# **Weekly Report – W15 Fall 2022**

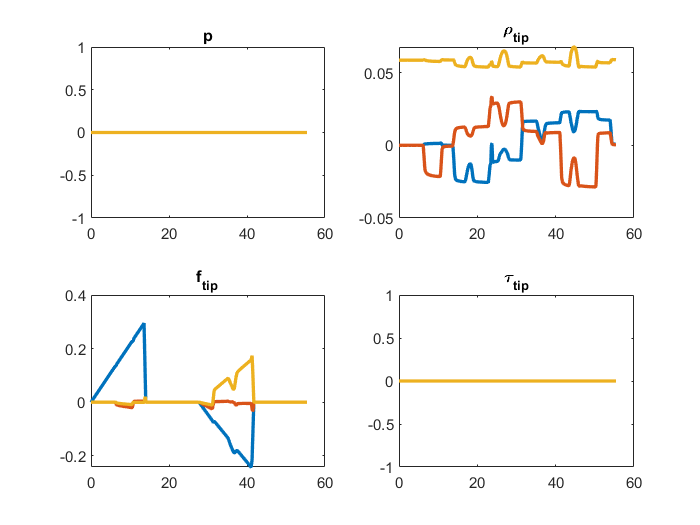
## **Problem & Task**

1. Check if the simulation is a dynamic process, since to trigger the ODE solver, the initial conditions are the prerequisites, so we can drag the tip of the SRA with a small bending angle (change the initial released position) with respect to the z axis to see if the arm will swing back to the straight state (along z axis);
2. The simulation results (animation) with SRL, EBA and EBR methods look like multiple discrete segments or links jointed together, with the increasing of number of segments or links, the accuracy of simulation results will be improved and the animation will more look like a continuum soft robot. However, we need to make sure how each segment was modelled, using lumped method or variable curvature? If they were modelled separately and individually, how they were connected together, what will be the interaction force between each two? Or if not, they were treated as a single body?
3. Another way to identify the modelling method stated in Problem 2 above is to check the definition of each state variables (or velocities, accelerations). For example, if the SRA was composed of 4 segments, and the degree of freedom (DoF) of each segment is defined as follows: the bending angle respect to x and y axis, and ; the twist angle respect to z axis , there will be 12 state variables to record the position of each joint (or the COM of each segment), these are also vital for us to know the system state for each time step instant, so we need to think about a proper way to export these intermediate variables and visualize them;
4. Try if we can add another ball to the tip (force sensor) of the manipulator, temporarily we can assume that the ball is rigidly linked with the tip.

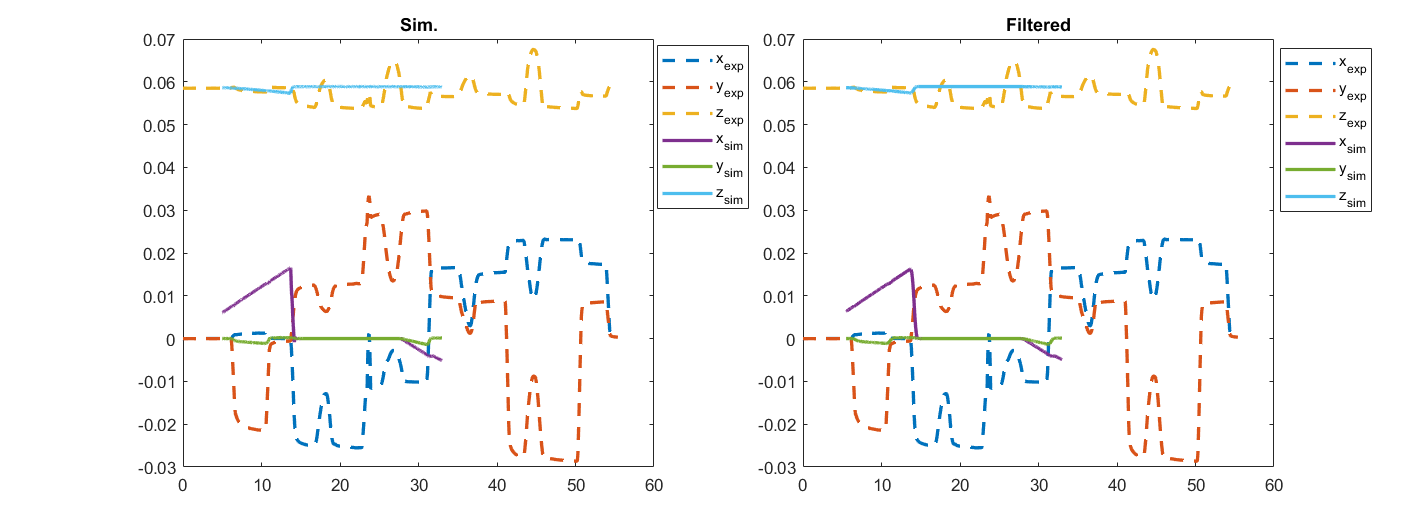
## **Solution**

1. *Dynamic process verification*

To achieve the same effect as changing the initial position of the SRA tip, we can either choose to modify the input external force acting on the tip, then eliminate the force suddenly without any phase lag; more detailed information for identifying the influence on each input parameter can be seen in Section 5. Additional tests. Fig. W15-1 illustrates the process of external force exerting on the tip, gradually increasing from zero to the maximum, then being cancelled like a step input, from the animation we can see that the bended SRA will swing back to the equilibrium position even without pressure infilled.



**Fig. W15-1** The designed force input for dynamic process verification

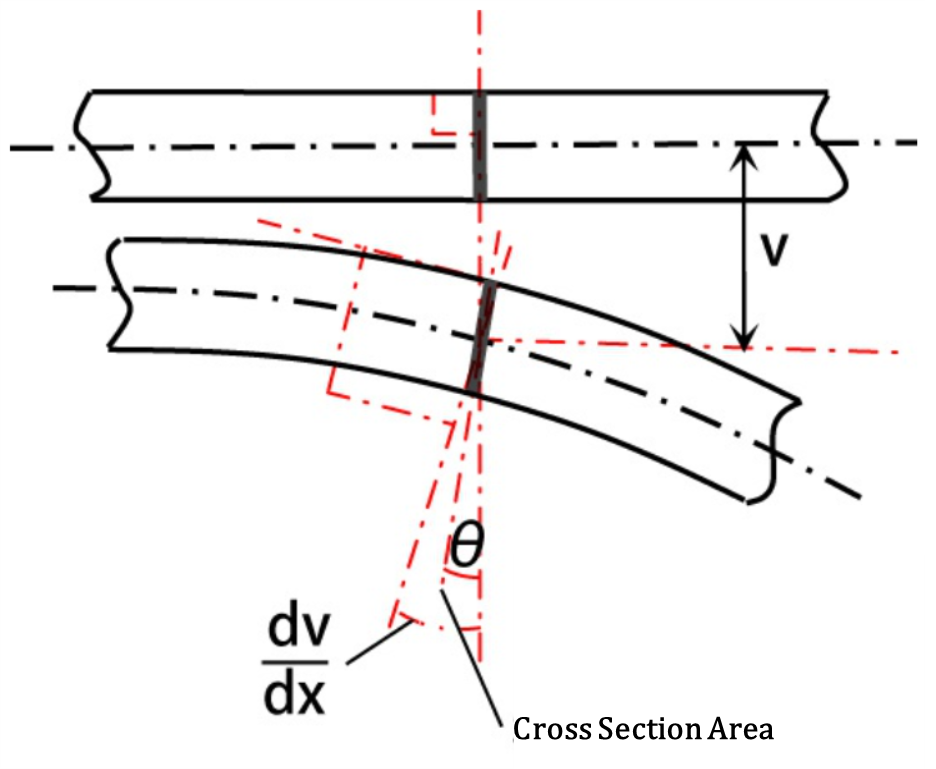


**Fig. W15-2** The simulation result of position change of the SRA tip given the input above

1. *The connection of each adjacent link/segment*

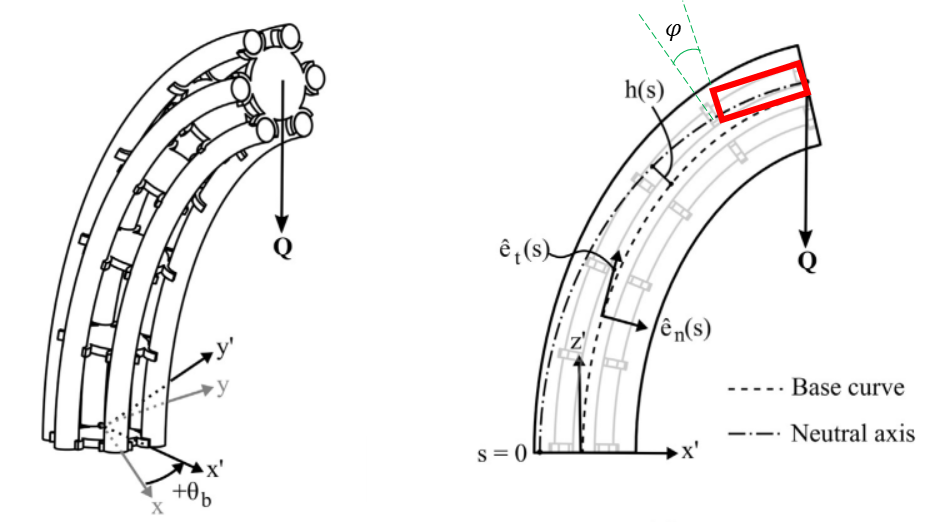
As EBR and EBA modelling are using the same theory, the Euler Bernoulli’s Beam, to clarify how each segment was connected with each other, we have to investigate the principle of the theory itself first. The basic principles are as follows,

* The theory complies with small deformation, linear elasticity, isotropy materials;
* When the beam deforms due to external forces, the cross section area will always be planar and perpendicular to the neutral line/axis.



**Fig. W15-3** The schematic of Euler Bernoulli Beam Theory

Thus that is the reason why EBR and EBA are slightly different, we need to find a reference plane to identify the rotational angle shown in the figure above for EBR (R means relative states) method, and using the figure (Gina, et al., 2020) below could better illustrate the their connection,



**Fig. W15-4** The schematic of ideal Euler Bernoulli’s beam deformation

So from the theory itself, we can judge that each segment/link is connected with others rigidly, in the simulation animation, the SRA performed like a discrete model with an obviously gap in the joint portion, however, it is continuous though it looked like modelling by constant curvature theory. The real reason for being discrete in the animation is that there is a “tubeplot” function stored in the “tmtdyn” file, which is used to animate the SRA with multiple tubes or straight cylinders; since the cylinders cannot be bended freely like the beam in the figure above, an angle (for absolute states, the angle should be the one between two adjacent cross section surface; for relative states, the angle should be measured from a reference plane) needs to be forged to represent the deformation. In summary, all the links/segments are connected with each other rigidly, for the convenience of visualization, they have to behave like a discrete model, but actually no matter for EBA or EBR, they are all continuous modelling methods.

1. *Intermediate variables exporting*

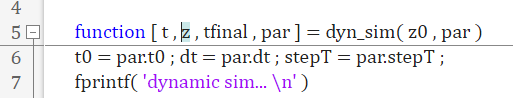
For a most simple second order system as follows,

where is the mass, is the damping coefficient and is the stiffness coefficient of the system, to identify its state on each simulation time step, we have to calculate , and respectively, which should be the acceleration, velocity and position. To solve such kind of ODEs, no matter for what kind of ODE solvers, the basic rule for them is to use numerical integration method to obtain a very close value to the theoretical one controlled by tolerance error and step size, so there must be somewhere used to stored these variables in the “TMTDyn” package. On the other hand, to animate the simulation process, we have to know the state (the position change at each joint at least) for each time step to update each frame, that is to say such file to store these intermediate variables does exist.

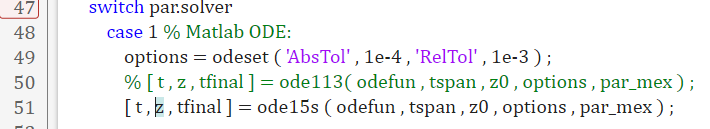
Starting from the animation function, we find that there are three inputs for this function, , and , which stand for the simulation time, results (intermediate variables) and parameter settings respectively. Tracing those inputs, we can find the more inside layer of the package, the “dyn\_sim” function with outputs are , , and , after checking the code logic, it can be confirmed that is the intermediate variables stored, as it is the output of ode15s, which can be seen in the following series of figures.



**Fig. W15-5** The inputs and outputs of function anim

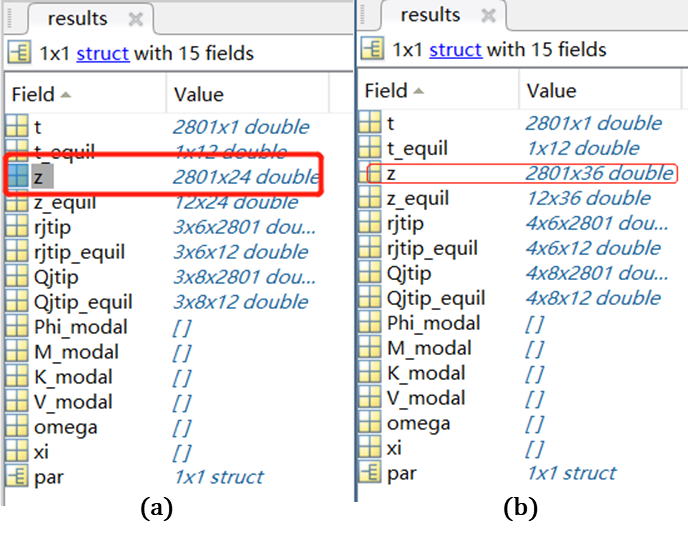


**Fig. W15-6** The inputs and outputs of function dyn\_sim

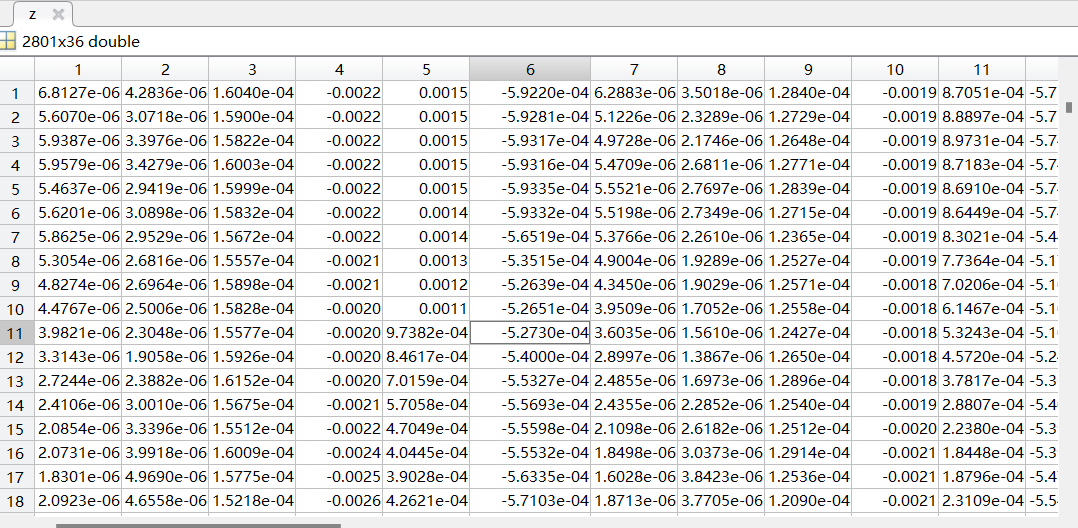


**Fig. W15-7** The direct proof of being the intermediate variables as the output of ODE solver

Since we have already known where we can find the intermediate variables, the next thing for us to do is to check how the number of links/segments affect the size of the output variables. We still use the EBR model, the only parameter change is the number of links, from the figure below we can see that the size of for and are and respectively, which means that for each link we add, there will be another 12 variables emerged, if we assume for each degree of freedom (DoF), there will be three variables accordingly, acceleration, velocity and position, in this case, each link will have 4 DoF. Temporarily we cannot specify what each column of data stands for from their magnitudes as shown in the second figure below, I still need to spend some more time working on it.



**Fig. W15-8** The comparison between the size of output variables by setting different numbers of links/segments of SRA. (a) , (b) .



**Fig. W15-9** The example of intermediate variables for

1. *Add a ball on the tip*

This task is pending, but since we have already found where the intermediate variables were stored and where the animation function was located, it will not be so hard to add another object in the original animation.

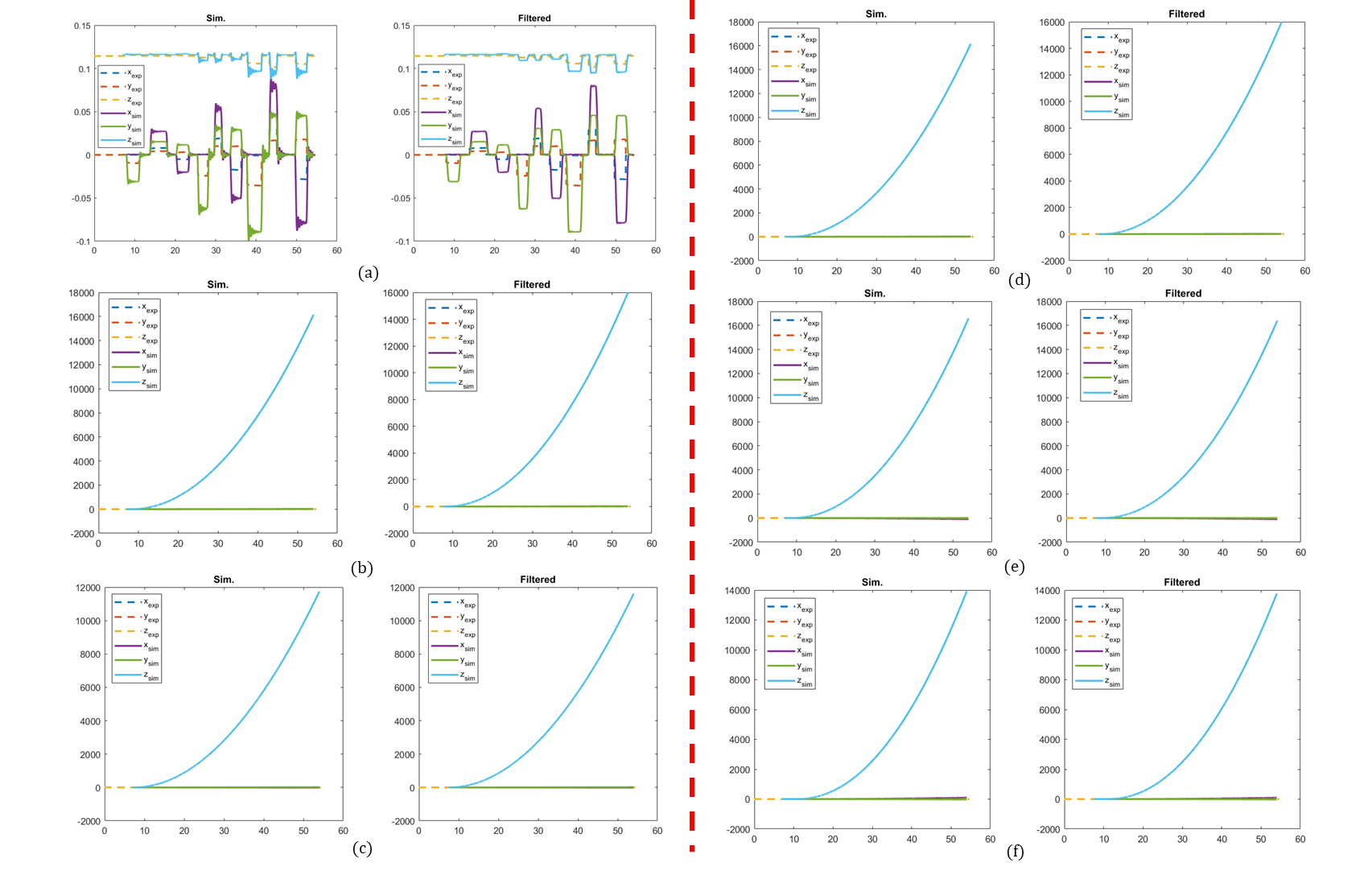
1. *Additional tests*

(1). The influence of manipulator parameters (length) on simulation results

Last week I found that when I set the length of the manipulator as 50 meters, the package could run with no errors after fixing the redundant “end”s problem, but the animation looked not so good, and the magnitude of displacement along each axis can be up to m, which was apparently not correct, therefore this week I need to explore more about aspect.

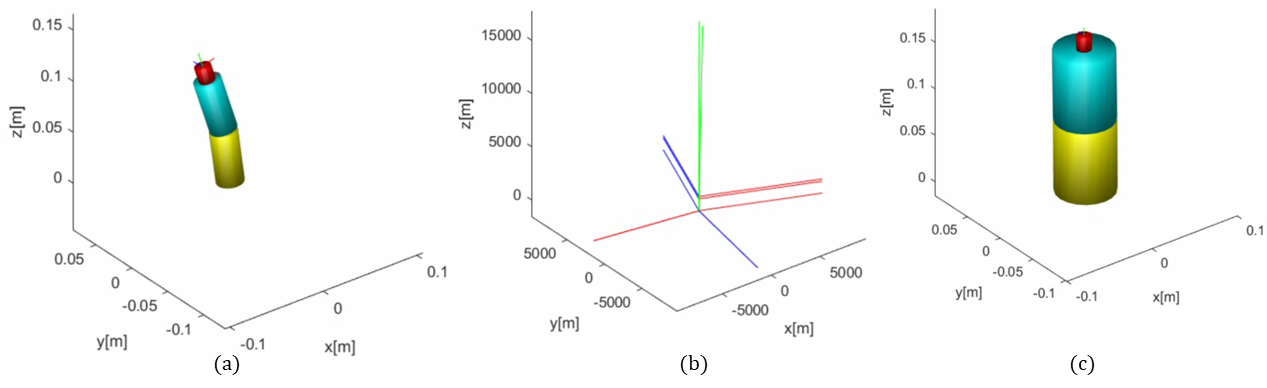
As our physical SRA has two links, the length of each is about 30 cm, the total length should be around 60 cm, which is about 15 times by the length of the manipulator stated in the Sadati’s paper, and the package seemed not to have a good performance for very long manipulator, it’s necessary for us to do more tests.

My first control test group was to increase the length of the manipulator from 100 mm to 200 mm, since I found that when the length was set to be 150 mm, the animation and the position of the tip seemed not good, here comes the second test group to verify the influence of number of links/segments on the simulation results, the number was set to be 2, 3 and 4 respectively; and for the third one, since issue began to occur for length equal to 150 mm, I decided to modify the outer radius of the manipulator to see if the animation is normal. And all the results are shown below.



**Fig. W15-10** The simulation results from “TMTDyn” package based on different manipulator lengths and numbers of links/segments, left side: increasing length for number of links with 2; right side: increasing number of links with manipulator length with 150 mm. (a) 100 mm, (b) 150 mm, (c) 200 mm; (d) 2, (e) , (d) .

From the simulation results above, we can clearly find that for the test of increasing length for the number of links with 2 (on the left side isolated by the red dash line in the middle), the position of the tip began to be abnormal when the length is greater than 150 mm, and the unit for y axis for these plots are all in meter, so obviously the results are not rational. Then I thought if the number of links will affect the accuracy and results of the simulation entirely, to this end, we found that indeed the increasing number of links will improve the accuracy (from about 16000 m max to about 14000 m max) sacrificing a lot of computation time, however it did not have any decisive effect on the final trend.



**Fig. W15-11** The simulation animation from “TMTDym” package based on changing the outer radius of the manipulator when s. (a) mm, 12.5 mm, (b) mm, 12.5 mm, (c) mm, 30 mm.

And to investigate what other parameters of the manipulator determine the simulation results, I decided to change the outer radius of the manipulator for the specific length whose simulation results (position of the tip) seemed quite abnormal, from the figure above, it’s not difficult to draw that except for the length of the manipulator, if we keep all the other parameter the same, the results went crack for length 150 mm, however, when we modified the outer radius of it from 12.5mm to 30 mm, the simulation animation went back into control again even though the position variation was not so obvious, partially it was owing to the enlarged stiffness of it by increasing the radius (more material infilled).

So far I suppose we can draw a very rough conclusion based on the simulation results from the two control test groups, which will be listed as follows,

* The length of the manipulator will affect the simulation results largely, however, it is not the limitation or fault of the package itself to cause this kind of failure, controlling the ration between the length and outer radius of the manipulator will lead to a good simulation result in most cases;
* The number of links/segments will affect the accuracy of simulation, the basic principle behind is like FEM, but when it exceeds a certain value, the improvement will not be so apparent any more, it will become meaningless to sacrifice so much computation efficiency then;
* This one is more like a hypothesis rather than a conclusion, which needs to be proved in the near future and has much relevance to the Problem 1. I think the main reason for the failure of most simulations mentioned above can be owing to the geometry factor of the manipulator itself, the manipulators in all the failed cases were deemed to be very “thin”, they would easily fall down on the ground even subjected to a minor disturbance at the equilibrium point. The additional test by increasing the outer radius of the manipulator can successfully support this standing, when the radius is becoming greater and greater, it will gradually become a “cylinder” from “stick”, it will be stronger to support its own gravity and easier to swing back to the equilibrium point after the force/torque acting on the tip is eliminated. Another way to testify it is to enlarge the pressure input in all three chambers for the very “thin” manipulators, and see if they can acquire better simulation results.

(2). The influence of pressure change in the chamber on simulation results

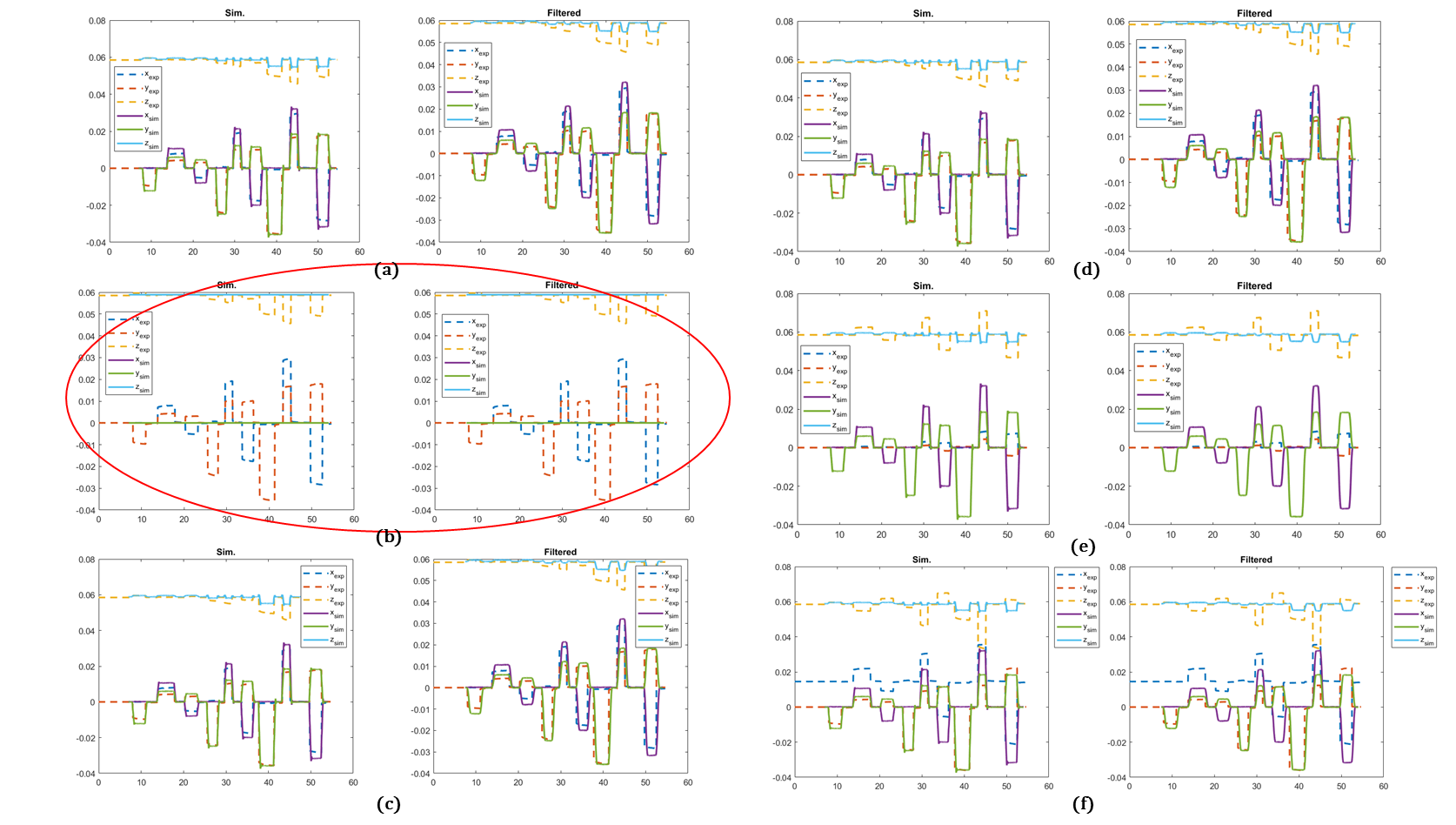
According to the hypothesis proposed previously, if we increase the pressure in certain chamber(s), and the SRA would swing back, it can be proved that the length setting of the model is not the limitation of the package. However, before doing that we have to make sure the meaning of data of each column in “NoLoad.mat” file, from the comments in the “sample\_exp\_data.m” file for the input mode 1, we have the following definitions.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| Time | Pres0 | Pres1 | Pres2 | X\_val | Y\_val | Z\_val | Q0\_val | Qx\_val | Qy\_val | Qz\_val | X1\_val | Y1\_val | Z1\_val | Q01\_val | Qx1\_val | Qy1\_val | Qz1\_val |

**Table W15-1.** The definitions of the inputs for mode 1

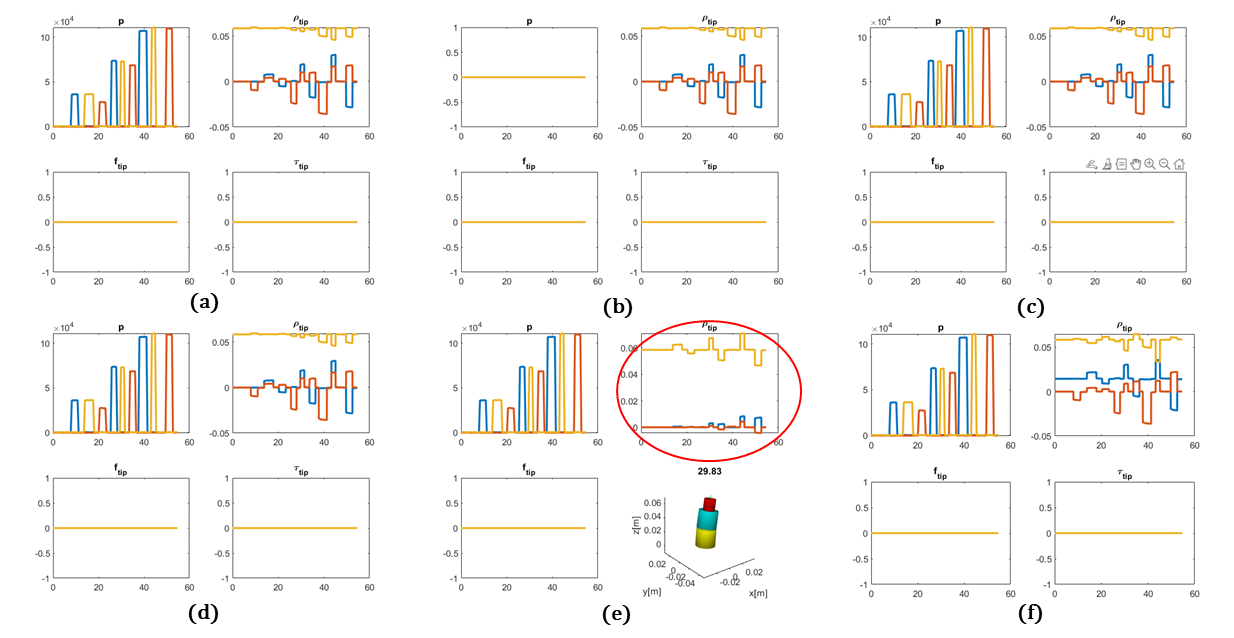
Still using the control test method, we find that the names of the columns comply with some basic principles, for example, the prefix of the column 2 to 4 is Pres, which probably means the pressure supply for each pneumatic chamber for the SRA (needs to be further confirmed), to this end, our secondary experiment can be classified into 6 groups, the original group, zero elements (input) for column 2 to 4, zero elements (input) for 5 to 7, from 8 to 11, from 12 to 14 and from 15 to 18 respectively. My plan was to set all the elements of the columns mentioned for each test group as zero, on one hand we can figure out which parameters are the real inputs to trigger the simulation; on the other hand, we can check if the pressure supply domains the dynamics of the SRA by multiplying it in the subsequent tests.

The simulation results for the six tests have been showcased as follows, in which the dash lines represent the experimental results and the solid lines stand for simulation results, except for (b), the simulation results are exactly the same, the most likely reason is the input changed in group (b) is the real input of the simulation, which is pressure; for the rest figures, the experimental results vary for each one, which seemed pretty weird because we have just confirmed the pressure is the only input and real experiment data would not be affected by simulation input. To find the reason behind,



**Fig. W15-12** The comparison of tip position of physical experiments and simulation results. (a) original without any parameter or input change, (b) the elements from 2 to 4 were set to be zero, (c) the elements from 5 to 7 were set to be zero, (d) the elements from 8 to 11 were set to be zero, (e) the elements from 12 to 14 were set to be zero, (f) the elements from 14 to 18 were set to be zero.

we probably need to check the specific input in the next figure below. Still in the figure (b), we can clearly find that no matter for the pressure of each chamber, the force acting on the tip or the torque on the tip, they are all zero, but the position of the tip () is not always zero, so there is only one chance that it should be the position record of the real experiment. And the position of the tip varies with the same principle for the first four figures, in (e) and (f), the position performed with a totally different manner, the summation of (e) and (f) coincides with the other four, maybe there are some system offsets or filters in the last seven columns of input data, we can study it in the future, as it is not the main focus of our research right now.



**Fig. W15-13** The specific input of the simulation for 6 groups of tests mentioned above. (a) original without any parameter or input change, (b) the elements from 2 to 4 were set to be zero, (c) the elements from 5 to 7 were set to be zero, (d) the elements from 8 to 11 were set to be zero, (e) the elements from 12 to 14 were set to be zero, (f) the elements from 14 to 18 were set to be zero.

## **Plan**

1. Still test the performance of the package on simulation of very long SRA, the pressure of each chamber can be set equal, then evaluate the simulation results with the same amount of force acting on the SRA tip by multiplying different times of the pressure compressed;
2. Continue to think about how to add a ball on the tip.

## **Reference**

[1] Olson, G. *et al.* (2020) “An euler–bernoulli beam model for soft robot arms bent through self-stress and external loads,” *International Journal of Solids and Structures*, 207, pp. 113–131. Available at: <https://doi.org/10.1016/j.ijsolstr.2020.09.015>.