Pulse model Documentation

Summary

The purpose of this document is to document the team's findings on extending Pulse Engine to integrate ECMO Machine.

Navigating Pulse Engine files/source codes

Useful Directories of Pulse Engine
Link: https://gitlab.kitware.com/physiology/engine/-/tree/master/src/cpp (engine\src\cpp\~)

Terms definition:

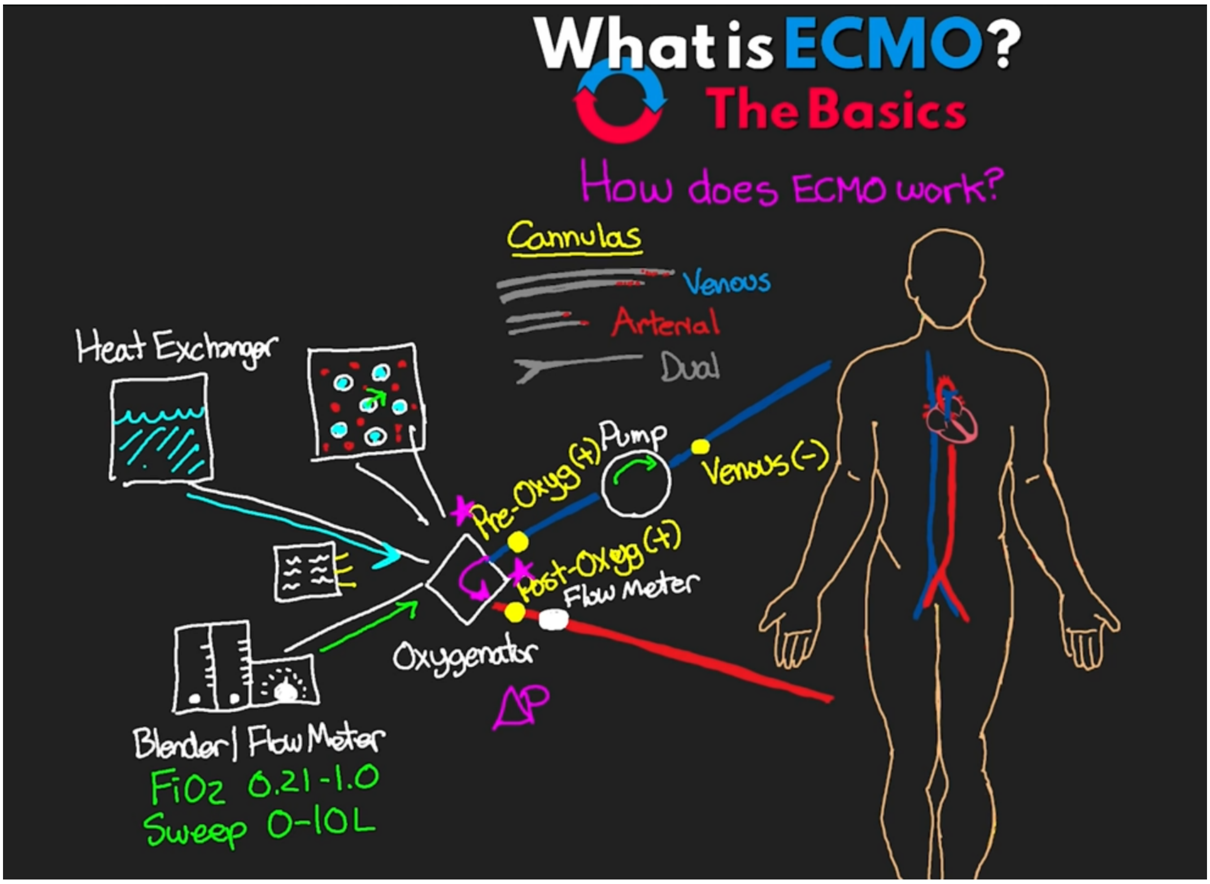
cdm: common data model

cpm: common pulse model, built on top of CDM

cdm\compartment	Extend "fluid", "substances", "thermal" compartments needed for ECMO
cpm\equipment	Add modules and implementing the logic of ECMO machine
cpm\physiology	Simulate physiology data (has bloodChemistry, Respiratory, Cardiovascular files)
howto\	Simulate logic of ECMO machine functionalities and patient scenarios
how-to\Howto-RunThreaded.cpp howto\EngineHowTo.cpp	Example of threading (running model in the background while the main function to get user input will be paused after each command execution) Uncomment HowToDynamicHemorrhage() function for threading example For real time simulation, we will need to modify the codes such that the thread running the model can be paused/put to sleep temporarily
scenario_driver\	Run how-to file to simulate ECMO
ecmo	Our own controller/logic implementations for ECMOSTP example. It contains the code and logic for **communication with DB, threading function, processing scenarios and actions, simulating the equipments**

ECMO Methodology

Reference: https://pulse.kitware.com/_mechanical_ventilator_methodology.htm I
Overview:



Initialisation Stage

- Set RPM
- Set Flow
- Initialise patient
- initialize pump
- initialize oxygenator

Pre-process Stage

- Changing/tuning ECMO RPM & gas flow if required
- Pull Data from console and patients

Process Stage

Simulation of ECMO

- Pump
- Oxygenator
- Gas blender

Calculation per cycle

1. Pressure Drop Cannula
2. Pressure Drop Tubing

3. Pressure Drop Oxygenator
4. Function for RMP < 1500
5. Function Oxygen through membrane -> saturation
6. CO2

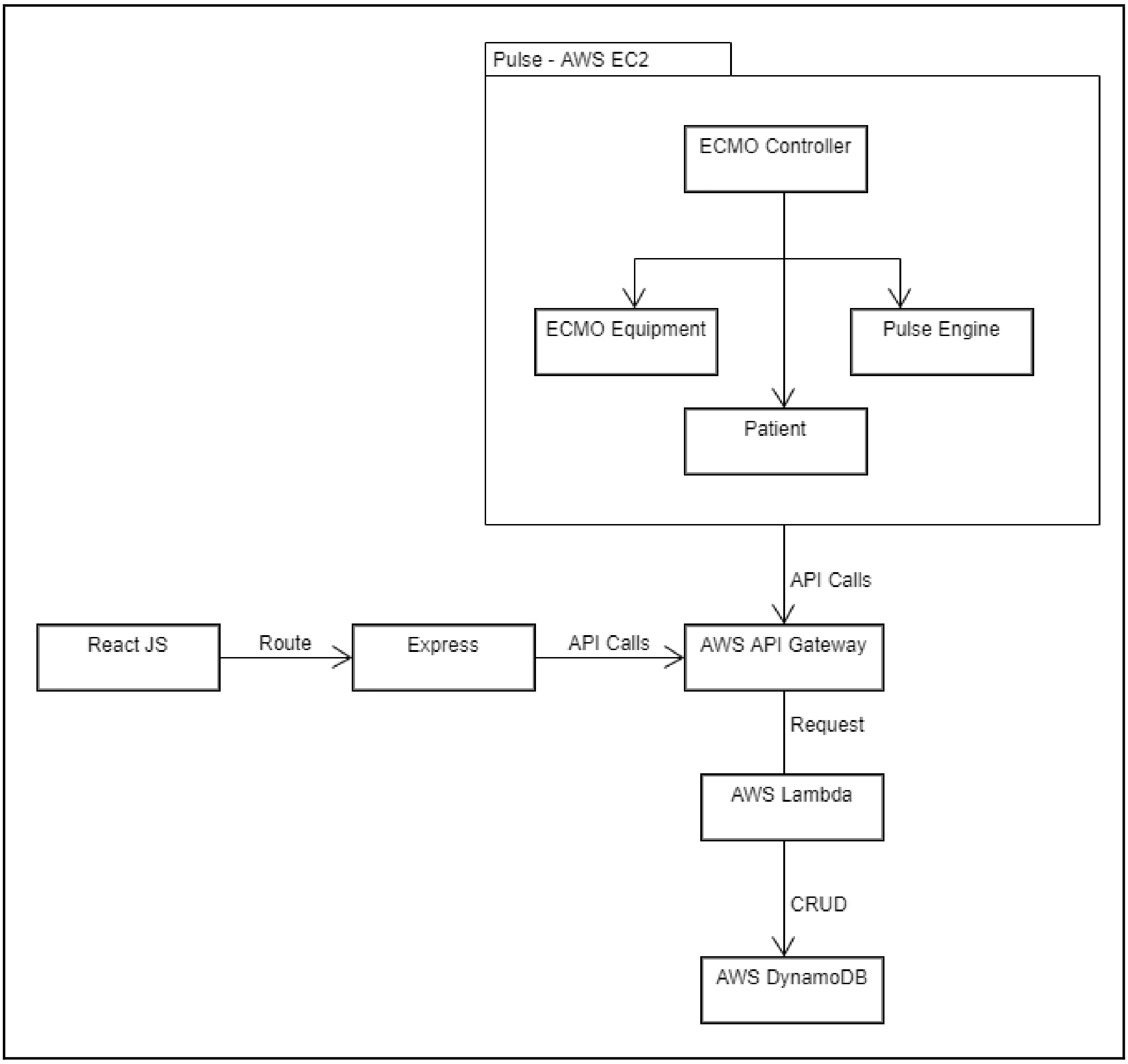
Post Process Stage

Flow back into the body?

How to build and run Pulse Engine on terminal?

```
#Install dependencies (CMAKE not included)
sudo apt-get install openjdk-8-jdk

#add JAVA to Path
export JAVA_HOME='/usr/lib/jvm/java-8-openjdk-amd64'

# cd to pulse directory after git clone
cd Pulse ### Change this line for pulse dir

#Build Engine
mkdir builds
cd builds
cmake -DCMAKE_BUILD_TYPE:STRING=Release ../engine
make -j4
OR
cmake --build .

# To Run
cd install/bin
./PulseScenarioDriver VitalsMonitor.json ### This runs the scenario driver application

### if you're running the version of pulse engine on Google doc, you should have an
executable called ecmoDriver run the following line for the executable:
./ecmoDriver

OR
./HowToDriver   # to run How-to (need to uncomment/comment functions for example)
```

How to stream data from Pulse Engine to EMCO Web Application ?



Why we need to do real-time simulation - Threading example

0	100	200	300	400	500	600	700	800	900	1000	1100
1 s (without pause)			Patient dies								second
With pause											Patient dies

Q&A

The documentation below is our meeting minutes with Andrew regarding ECMO model conversion.

Meeting With Andrew #2 (14 Sep)

Pulse Engine Extending ECMO Simulation

Simulating Pump

Equation for RPM < 1500:
Pressure (what pressure exactly is this?) = (0.0285714 * RPM) + Q_LVAD

Terminology:
- Pressure: What pressure is this?
- Q_LVAD - Flow from left ventricle? Blood flow from pump?

Equation for RPM > 1500:
$\Delta p(q_b, RPM) = p_1 + p_2 \times q_b + p_3 \times RPM + p_4 \times q_b^2 + p_5 \times q_b \times RPM + p_6 \times q_b^2 + p_7 \times RPM^3 + p_8 \times q_b^2 \times RPM + p_9 \times q_b \times RPM^2$

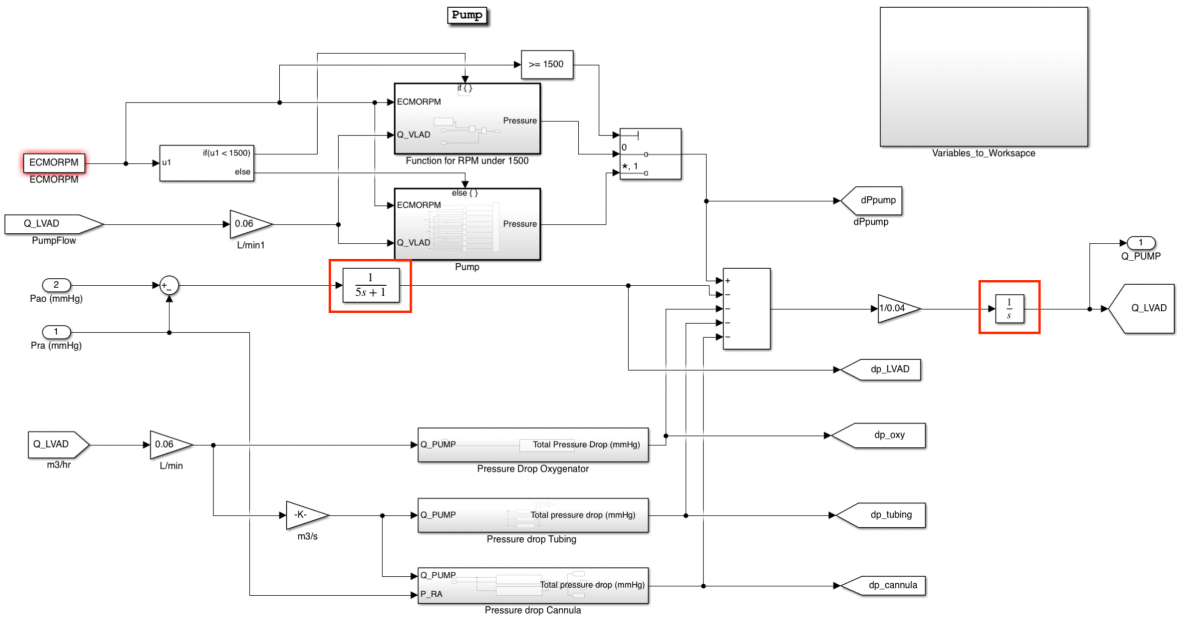
We referred to this Simulink model for the pump:
https://drive.google.com/drive/folders/1jjLR0wNTR6vguy3gcvEmEY4e5S9aCUpi?usp=sharing

All available/readable values from Pulse Engine:
https://pulse.kitware.com/_c_d_m_tables.html

Questions:

1. There is an exampledataset.mat file attached to the Simulink model. Some of the variables, e.g. OxygenatorMembraneThrombosis, pericade tamponade, Cannula sizes, tubing length, etc. seem to be constants in the model. **Are we safe to assume that these constant values will not change**? (at the moment can be constant, may need to intialise)

2. **We need help with converting the transfer and integrator functions** (e.g. 1/s blocks) **into our regular C++ codes**. From what I have found from Google search, we would need to integrate it to get some form of ODE. **What are the functions that we need to integrate in this case**?



Transfer function

(Red box on the left) Transfer function - Laplace transformation

→ Take mean value of data for 5 seconds (moving average) for mean aorta pressure value.

Integrator (Red box on the right)

- pressure divided by resistance to give instantaneous flow that is being integrated over time to give total flow (ml/min or s)
    - 1/0.04 is resistance
- Trapezoidal integration is used here

Implementation:
Var init - can initiate/calculate velocity based on the acceleration

```
double flowAcceleration = totalGeneratedPressure() / 0.04;

pumpFlow = flowVelocity + flowAcceleration * (1/refreshRate); //
Integrator block
```

Flow acceleration is pressure (result from the +---- block) divided by the resistance (0.04)
Flow velocity initialised as 58.65.

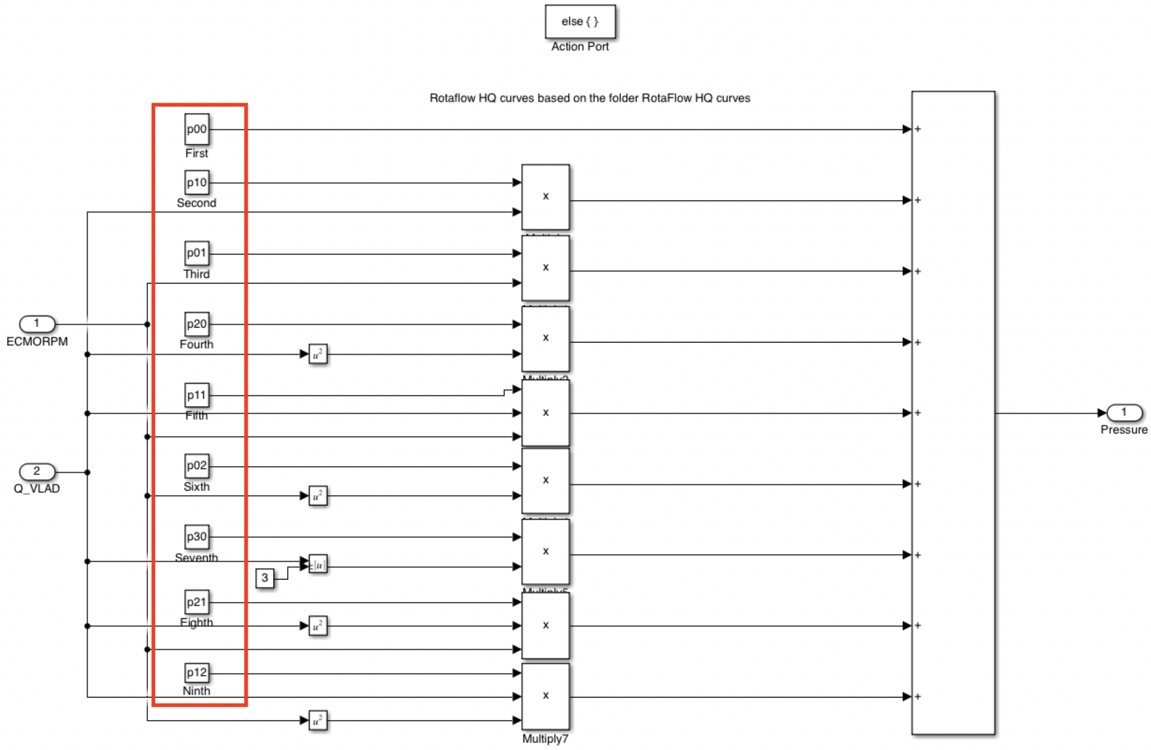
3. **Gain block: if it's a gain of 0.06, do we multiply the input by 0.06?**

Yupp

4. **Could you help us verify if the following variables/terminologies are correct? And what are the units for these variables?**

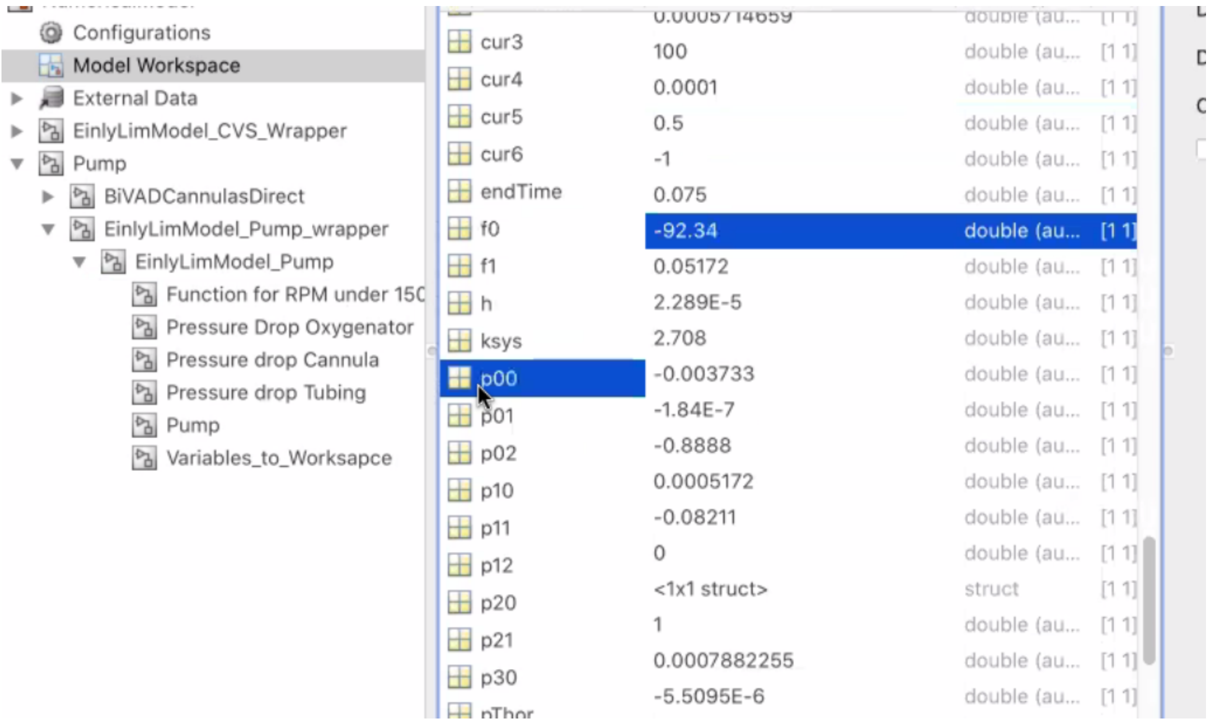
Variable	**Unit**	**Name**	**Location in Pulse**
Pao	mmHg	Aortic pressure	cdm/system/physiology/bloodchemistrysystem
Q_LVAD	(depends)	Blood flow (volume/time) from pump (ecmo circuit) Pump Q from last to being fed to next cycle	
Pra	mmHg	Right atrium blood pressure	
dPpump	mmHgmmHg	Pressure generated by pump	ECMO variables
dp_LVAD	mmHg	Pressure drop in pump (against aortic pressure)	ECMO variables
dp_oxy	mmHg	drop in blood pressure in oxygenator	ECMO variables
dp_tubing	mmHg	Pressure drop in blood across tubing	ECMO variables
dp_cannula	mmHg	Pressure drop in blood across cannula	ECMO variables

Subsystem for ECMORPM > 1500:



5. **Equation for RPM > 1500:**
   $\Delta p\,(q_b\,,\,RPM) = p_1 + p_2 \times q_b + p_3 \times RPM + p_4 \times q_b^2 + p_5 \times q_b \times RPM + p_6 \times q_b^2 + p_7 \times RPM^3 + p_8 \times q_b^2 \times RPM + p_9 \times q_b \times RPM^2$

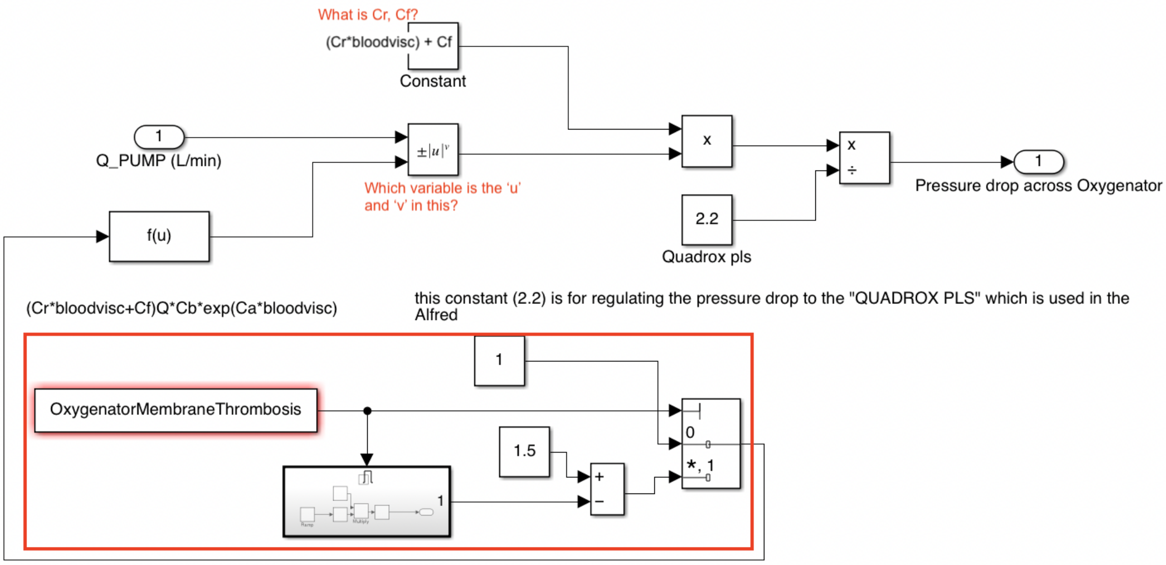
   We couldn't locate the constants (p1 - p9 in the red box) for pump modelling equation 2.15 of the Master Thesis. How do we obtain the constants p1-p9 using curve fitting?
   Try running MiniScript.m (error- lack of libraries)
   Model workspace



6. ***Is the resulting pressure output (the rightmost outlet) the blood pressure generated by the pump? What should be the correct unit?***

Yes. mmHg

Subsystem for Pressure drop across oxygenator



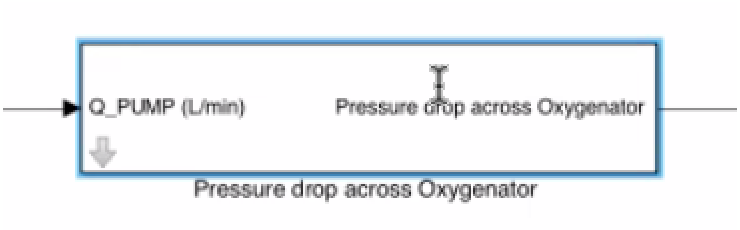
Oxygenator membrane thrombosis function, ads exp. resistance based on OxygenatorMembraneThrombosis folder

7. For the exponential math function block highlighted in the diagram, **which input variable is the *u* and *v***?
   u = input 1
   v= input 2
   Qpump[u] to f(u)[v],

8. The following question refers to the function:

   *Cr*bloodvisc + Cf)Q*Cb*exp(Ca*bloodVisc)*
   (pressure drop not gas exchange)

   a. **What is Cr, Cf, Cb, Ca**? (modifying pump flow due to the increase in resistance in oxygenator, how blood clot size affect the resistance/pressure drop) constants (double click,

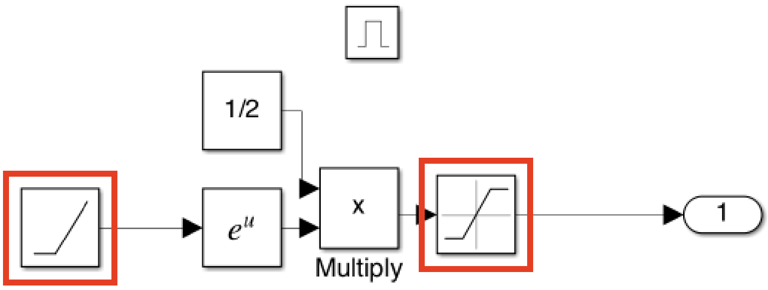
   

   Check function wrapper

   b. **The Blood Viscosity is not found in Pulse Engine**. Is there a range of values that we can refer to for the calculations?
   Range: 1.5 - 5 cP, typically around 2-4 for healthy, lookup table for hematocrit

-> Blood viscosity, 37c

Subsystem in Pressure drop across oxygenator

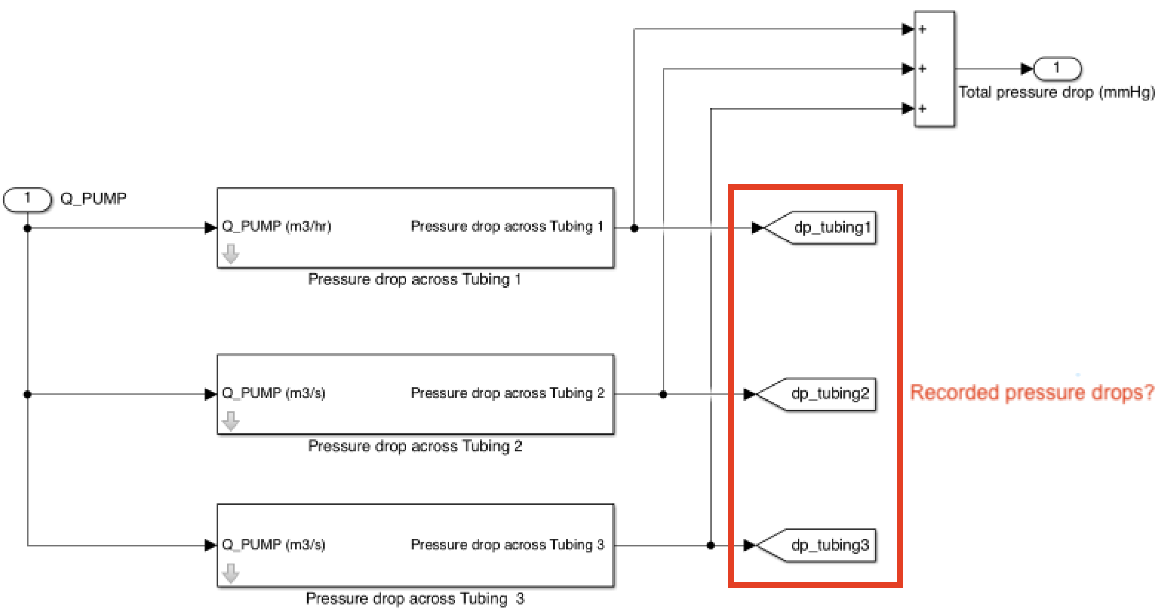


for blood clot

9. **Ramp: if the slope is 0.035, is it right to say that the output value of the block increases by 0.035 every cycle?**

Saturation (Lower limit = 0, Upper limit = 16.5). Is it right to say that the saturation block allows input values coming from Multiply block to increase or decrease within the range of 0 - 16.5? Once it hits the upper limit it will not increase anymore

Subsystem in Pressure drop across Tubing

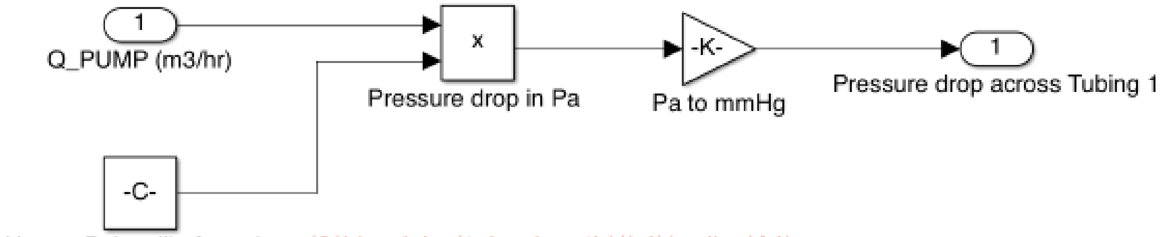


10. **Are the dp_tubing[1,2,3] recorded pressure drops from the function wrap?**
    mmHg, pressure drop

11. **Tubing 1, 2 and 3 are essentially the same functions. What is the purpose of having 3 different functions to simulate the pressure drop?**
    PD caused by friction in the tube. Tubing p -> o (70-75mm?) , o -> c , a -> p, same type of tube but different LENGTHs. Calculating different elements of the system.

(include lookup table for blood viscosity)

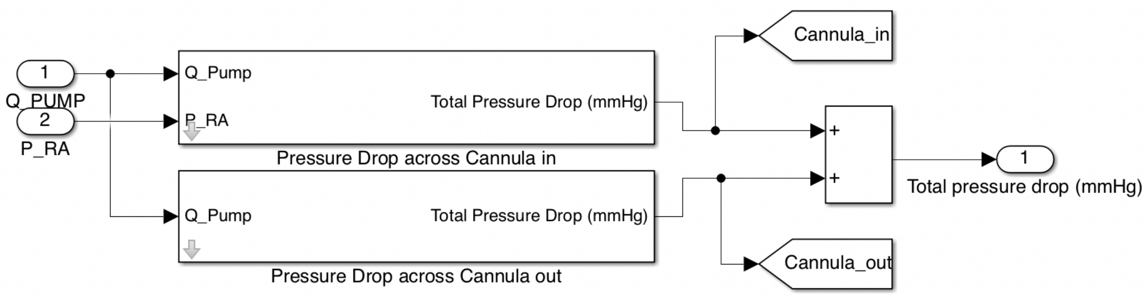
Tubing Subsystems:



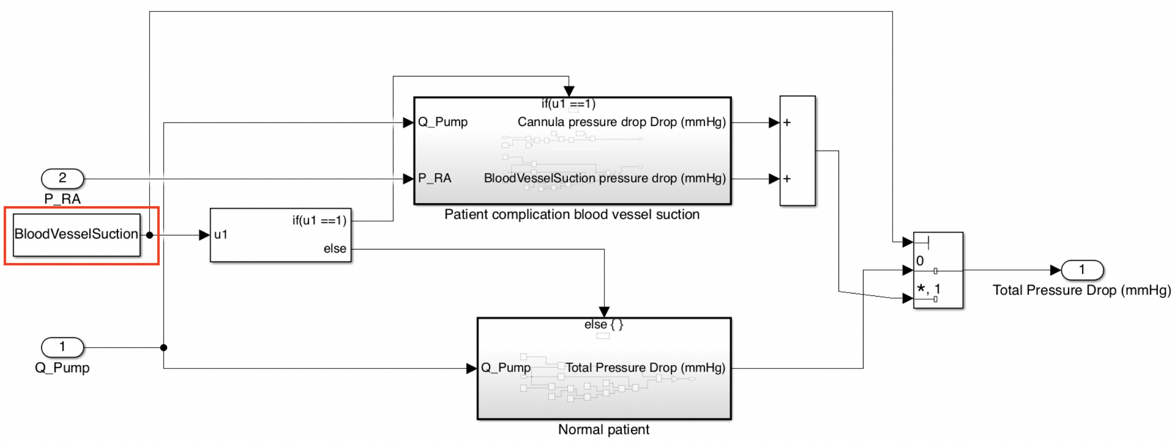
Hagen-Poiseuille formula = (8*bloodvisc*tube_length)/(pi*(radius)^4)

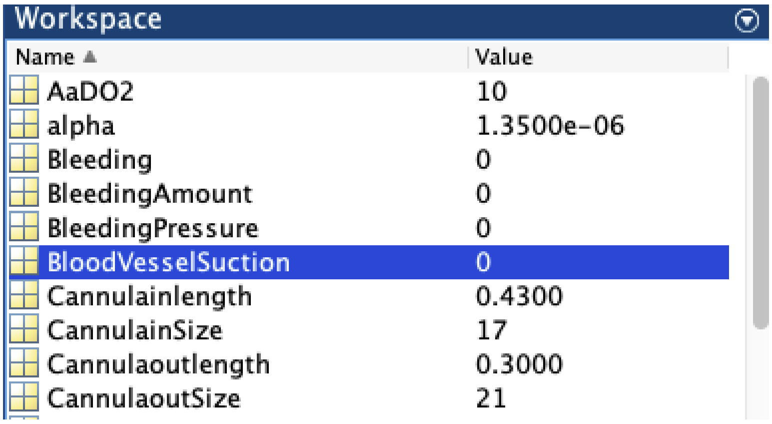
12. **Hagen-Poiseuille formula requires blood viscosity, but blood viscosity is not a simulated variable in Pulse - Can we assume a range for the viscosity in this case?**
(Already answered)

Subsystem in Pressure drop across Cannula



Pressure drop across **cannula in**





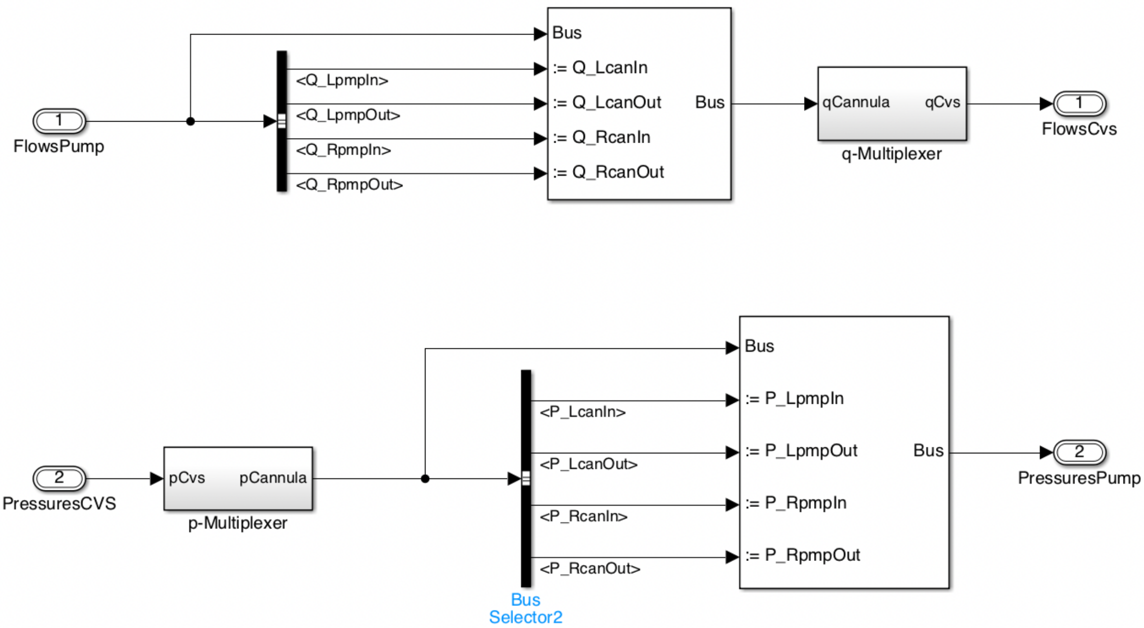
Workspace

Name ▲	Value
AaDO2	10
alpha	1.3500e-06
Bleeding	0
BleedingAmount	0
BleedingPressure	0
BloodVesselSuction	0
Cannulainlength	0.4300
CannulainSize	17
Cannulaoutlength	0.3000
CannulaoutSize	21

13. **BloodVesselSuction - Initialised as 0 in the sample data. Is the value boolean true/false flag?**
    Will change in real-time - to be developed for version 2.0

14. **Does the complication of blood vessel suction occur only in cannula in, and kinked return line occur only in cannula out?**
    Suction on cannula in only because of pump

BiVADCannulasDirect



15. **What's the difference between L and R variables?**

L and R should have the same value, don't need in ecmo. Only need "L" values

Meeting With Andrew #3 (21 Sep)

Pulse Engine Extending ECMO Simulation

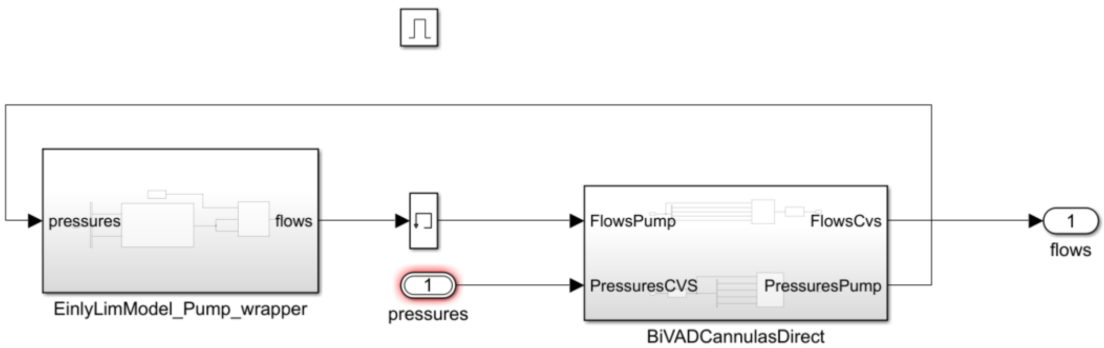
Simulating Oxygenator

Terminology/variables and where to locate them in pulse and exampleDataSet.mat file

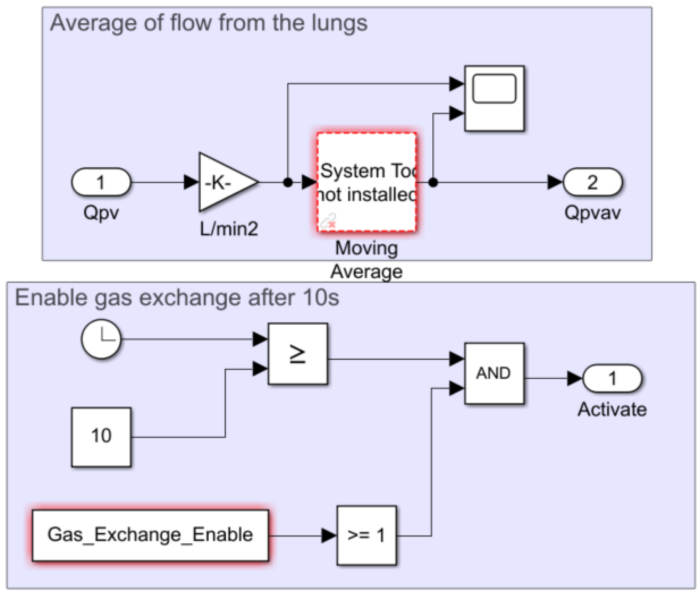
- **hct** - hematocrit, amount of haemoglobin in red blood cells
    - Pulse: cdm/system/physiology/BloodChemistry OR cdm/patient/assessment/CompleteBloodCount
- α_{O2} - Solubility of O_2
    - Pulse: https://pulse.kitware.com/_c_d_m_tables.html#SubstanceTable cdm/Substance/substance - Solubility coefficient
- $[H]_{rbc}$ - haemoglobin concentration
    - Pulse: cdm/patient/assessment/CompleteBloodCount (hemoglobin)
    - Can we use Hemoglobin content for this? https://pulse.kitware.com/_c_d_m_tables.html#BloodChemistrySystemTable
- cap_b - binding capacity
    - $cap_b = hct \times [H]_{rbc}$

Questions:

Pump

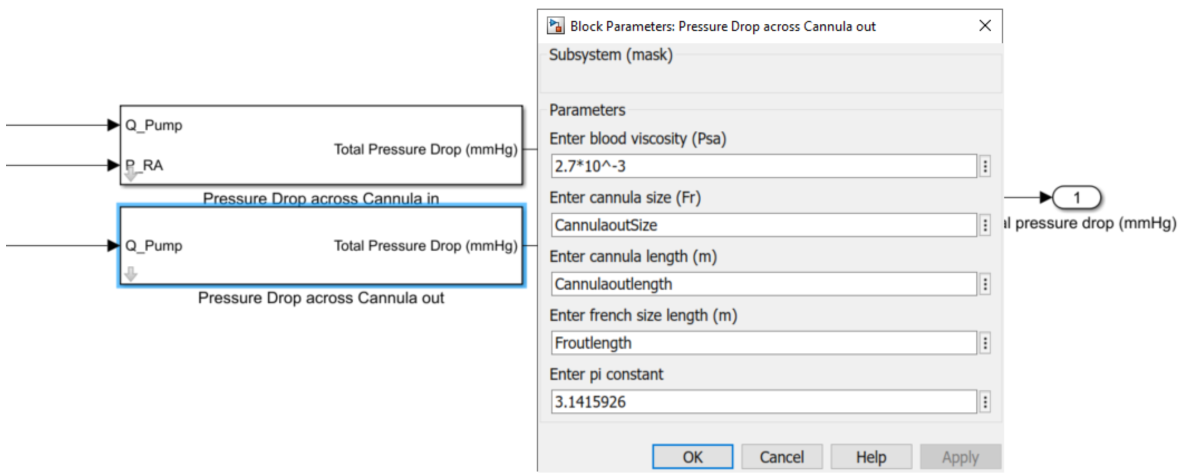


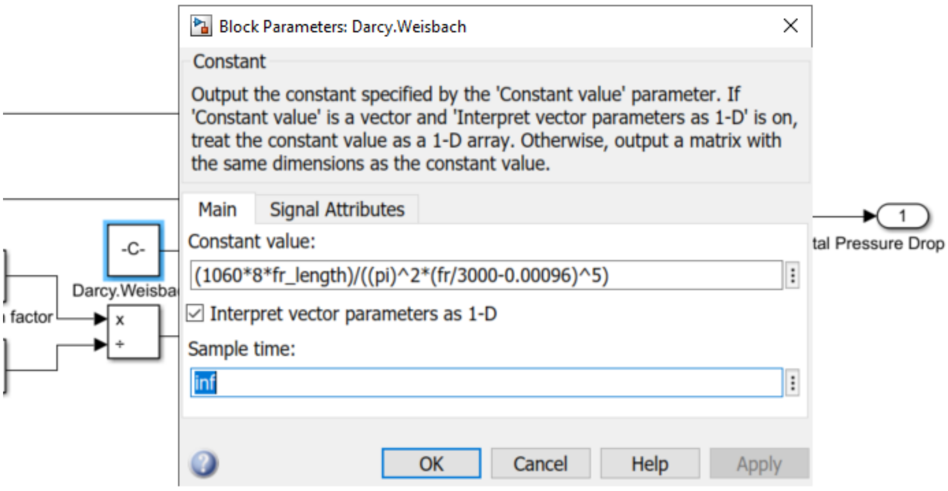
1. Pump_wrapper needs *pressures*(PressuresPump) from BiVADCannulasDirect to calculate *flows* for flows pump (Cyclical). How would we initialise/get around this?
    - Starts with 0 as initialised, then reaches a steady state as the simulation goes on



2. What does the *average flow from the lungs* in "pre logic" do?

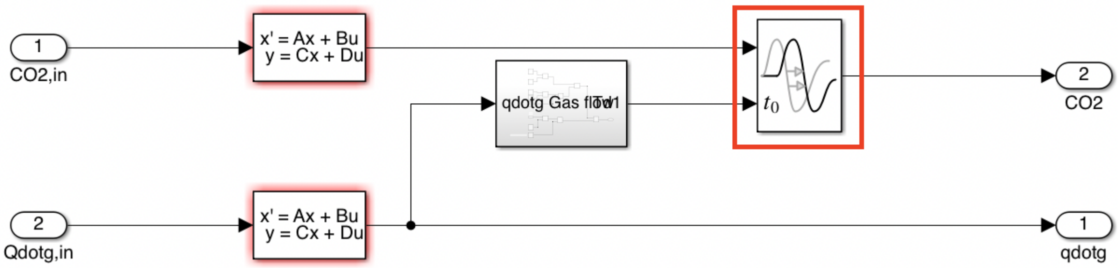
- Doing pre-calculations and changing the units for the vars
- Gas exchange starts after 10s
- Moving average (updated every 5 seconds)





3. Is Fr (cannula size) and fr_length the same? (formula for darcy weisbach)
   fr_length is length of cannula, fr is the size (diameter)

Gas Exchange/ Gas Blender



4. x', y - partial differential equations
   Read on state space models

5. What does the variable time delay mean? What does the buffer size in the block do?
- Delay the output by a certain time

6. Does the diagram above represent equation 2.16 and 2.17 (below) ? If we didn't misunderstand this, where is exponential in equation 2.16 modelled?

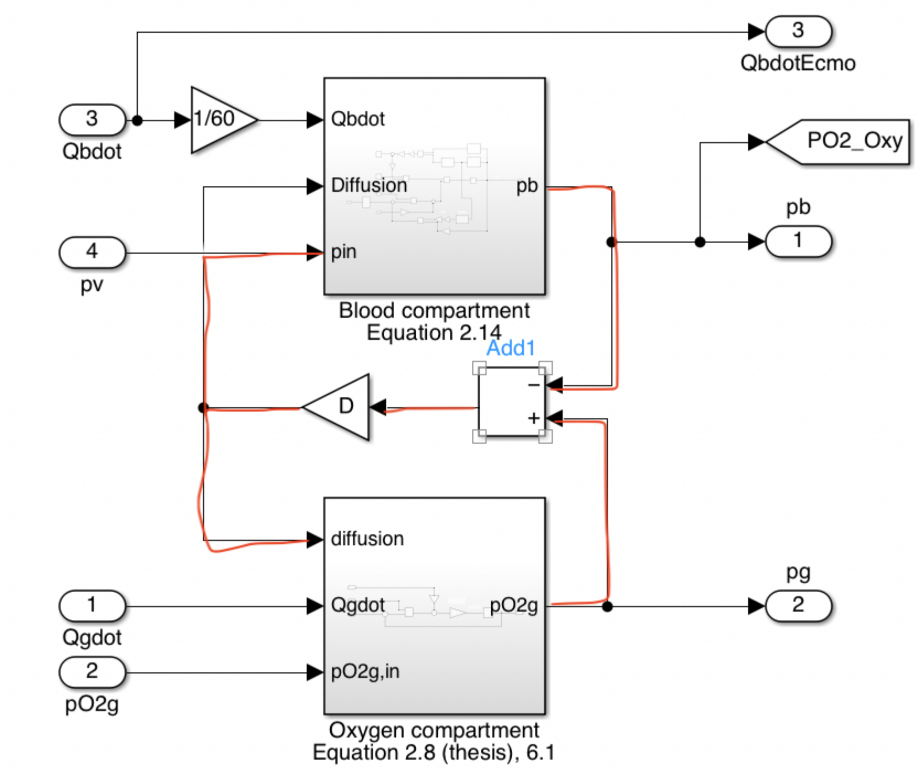
$$FiO_2(s) = \frac{1}{T_{gb} * s + 1} * e^{-T_d(\dot{Q}_g) * s} * FiO_{2,in}(s) \quad (2.16)$$

$$\dot{Q}_g(s) = \frac{1}{T_{gb} * s + 1} * \dot{Q}_{g,in}(s) \quad (2.17)$$

7. Does the block with x' = Ax + Bu and y = Cx + Du model the laplace fraction

a. Read on state space models

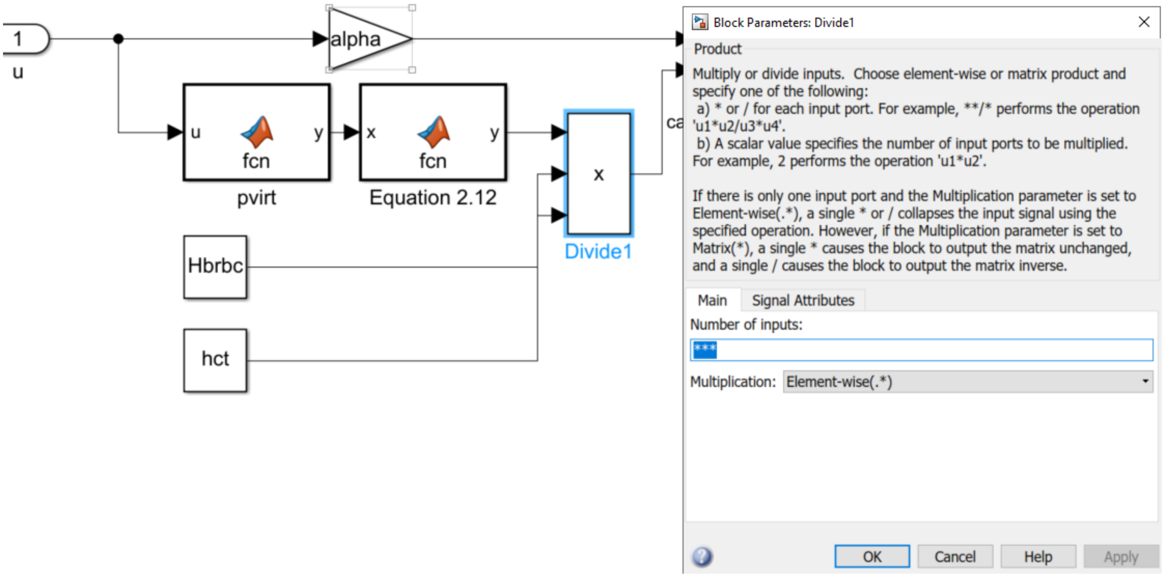
Oxygenator



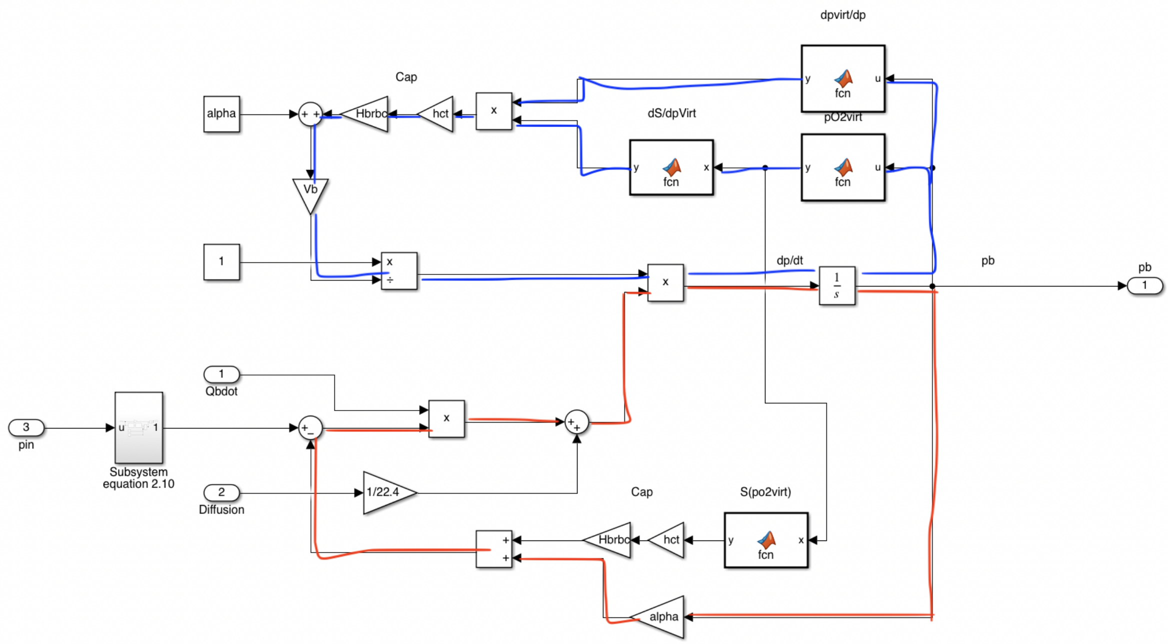
8. Where do we start with this cycle (for initialisation)? Because the blood compartment equation needs the "Diffusion" value.

Oxygen compartment calculated first then blood compartment. Have a memory element somewhere for the diffusion element.

9. What does the constant above mean?
    a. Pv =
    b. Pb = PO2_Oxy = partial pressure of blood? Is is the same as PO2_Oxy
    c. Pg (PO2g) = partial pressure of oxygen in gas?



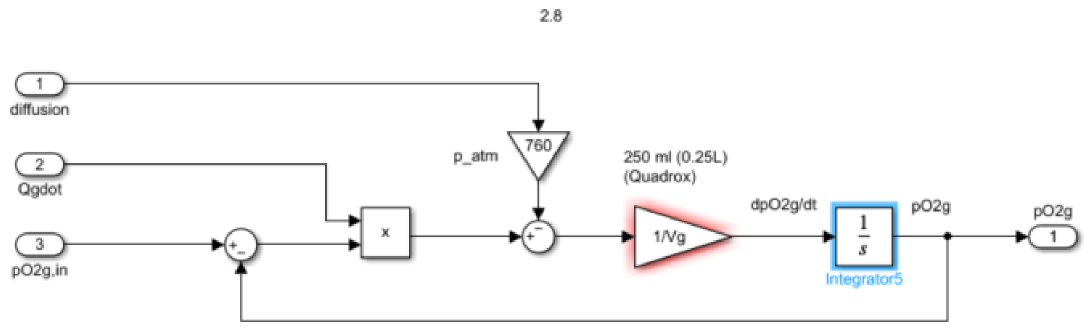
10. Do we do division or multiplication for this block?



11. Clarify Integrator [1/s] for **blood compartment**. Equation 2.14

12. Another cycle as drawn in the diagram. How do we initialise this?

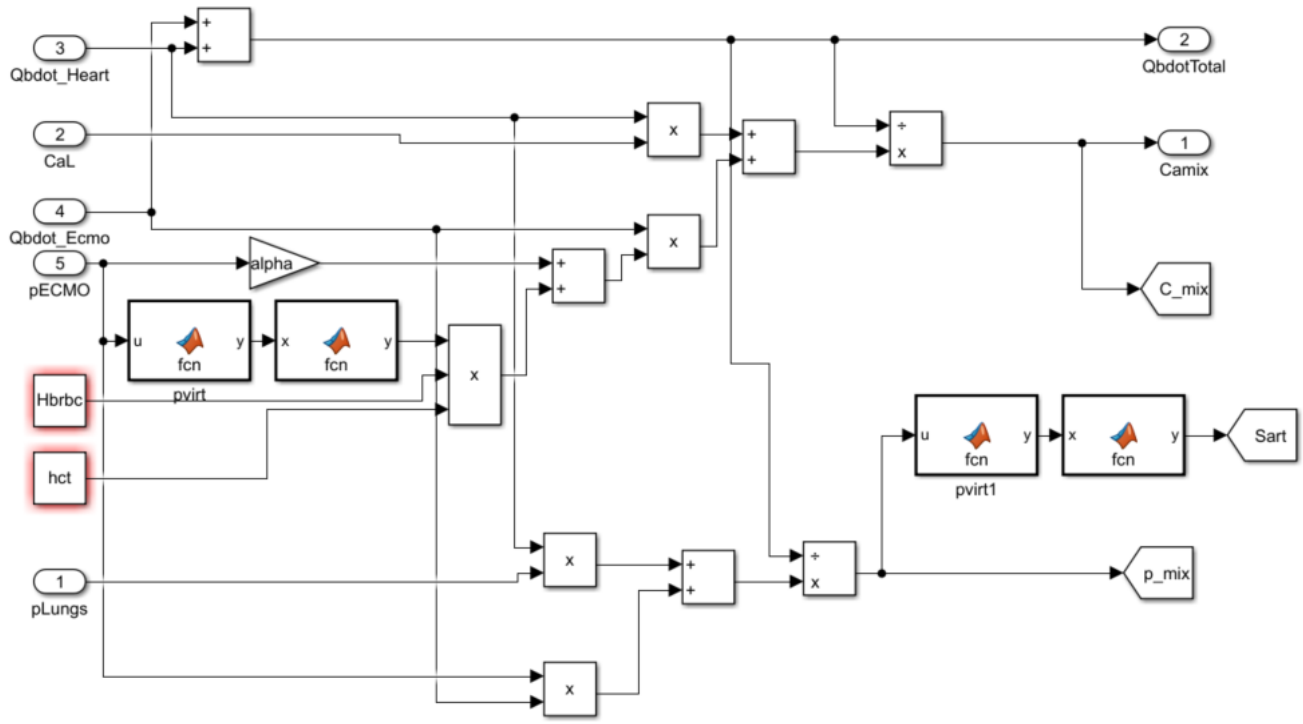
13. Can we apply the same understanding of **blood compartment** to **Tissues**?



14. Clarify integrator [1/s] for **both oxygen compartments**. (equation 2.8)
- Converts acceleration to velocity.
- Integrator init condition = 450

*Might be better to just focus on getting the blood flow, pump done to get the ecmo started. Show if we increase pump then arterial pressure increases

Mixing Ecmo + heart



15. What do the variables above mean? (the equation is not in the thesis)
    a. CaL

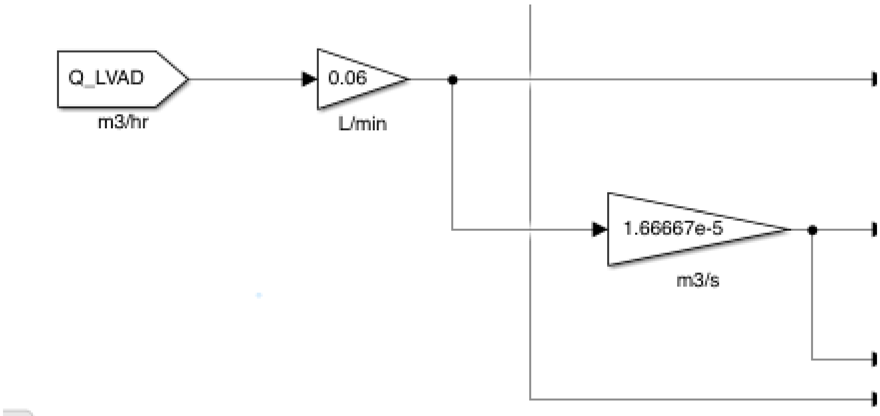
b. pEcmo
c. pLungs
d. C_mix
e. P_mix
f. Camix
g. QbdotTotal, total from ecmo + heart?
h. Sart

Meeting with Andrew #4 (30 Sep)

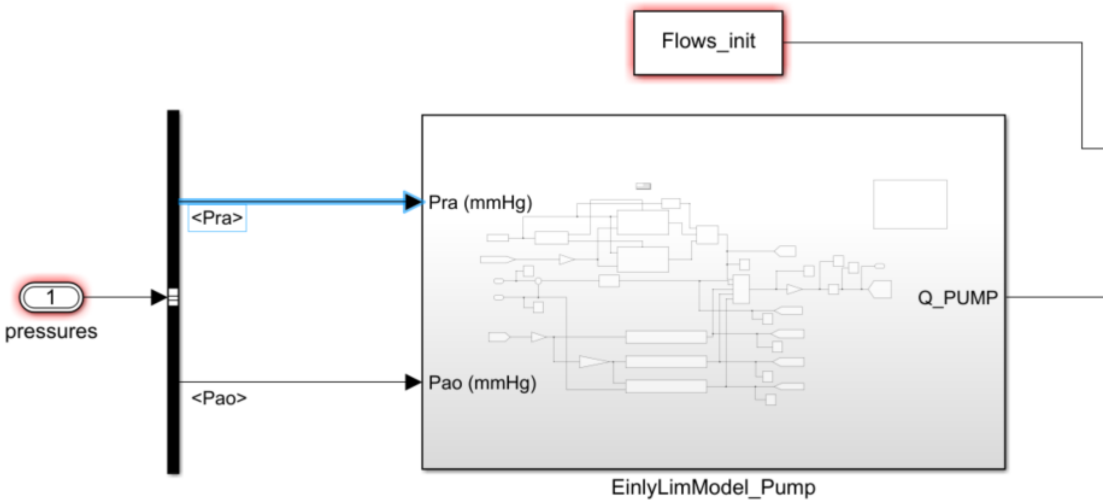
Questions

1. Unit conversion for pump flow
   Shouldn't we multiply Q_LVAD (m3/hr - should be m3/s) by 16.667 to get flow in

   The correct units should be
   ml/s → l/min → m3/s



2. Transfer function
   Since we are doing a 5 seconds moving average, the total number of values we need to consider at any one points will be 250 (since refresh rate is 50Hz). Is this correct?



3. The right atrium pressure is not simulated in the Pulse Engine. Should we implement a side cardiovascular system to calculate the right atrium pressure? Or can we use mock data for now?

- Possibly this
    - PulmonaryCapillariesWedgePressure
    - Mean central venous pressure

4. Is this code translation correct for the integrator block?

//from initial condition of integrator block

```
double flowVelocity = 58.65;
```

```
double flowAcceleration = totalGeneratedPressure() / 0.04;

pumpFlow = flowVelocity + flowAcceleration * (1/refreshRate); //
Integrator block
```

prev flow + 0.5 * (Value from this step - value from last) * (time step)