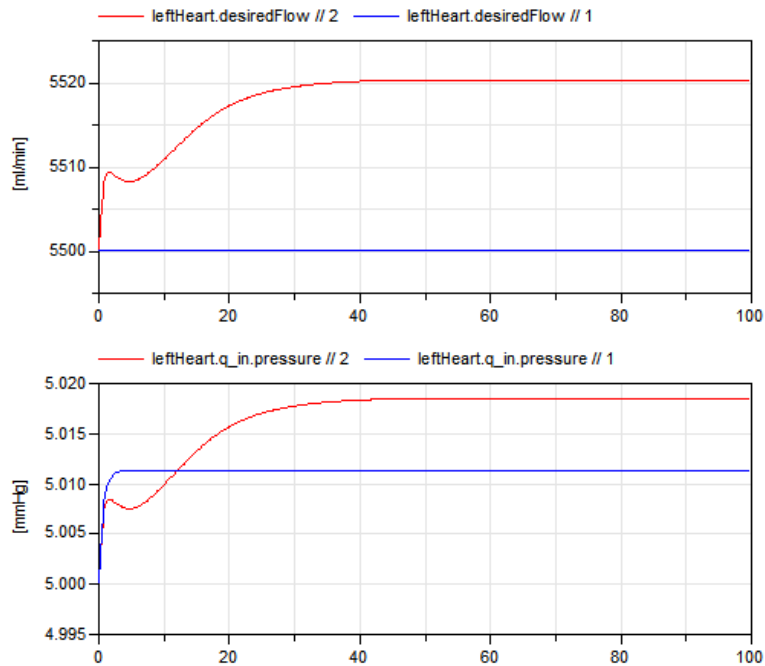


Obrázek 2: Model regulace srdečního výdeje. Výstup ze senzoru tlaku je podělen normálním tlakem (tj. mezivýsledek je 1) - pozor co s čím dělíte - výsledek je pak pronásoben tokem při normálním tlaku. Klesne-li tedy tlak, klesne i výsledek dělení a tím klesne i celkový průtok pumpy.

U levého srdce máme připraven další prvek pro možnost regulace. Můžeme je takto řadit v kaskádě.

² Tedy hodnota tlaku na vstupu pravého srdce předchozího neregulovaného modelu

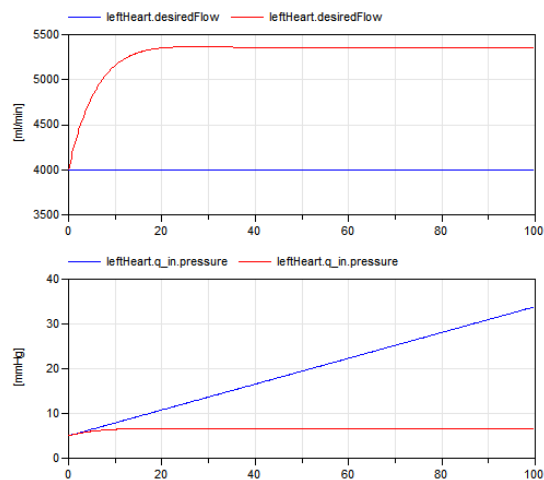
³ Tedy hodnota tlaku na vstupu levého srdce předchozího neregulovaného modelu



Obrázek 3: Obvod s regulací (červeně) vs obvod bez regulace. Rozdíl v ustálené hodnotě je dán mírně odlišnou referencí (hodnoty parametrů normalPulmonaryVeinsPressure a normalSystemicVeinsPressure oproti hodnotám v neregulovaném modelu).

Experiment

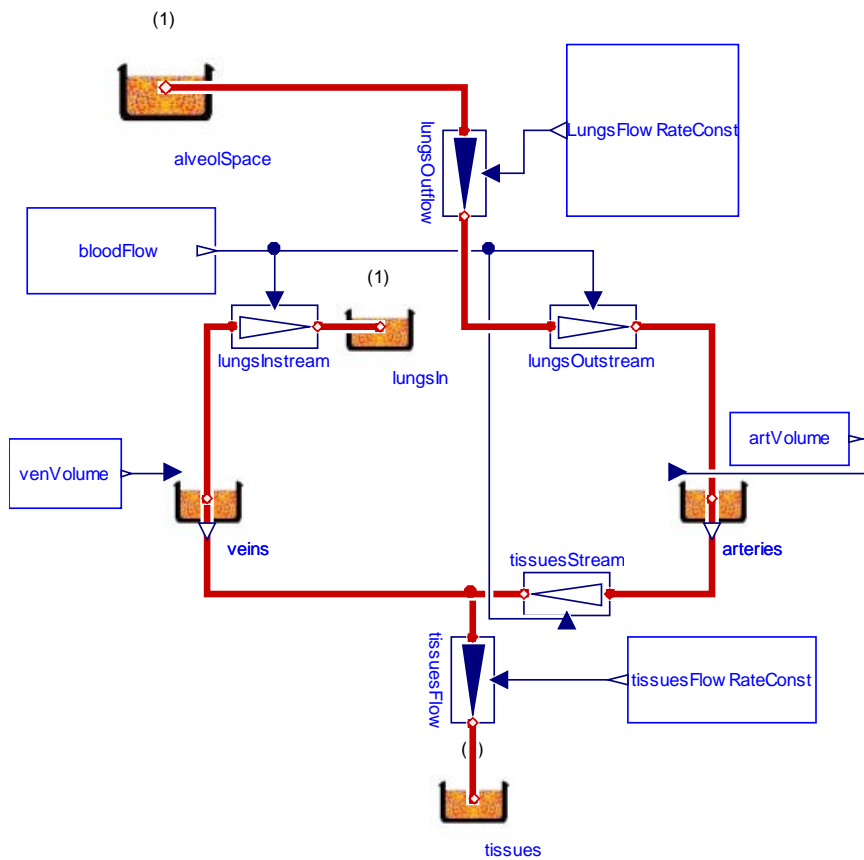
Podstatu a důvod regulace ozřejmíme v následující experimentu – snižte výkon (zde tedy parametr volumeFlorRateConst) **POUZE levého srdce** (například z důvodu infarktu) na 75 % a to jak v minimálním neregulovaném modelu (budete muset každému srdci přiřadit jednu konstantu, ne jednu společnou jako dosud), tak v regulovaném.



Obrázek 4: Výsledek experimentu. Zatímco v regulovaném obvodu stoupne tlak před levým srdcem, tím stimuluje tok a ten se normalizuje, v neregulovaném obvodu sice držíme průtok, ale tlak (a objem) lineárně roste.

02

Přenos kyslíku – podobný modelu, který jsme dělali minule. Zde si dávejte pozor na jednotky, jestli je zapisujete správně.

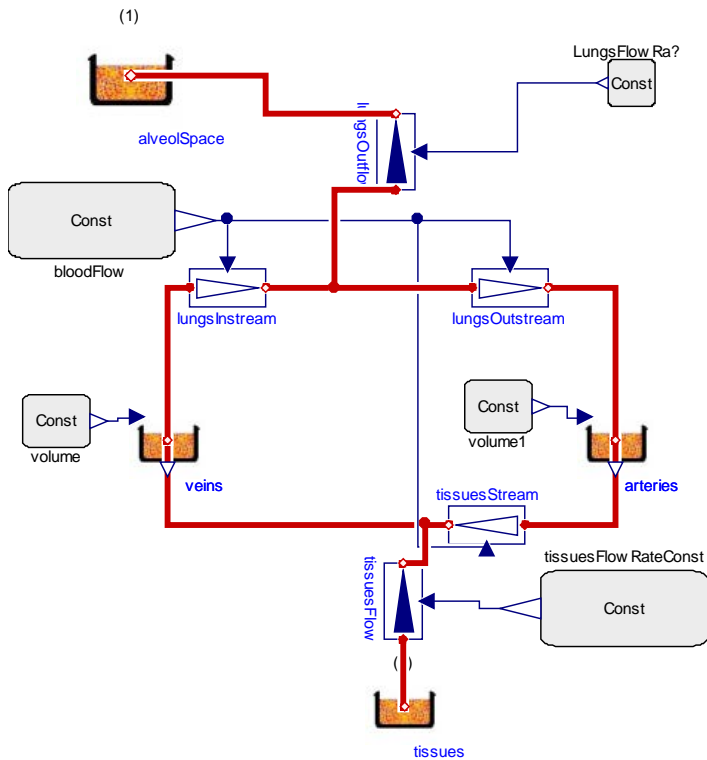


Chemical.UnlimitedStorage	alveolSpace	1 (je to fuk)
Chemical.UnlimitedStorage	lungsIn	1 (je to fuk)
Chemical.UnlimitedStorage	Tissues	1 (je to fuk)
Chemical.Substance	Arteries	0.02 mol
Chemical.Substance	Veins	0.02 mol
Chemical.MolarStream	LungStream	
Chemical.MolarStream	lungOutStream	
Chemical.MolarStream	tissuesStream	
Chemical.SoluteFlowPump	lungsOutFlow	
Chemical.SoluteFlowPump	tissuesFlow	
Types.Constants.VolumeFlowRateConst	bloodFlow	5.5 l/min (9.1e-05 m ³ /s)
Types.Constants.MolarFlowRateConst	LungsFlowRate	42 mmol/min (? mol/s)
Types.Constants.MolarFlowRateConst	TissuesFlowRate	18 mmol/min (0.0003 mol/s)
Types.Constants.VolumeConst	VenVolume	3.5 l (? m ³)
Types.Constants.VolumeConst	ArtVolume	1.5 l (? m ³)

CO₂

Sestavte model CO₂ podobně jako model O₂ s tím rozdílem, že plíce nám oxid uhličitý odebírají a tkáň naopak přidávají. Hodnoty vezměme stejné jako u kyslíku, protože musí probíhat výměna 1:1.

Zároveň si model jemně přiblížíme realitě – CO₂ se nebude na konci okruhu ztrácet, ale bude skutečně cirkulovat. Odebereme tedy blok lungsIn a zapojíme do kruhu. Množství přidaného a odebraného CO₂ v plicích a tkáních tedy **musí být stejný** (18 mmol/min).

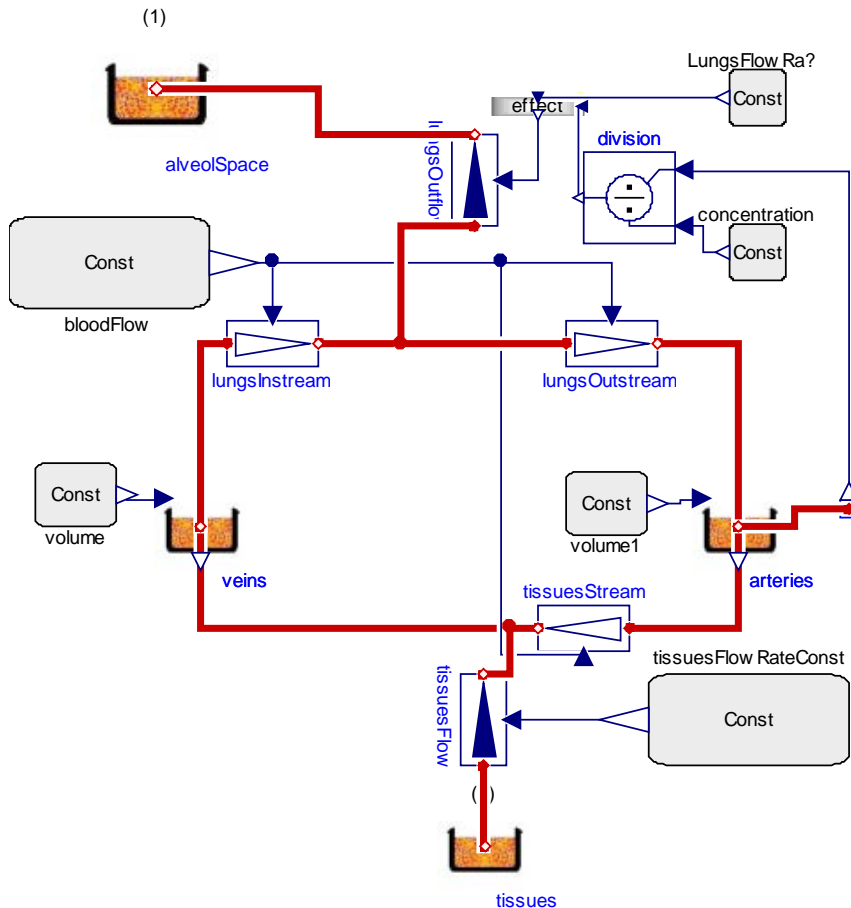


Obrázek 5: Cirkulace CO₂

Regulace koncentrace oxidu uhličitého

Když najednou začneme více cvičit, tak tkáň produkuje více CO₂. To se musí projevit i v práci plic (hlouběji dýcháme).

Vezmeme si tedy senzor koncentrace v arteriích a podobným způsobem jako s regulací tlaku budeme řídit vydýchávání CO₂.



Obrázek 6: Regulace cirkulace CO₂. Dejte si pozor na pořadí vstupů dělení (bloček je zrcadlený, nikoli narotovaný) – co se s čím dělí, jelikož plicní pumpa odčerpává CO₂, regulace spočívá v aktuální_koncentrace / chtěná koncentrace.

Blocks.Factors.Effect	Effect	
Types.constants.ConcentrationConst	normalConcentration	8.2 ⁴ mmol/l = mol/m ³
Types.constants.MolarFlowRateConst	normalFlowRate	18 mmol/min (0.0003 mol/s)
Chemical.ConcentrationMeasure	concentrationMeasure	
Modelica.Blocks.Math.Division	Division	

Poté proveďte test regulace dýchání tak, že nahradíte konstantu nastavující tissuesFlow (tj. nahradíme tissuesFlowRateConst) skokem z Modelica.Blocks.Step. **Pozor ale na jednotky!!** Zde totiž zadáváme jednotky v mmol/min, ale skok bude v základních jednotkách SI (tj. mol/s). Podívejte se do kódu, jaký parametr Dymola vygenerovala. Také pozor na Step – nezadáme **na** kolik má skočit, ale **o** kolik.

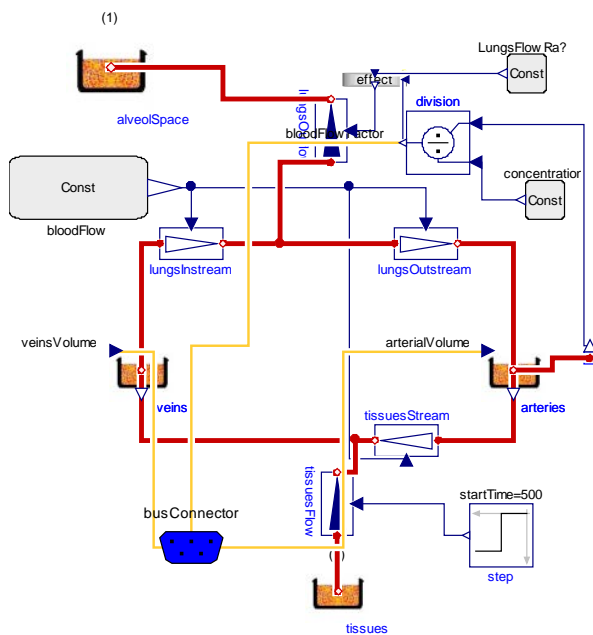
⁴ Tj. ustálená hodnota arteriální koncentrace z předchozího modelu bez řízení

Spojení

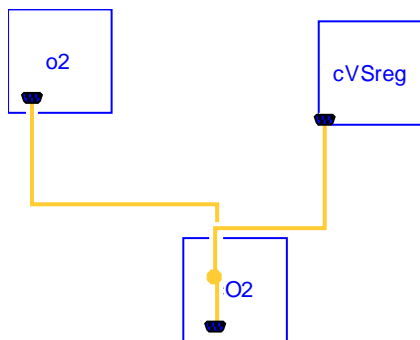
Do jednotlivých submodelů (tj. O2, CO2 a CVS) vložte BusConnector (*Types.BusConnector*). Vytvořte nový model (např. *allSystems*), vložte do něj všechny tři submodely a spojte jejich *Busy*.

Objem kompartmentů *solute*, které nám představují žíly a arterie, v modelech O2 a CO2 napojte na objem *veins* a *arteries* v modelu cirkulace.

Faktor regulace CO2 (výstup z dělení) přiveďte jako faktor minutového objemu levého srdce (*effect1* – *tam, co jsme měli připravenou konstantu*). Tím zavedeme regulaci průtoku krve jen v levé komoře⁵. Jak bude vypadat průběh toku skrz pravé srdce?



Obrázek 7: Zapojení v Modelu CO2



Obrázek 8: Celkové zapojení

⁵ Nerealistické zjednodušení – zaprvé ovládáme obě komory, zadruhé vazba je daleko složitější, neproporcionální a nelineární. Zde jde jen o demonstraci principu.

Testování zvlášť

Chceme-li nyní testovat jednotlivé submodely zvlášť, musíme si vytvořit nový submodel, kde budou natahané vstupy do BusKonektoru jako konstanty, jinak nám model nebude fungovat. Zkuste to pro model cirkulace, kde chceme otestovat jeho chování pro nárůst CO₂ o 10 %.



Obrázek 9 Testovací setup

model CVSTestParams



Obrázek 10: A vnitřek testova cího dummy bloku