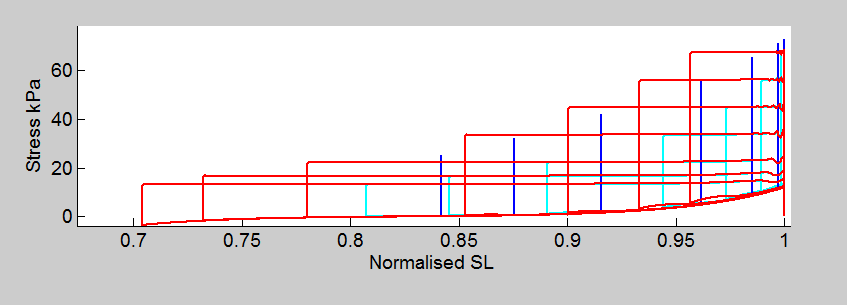


Above is the work-loop data (cyan) and isometric data. These data come from the 20th work-loop in a series of work-loops (steady-state). In the previous week I was having trouble recreating this figure (that I first created on 23-3-2017. The issue I was having was I was plotting the 20th iteration of the isometric contraction vs. the 1st iteration of the WL with isometric calcium. What I am going to do next is compare subplot 3 of the above figure with a similar figure that plots the 1st iteration data (for the isometric contraction and work-loop contraction).

Below is the Work-loop and Isometric data overlay for the FIRST contraction of each type. No achieving steady-state in this scenario. Now, it is more appropriate to compare the work-loops with isometric calcium to THIS figure (since the work-loops with isometric calcium only go through one full contraction per simulation… not 20)



red- Work-loop with isometric calcium

blue- 1st isometric contraction

cyan- 1st work-loop with dynamic calcium



The blue in this figure is the isometric calcium transient that results from running the megan-model at varying sarcomere lengths. The red is the fixed calcium transient that has been inserted into the work-loop simulation. It is no wonder that the work-loops with “isometric” calcium look so funny… the isometric calcium transient inserted into the model does not actually resemble an isometric calcium transient.



cyan = the “fitted” calcium transient that is being inserted into the work-loops. it should be fitted to look similar to the red trace… but that is clearly not the case!!

red = the calcium transient data that we used for the python optimisation code.

blue = the isometric calcium transients for varying sarcomere lengths



Above, the work-loops and isometric bars are run to steady state. The WL with isometric Cai is NOT to steady state.



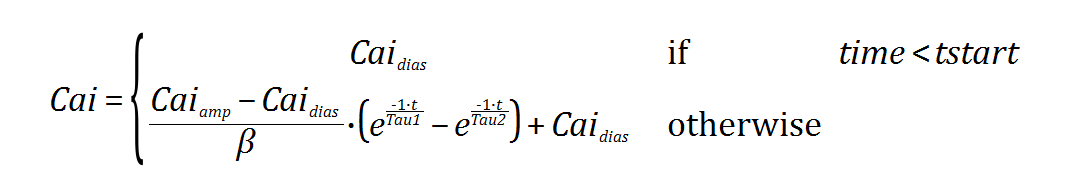


RED = isometric

Blue = dynamic WL

Cyan = WL eith isometric calcium

here are the ES curves only (easier to see). This time all three contraction scenarios are run only onece (NOT to steady state)





Sad news—Running the WL with isometric calcium to steady state makes no difference whatsoever ☹



Questions to ask in order to understand why running the work-loops with isometric calcium to steady state does not influence the ES curve at all… (see above figure with light blue and light green work-loops—the light blue is NOT to steady state while the green IS to steady state)

1. Does the “starting” afterload make a difference? Perhaps I should find the calcium transient for an isometric contraction whose max force is higher than the diastolic passive force for the WL scenario…
   1. What is the passive force for WL scenario? 🡪 12.03kPa (0.1074)… the lowest afterload setting is 0.12 so this would not be the issue
2. Why would the work-loop protocol be able to shorten to isometric values at lower afterloads?
   1. The force plateau lasts longer when afterload is longer… so the sarcomere has more time to shorten and therefore can shorten more (proportionally)
3. could the calcium fit be not good enough?
4. I could try fitting a different isometric Ca2+ transient?
5. I could rerun the WL with isometric calcium so that we can improve the look of the passive recovery. (this should be taken care of in the WL run to steady state though…) Why isn’t this working??
6. Have you gone and checked the “main” parameters to make sure there is no glaring mistake or value inconsistency?
7. have you checked the equation errors that you picked up on (refer to notes)

What do I need in my GIT?

Main Models and the model versions that they use:

1. MeganModel\_workloop\_dynamicCai: This model is used to run the work-loop contractions and the isometric contractions that have dynamic calcium. There is an option to turn VD on or off here but that requires fiddling with the XB\_model at the moment.
   1. XB\_model used: RiceTran\_workloop\_cycling\_dynamicCai: As stated above, there is currently the option to have VD (velocity dependence) off or on in this model.
   2. QuickReleaseProtocol: This model is present only to add the option of running a “quick-release” experiment. This was necessary for my (Megan’s) project because I simulated the Kurihara quick-release experiment in order to validate the model that I am using.
   3. HINCHcalcium\_component: We use this Hinch model to dynamically simulate our intracellular calcium transient
      1. The HINCH\_calcium\_component model uses a model called multiple-stimulus.xml to initiate the start of the calcium transient.

TO RUN THE MODEL IN OpenCMISS (with dynamic calcium):

* 1. main\_dynamicCai: This is the main python code that is called when I run the MeganModel\_workloop\_dynamicCai.cellml on hpc. This main python code calls the actual protocol that is used to generate work-loops.
     1. Workloop\_protocol\_with\_passive\_cycling\_dynamicCai: This is the protocol mentioned above.
  2. Isometric\_protocol: this is the python code called when running MeganModel\_workloop\_dynamicCai.cellml for the purpose of simulating isometric contractions (NOT work-loops) that have dynamic Ca2+ transient.

1. MeganModel\_workloop\_isometricCai.cellml: This model is used when I want to simulate work-loops with isometric calcium.
   1. XB model used: RiceTran\_workloop\_cycling\_fixedCai.cellml
   2. QuickRrelease Protocol: same as above
   3. Calcium\_component: When inserting an isometric calcium component into work-loop simulations a fitted transient is used. This is done by changing the parameters within the RiceCaiTransient.cellml model.

TO RUN THE MODEL IN OpenCMISS (with fixed isometric calcium):

1. main.py: This is the main python code that is called when I run the MeganModel\_workloop\_isometricCai.cellml on hpc. This main python code calls the actual protocol that is used to generate work-loops.
   * 1. Workloop\_protocol\_with\_passive\_cycling2.py: This is the protocol mentioned above.

Questions to answer in order to FINISH this manuscript:

1. What models do I currently have in my GIT repository?
2. Can You create a map that shows how each of the models relate to each other
3. Have you renamed the models so that their purpose is easy to understand?
4. How are you going to get the fitted RiceTran calcium transient to be repeatable?
5. Have you tried to get hpc to run the MATLAB plotting? (To make this step faster)?
6. Have you renamed the Calcium component for the model that uses the Rice Calcium transient (the fitted isometric transient)
7. Have you run the WL simulation with the “matching” isometric calcium transient?
8. Do the models DO/ PRODUCE the results I expect them to?
9. Are your results repeatable?
10. Do you have ONE python interface from which you can recreate all data included in the Article?