Computation of instantaneous mechanical power #526

Closed **Liam-Guerrero** opened this issue on Mar 3 · 4 comments

Assignees

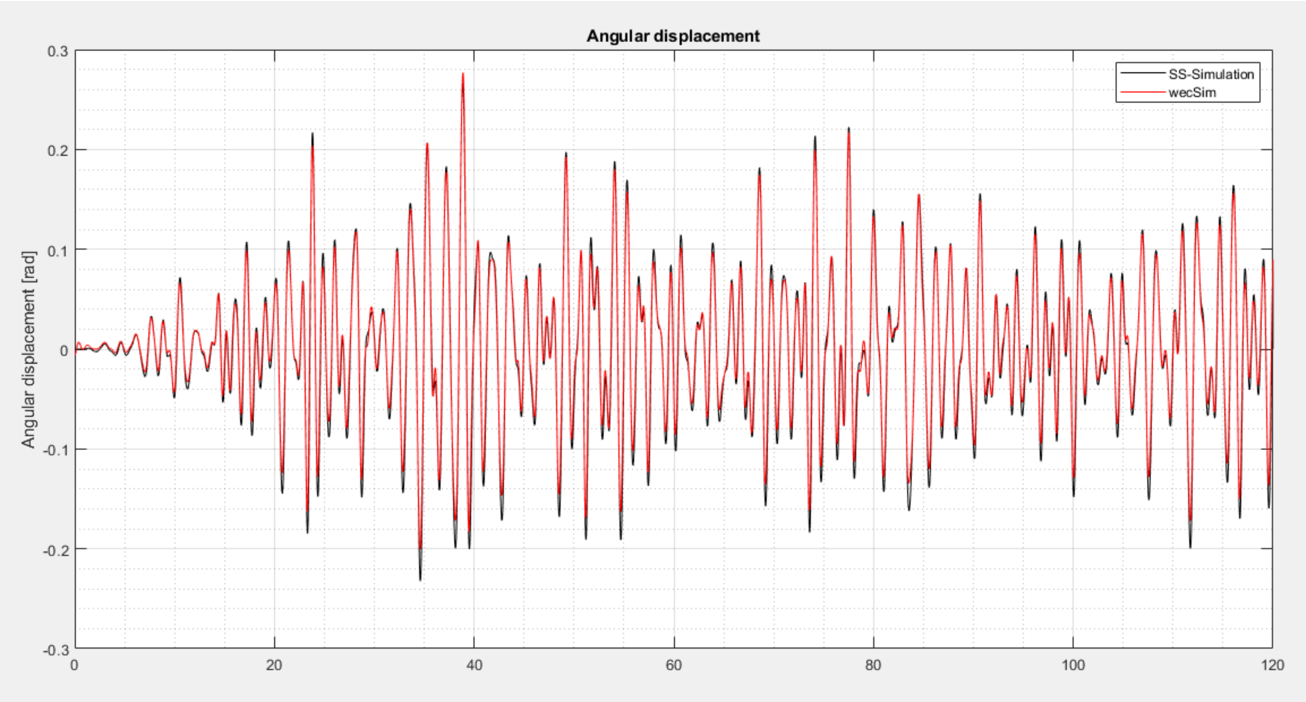
Labels Support wecccomp

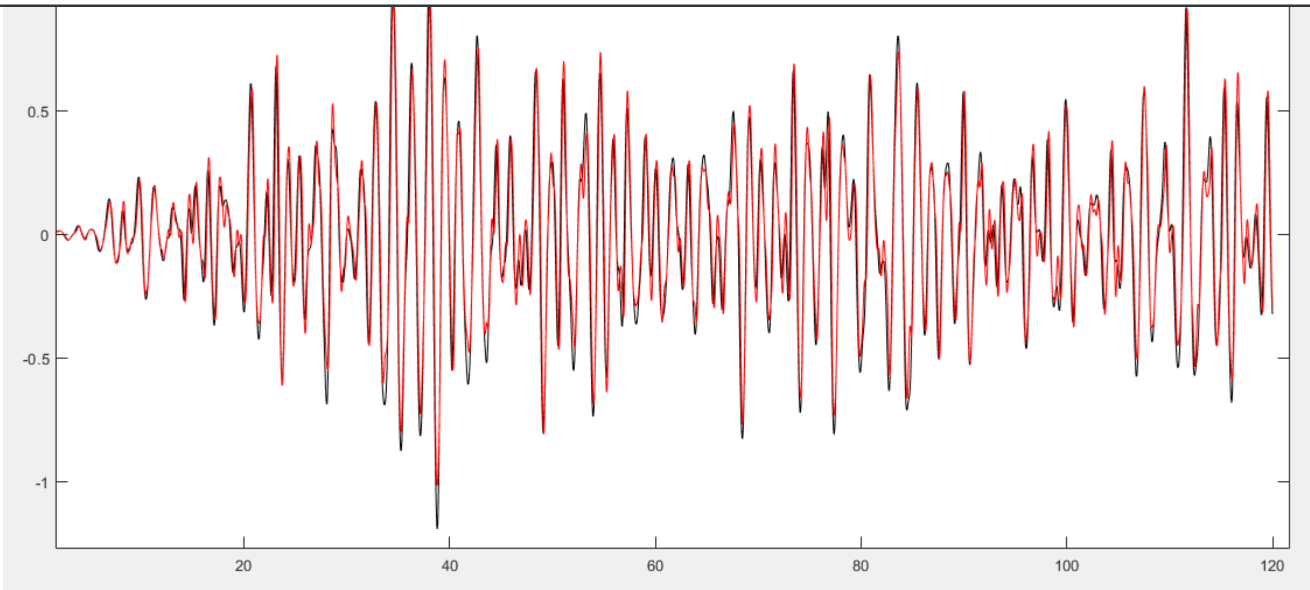
Projects Issues

Liam-Guerrero commented on Mar 3 • edited

z_wecSim.zip
Hi Guys,
I am using the linear state space developed for WECCCOMP, with the parameters given in "A competition for WEC control systems", and "An energy-maximising mpc solution to the wec control competition" but having some discrepancies with the results obtained from the wecSim model developed for that WEC.

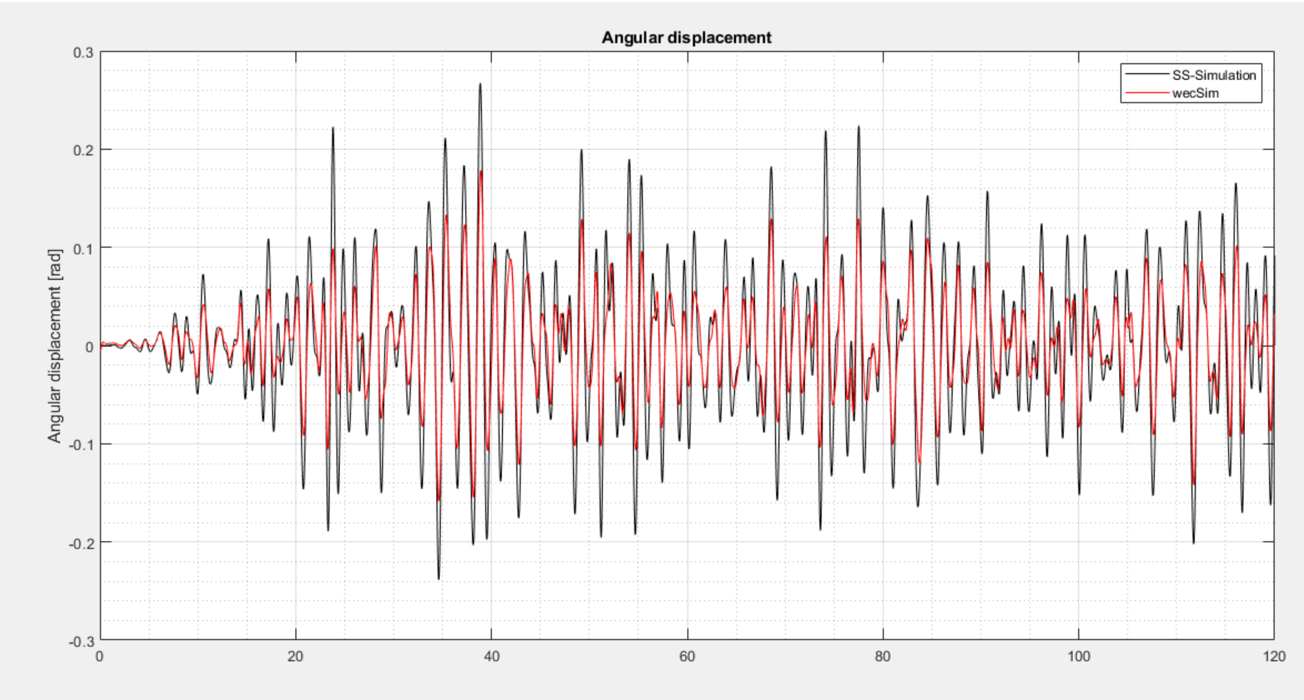
With no control (M_pto = 0)
I compared the results from the wecSim model with the obtained from the linear state space: where the arm position, velocity, and forces are seen to be in agreement. The excitation moment is in phase with the wave elevation. So far I think everything looks fine.





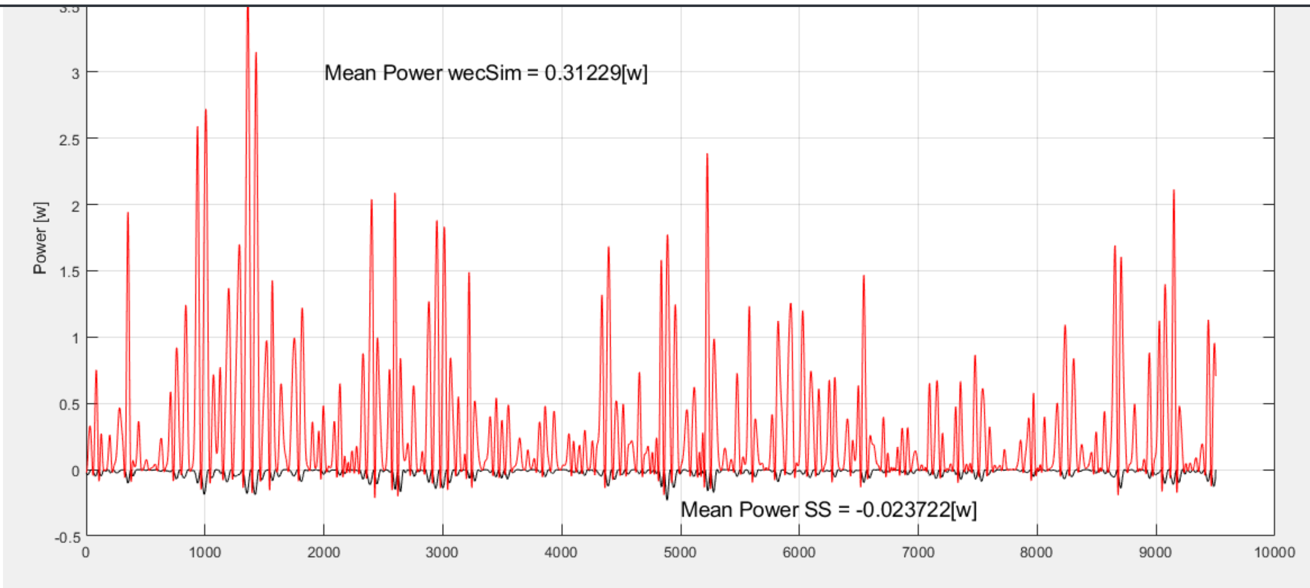
With Control (Proportional to velocity): M_pto = K3 x cc x angularvelocity (K3 and cc constants)

Here is where differences are notarial: the arm position, velocity from wecSim model are no longer in agreement with the obtained from the State-space model.



Also, the instantaneous mechanical power computed from the ss model hugely varies compared with the wecSim model.
For the instantaneous mechanical power, I am computing it as:
Power = - K3 x ControlForce. x AngularVelocity;

but if I compared this result with the power computed in the userDefinedFunctions script:
force = - output.ptos.forceActuation(ii:end,3);
vel = output.ptos.velocity(ii:end,3);
power = force. x vel;
the shape is reversed, but the worst thing is that magnitudes are very different.



Could someone give me a shed of light on what I am doing wrong?

Cheers,

Juan Guerrero

nathanmtom self-assigned this on Mar 3

kmruehl added WEC-Sim Application Support wecccomp and removed WEC-Sim Application labels on Mar 3

nathanmtom commented on Mar 4

@Liam-Guerrero Thanks for your inquiry, given my unfamiliarity with the state space model you are referencing it may take some time to understand your request.

Quickly, I would use the publication *Numerical Model Development and Validation for the WECCCOMP Control Competition* **here** in replace of *A competition for WEC control systems* to get the most accurate model parameters used in generating the WECCCOMP model under the WEC-Sim repository.

I hope to be able to better respond to your question in the near future.

Cheers,
Nathan
National Renewable Energy Laboratory

kmruehl moved this from **To do** to **In progress** in Issues on Mar 10

nathanmtom commented on Mar 11

@Liam-Guerrero Thank you for your patience! I was able to dive into your code a bit to grapple with your unique situation.

Based on reproducing your results from the .zip you attached, it would seem that you are not reproducing the same PTO force and direction in WEC-Sim compared to your state space model (SS). In WEC-Sim, we are actually calculating a force from the linear actuator and using the multibody dynamics solver generate the moment about Point A in the WECCCOMP set-up. Here you are implementing a control moment into you SS model, where your K3 constant is converting the linear force to a moment to be input into your SS model. I see that your K3 term is using some trigonometry to provide a static moment arm and without completing the derivation myself will assume you have done this correctly.

Therefore, it appears you would want to confirm that the PTO moment you are calculating and implementing in your SS model is sufficiently close to the WEC-Sim PTO moment about Point A.

I'd suggest changing the wave environment to `noWaveCIC` so we eliminate the wave excitation, which you seem to be calculating correctly given the fairly good match in float response when a PTO force isn't applied.

Then in WEC-Sim adjust the PTO force calculation to be a constant. When you run WEC-Sim the float should then rotate until the moment generated by the PTO is balanced by the other hydrostatic and hydrodynamic forces. You can then run your SS model, with the same constant moment, to see if you are getting the same direction and magnitude response of the float. This process should help you confirm your K3 calculation is correct and your input direction (+ or -) in the SS model is correct.

I hope my suggestions above lead you in the right direction to help further troubleshoot your model.

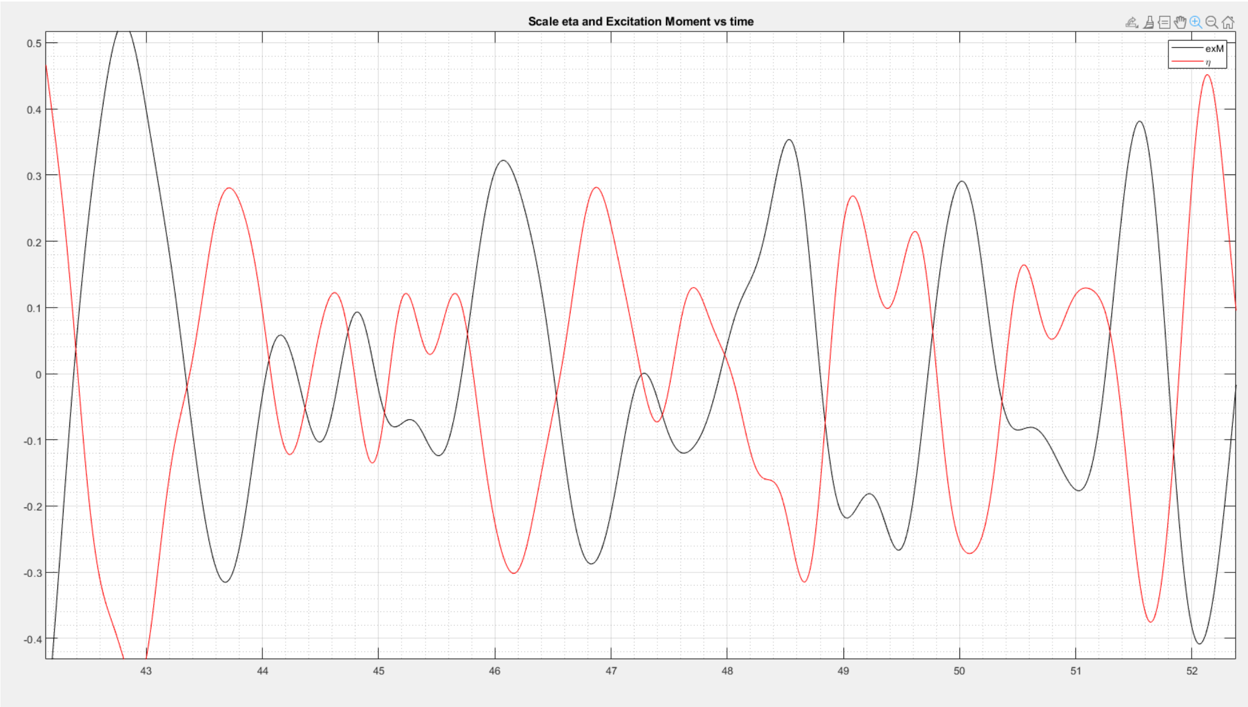
Let me know if you have any questions or comments.

Cheers,
Nathan
National Renewable Energy Laboratory

Liam-Guerrero commented on Mar 16 • edited

Hi Nathan,
Thanks for your following up on this issue.
See, you were right, I had a couple of problems with the SS-model.

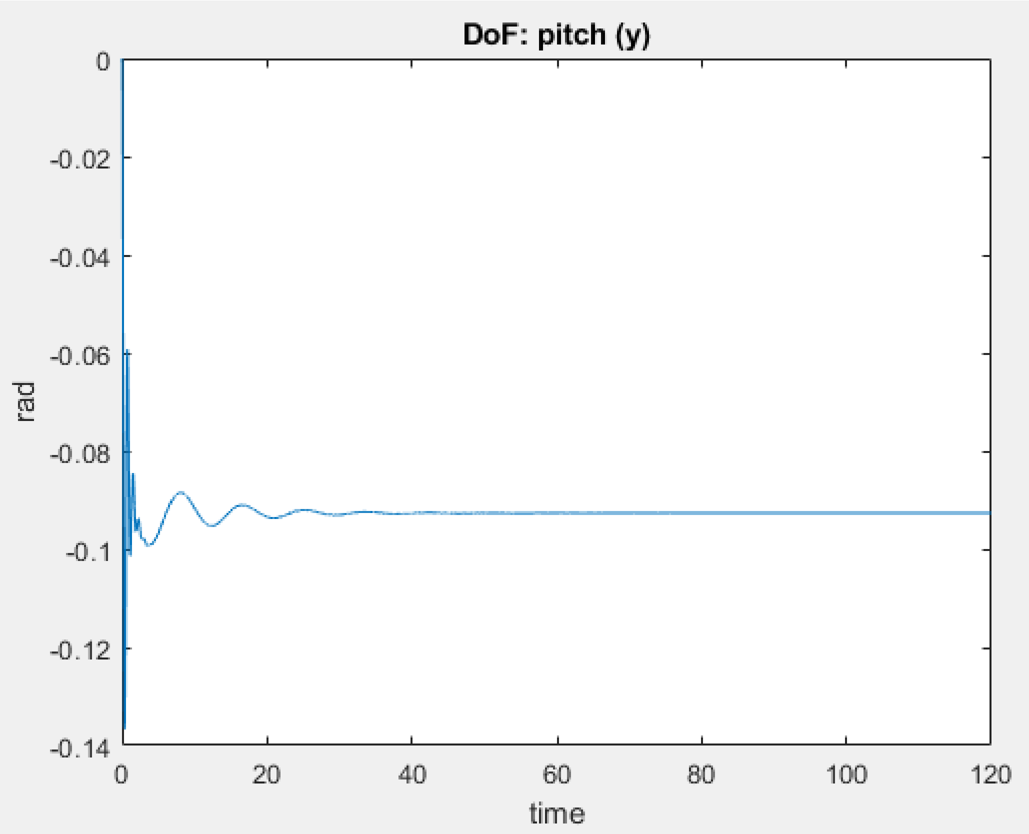
negative moment.



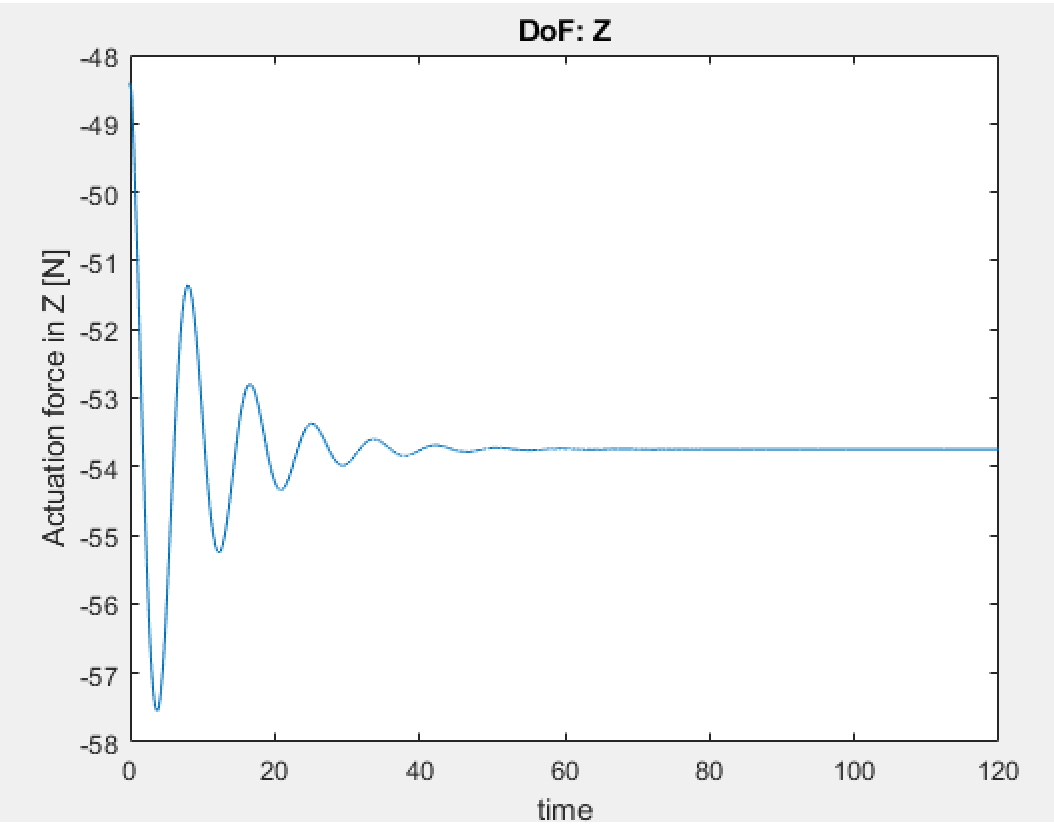
2. Having fixed that issue, I found that some signs were wrong on the ss-model.

3. As suggested, I ran the simulation with no waves and fixed force/moment, but now I have some issues with those results.

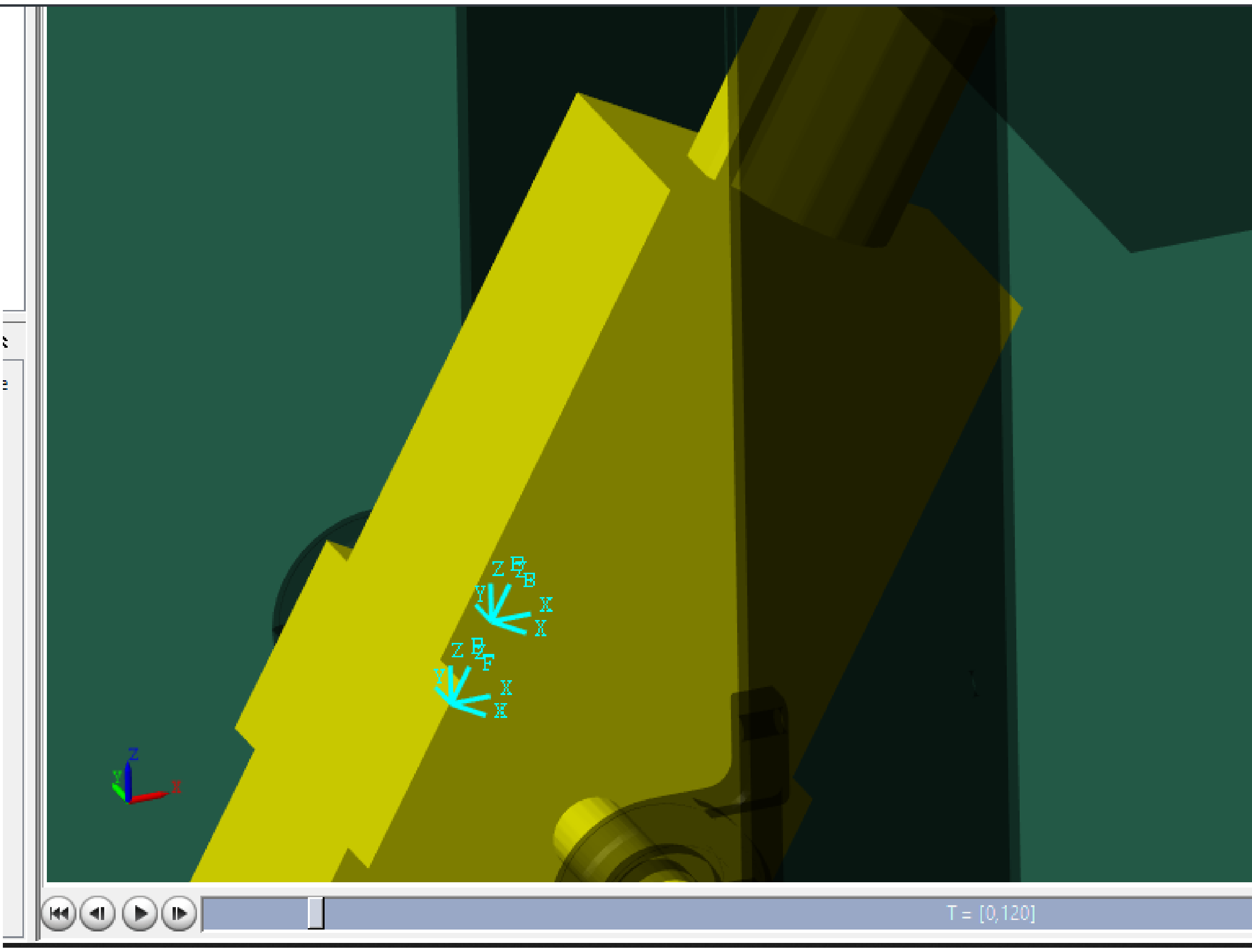
up, negative rotation.



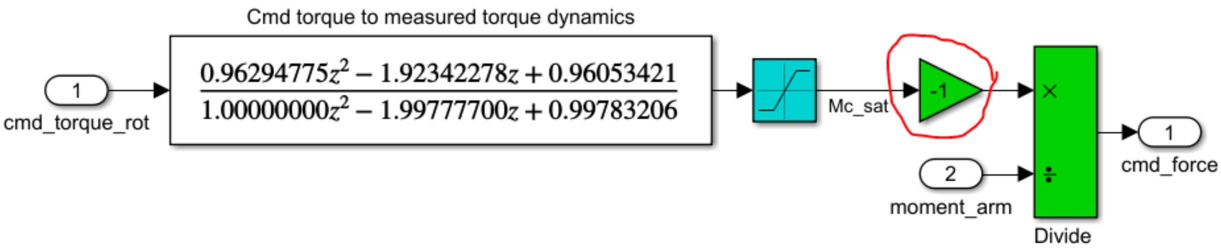
And I see that for a positive moment, I get a negative actuation force.

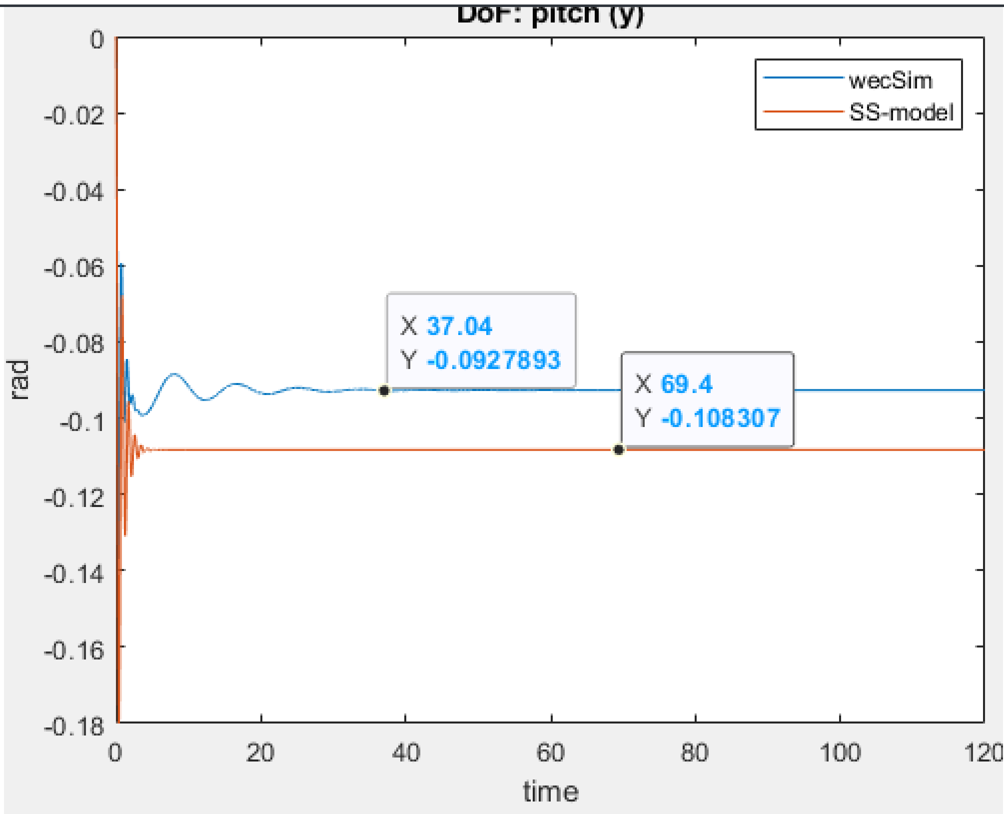


This part is a bit confusing for me because If I check the orientation of the linear motor on the mechanic's explorer, a positive value for force in the Z direction will produce a positive moment around hinge A

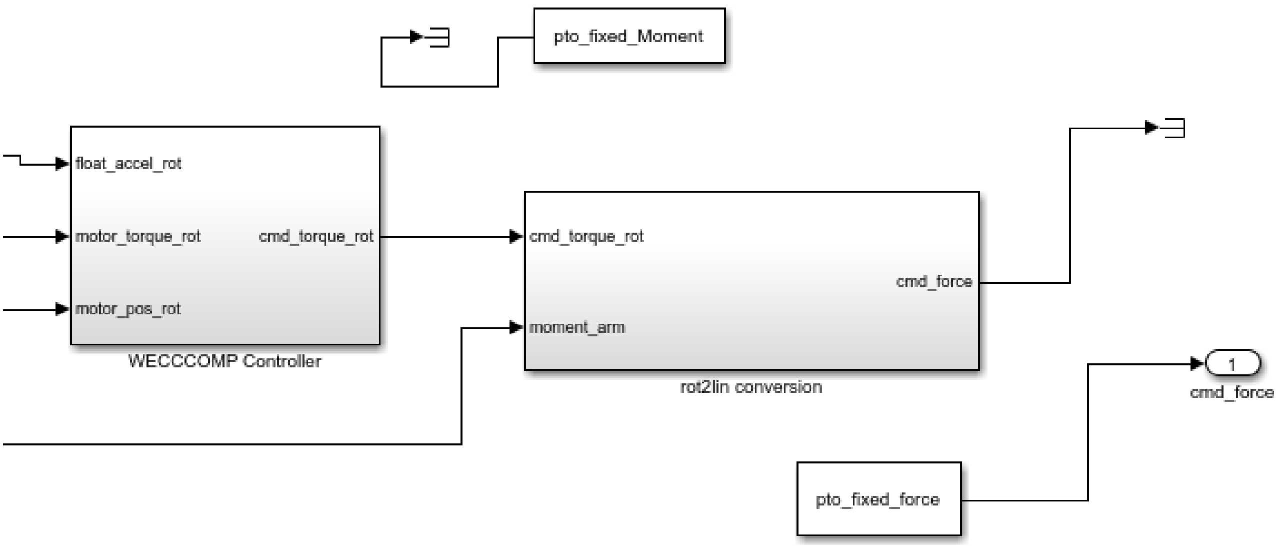


So I checked the Simulink file and found that there is a minus block that changes the sign of the cmd_torque_rot. meaning the cmd_force will always have a different sign from the cmd_torque_rot. This part is not clear to me why?



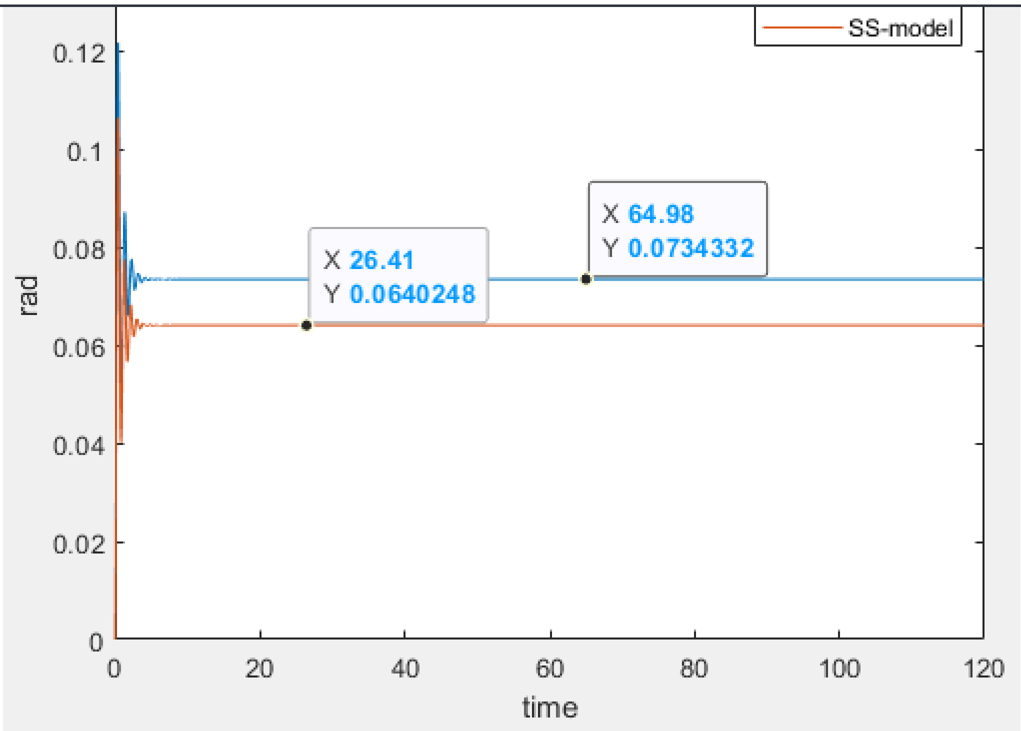


b) I did a similar simulation, but having now a fixed cmd_force.



This time the float went down, as I was expecting, so positive rotation for the arm.

Now, if I use that force and the moment_arm from wecsim for the equilibrium position in the ss-model, I get similar values for the angular displacement as long as I change the sign of the moment.



In summary, the difference I found so far between the wecSim and ss-model is that difference in the sign for force/moment. (I am aware wecsim would offer a better approximation for the real system), I mean for the ss-model a positive force in the actuator produce a positive moment around A, but in wecSim, It looks it is changing the sign.

4. Finally, yes, as the moment is a function of the angular displacement of the arm, I wanted to keep it simple for this first simulation and I am using an average value for the "moment arm" considering the max/min physical constraints.

$$M_{pto} = \frac{\overline{\mathrm{AB}} \cdot \overline{\mathrm{AC}} \cdot \sin(\alpha_0 + \theta)}{\sqrt{\overline{\mathrm{AB}}^2 + \overline{\mathrm{AC}}^2 - 2 \cdot \overline{\mathrm{AB}} \cdot \overline{\mathrm{AC}} \cdot \cos(\alpha_0 + \theta)}} \cdot u_k = f(\theta) \cdot u_k$$

K3

using a value of K3 = 0.19893.

Sorry for the long comments.

Once more, thanks for your insights.

Juan Guerrero

nathanmtom commented on Mar 17

In response to your question about the negative sign in front of the `cmd_force' for this initial model we have the initialization such that the user can define essentially a PID controller by setting mc, cc, and kc. Assuming the basic damping system, if cc is positive and we multiply by the velocity of the PTO extension we will need to multiply by -1 to get a resistive force proportional to the damping coefficient times velocity. If the torque and rotational velocity are in the same direction then we are motoring the system and injecting energy back into the system.

When you added your own static force then you could remove the -1 to keep consistent with the desired coordinate system as you mentioned above. In the competition, competitors could manipulate the controller mask to fit there needs and convert from their coordinate system to the WEC-Sim coordinate system.

I believe there are no longer any questions in your last comment so I am going to go ahead and close this issue. In case I have missed something in this last exchange please feel free to reopen the issue.

nathanmtom closed this on Mar 17

Issues automation moved this from **In progress** to **Done** on Mar 17

Assignees

nathanmtom

Labels

Support wecccomp

Projects

Issues
Done

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

3 participants