# **Weekly Report – W0 Spring 2023**

## **Problem & Task**

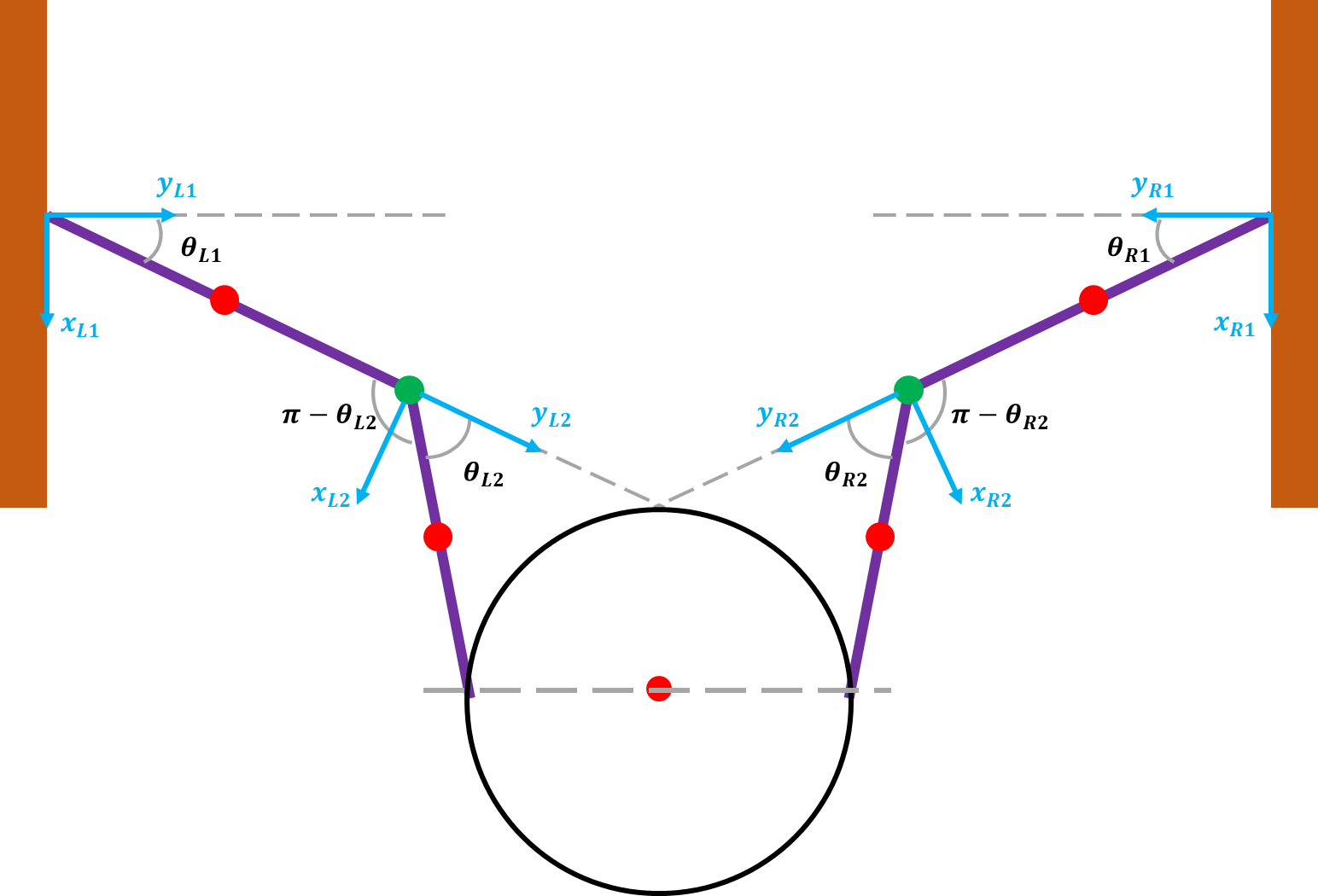
1. In our “Falling SRA V2” code, we treat the left and right arm, the ball as three individual objects, which made it hard to add the external force exerted on the SRA tip (no matter for the left or the right one), but the contemporary modelling method is beneficial for more complex cases in the future. Since Now we are considering the simplest case, the ball is released from an arbitrary position along the central line between the two walls, all the motions of the arms are symmetric about the central line as well, we can temporarily take them three (two arms and the ball) as a single object in this case the interaction force between the arm and the ball can be ignored;
2. Last week, I have tried two of the possible directions that might solve the problem that the different segments separate from each other in the animation, the Young’s Modulus has been proved to have a large impact on it; for another direction, I planned to change the size of the pneumatic chamber position matrix from to and the input pressure from to according to Dr. Sadati’s suggestion, but it failed due to other function size problems, therefore, this week we can try to eliminate the offset angles to make each two chambers in pairs now coincide with each other to see the response;
3. Do other simulation on the TMTDyn package in order to find proper range of SRA diameter-length-ratio which can lead to a “healthy” simulation result.

## **Solution**

1. “Falling SRA V2” project

As derived in the last week’s report, the governing equations of the system can be summarized by the following ones (left and right respectively),

Since it’s hard to compute the external force exerted on the left and right arms respectively, especially when the geometry and material properties are different for the two arms, we temporarily can treat all the motions are symmetric about the central line between the two walls, and treat all the components as a whole system (including the ball) so that the external forces can be eliminated and all of them will become interaction forces.



**Fig. W0-1** The schematic of the whole system

For normal case, the Eqn. (W0-1) and (W0-2) can be combined as one single equation with four variables as follows,

which can be simplified as

We can keep this version for future use, however, for simplified version, since , and , for the second segment, we have the similar relationship, , and , thus the dimension of Eqn. (W0-4) can be decreased to as follows,

where is the potential energy part of the ball, which should be equal to , is the real potential energy of the ball, , thus

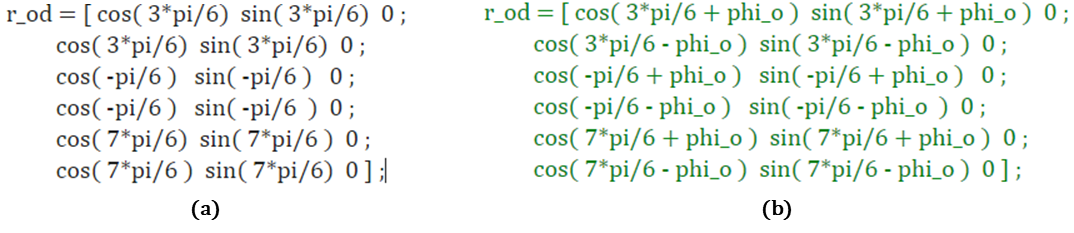
according to the computation of MATLAB. Due to the symmetric property of the special case, all the terms in Eqn. (W0-5) and (W0-6) can be simplified as

where .

1. Update about TMTDyn package

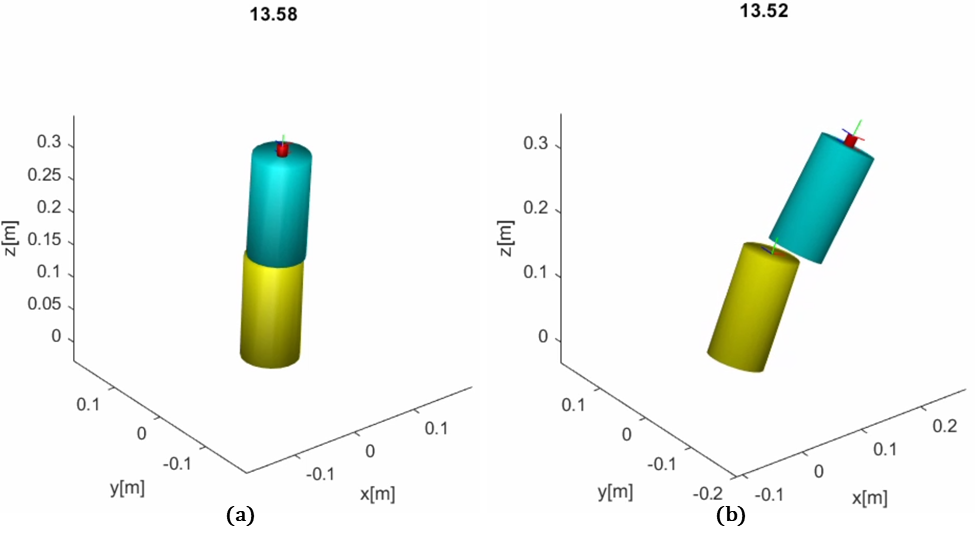
#### (1). Adjust the number of chambers from 6 to 3

By eliminating the offset angle phi\_o in the original chamber position matrix, which has been shown below,



**Fig. W0-2** The SRA pneumatic chamber position matrices. (a) The modified matrix, (b) The original matrix.

To test the performance of the animation, we still apply 0.5 N force along x direction on the SRA tip without any other pressure input, the package could successfully run without error and the separating problems didn’t occur for this situation as shown in the figure (a) below. However, when I increased the force input to 5N as shown in (b), the deformation is about the maximum at about simulation time 13.5 s, and we found that the separating situation still exists for large shear force, but not that serious since we have set the Young’s Modulus four times as the original value, which also indicated that Young’s Modulus domains the performance of animation except for geometry factors of the SRA.



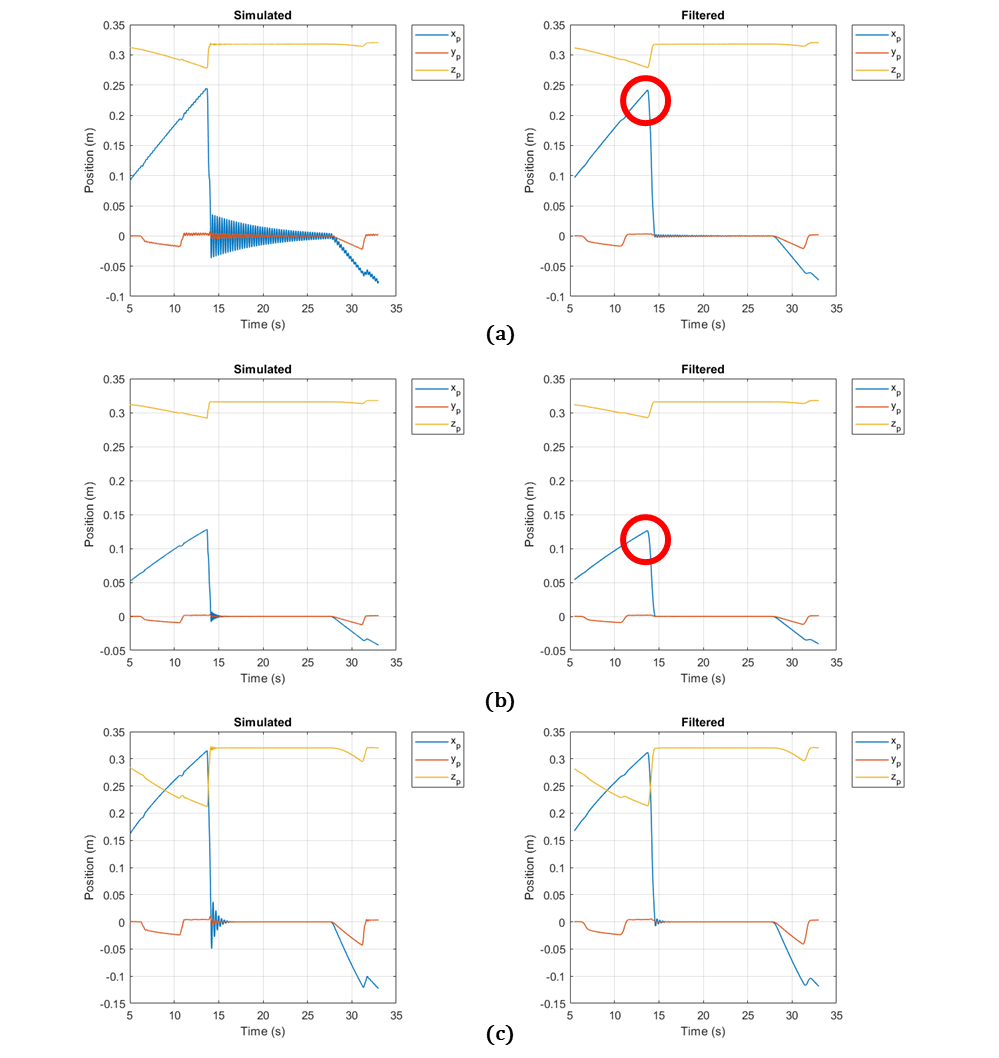
**Fig. W0-3** The animation screen shot at maximum deformation simulation instant 13.5 s with force exerted on the SRA tip in the x direction only. (a). input force: 0.5 N, (b) input force: 5 N.

So we can make a rough conclusion based on the results so far, the discrete modelling method (taking a whole SRA as several rigid links) like EBR and EBA might not be a proper way to show the properties of SRA deformation for large shear force, in the next section I will try other modelling method in the package to see if there would be any improvement for the animation.

#### (2). The effect comparison between EBR and ROM

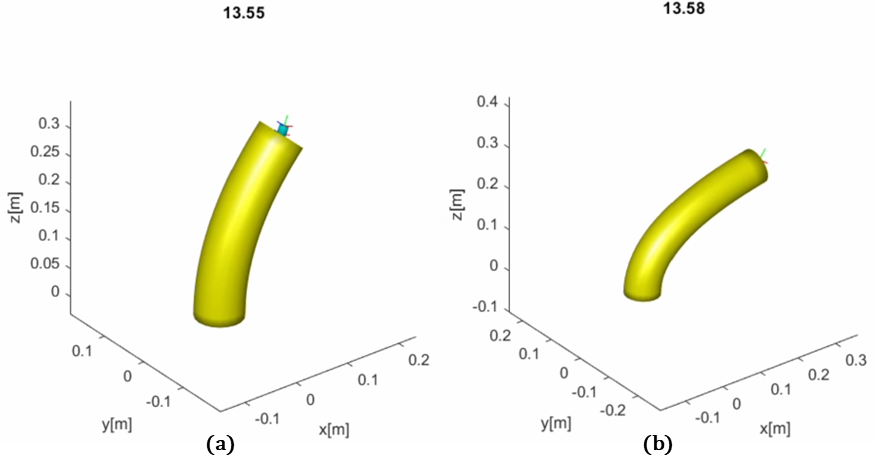
Since we have tried a lot based on the EBR modelling method, a conclusion can be made that EBR and EBA are only applicable for small deformation under small shear force (or any other forms of input), though the tip position result plot seems reasonable, the animation will cause a lot confusion to the audience and readers who are not familiar with this package. In this case, I began to explore other modelling methods in the package, for example, ROM, reduced order modelling method, which is a continuous modelling.

To compare the performance of different methods, I applied exactly the same conditions, 5N force exerted on the SRA tip with 4 times of original Young’s Modulus value (480 GPa), the results can be seen in the figure below. We can find that with the same input, the maximum displacement in x direction for EBR and ROM methods are about 0.25 m and 0.13 m respectively, which is a huge difference. So I looked up if there is any **relative explanation** in the author’s paper, it says for **static motion** (no external force on the tip) when the number of segment was set to be very small, like 1 or 2, the absolute and normalized error for ROM is smaller than EBR and EBA; however, with the increasing of number of segment, the accuracy of EBA and EBR will rapidly improve. But for the cases with **external loads**, the ROM modelling method remains the best accuracy among all the methods.



**Fig. W0-4** The SRA tip position results caused by different settings. (a) 5 N force exerted on the tip with 4 times of original Young’s Modulus by EBR modelling, (b) 5 N force exerted on the tip with 4 times of original Young’s Modulus by ROM modelling, (c) 5 N force exerted on the tip with default Young’s Modulus by ROM modelling.

Thus in conclusion, the result for ROM method is totally reliable, and to explore more about the performance of ROM, I changed the magnitude of Young’s Modulus since it domains the final simulation and the animation a lot, the force exerted on the tip was still 5 N maximum, the comparison of 4 times and 1 time Young’s Modulus can be seen in the figure below.



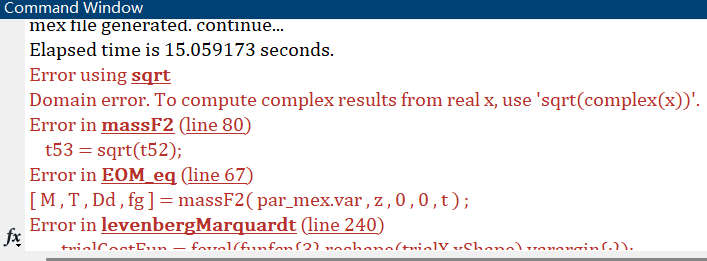
**Fig. W0-5** The animation comparison for input force 5 N maximum exerted on SRA tip with different values of Young’s Modulus. (a) , (b) .

From the screen shots of the animation and the simulation results in the previous figure (c), neither the simulation or the animation got crack, using ROM can perfectly solve the separating segment problem and meanwhile I think the “fake” animation for EBR is caused by the limitation of animation function itself (the tubeplot function), which cannot cope with elongation situation (just a guess). So maybe in the future, when dealing with large shear force or input, we can try to use ROM to avoid such issues.

1. Other tests

#### (1). Larger shear force for ROM

When I adjusted the input force from 5 N to 50 N, the error information emerged in the command window as shown in the figure below, just like we did for over 4 segments, there are still some problems with the derived equations of motion function, by correcting the code according to the indications in the command window 4 times, the simulation package will run without any errors and both the results and animation looked fine.



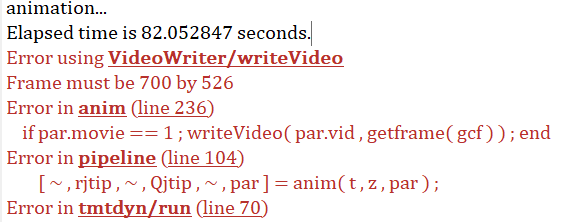
**Fig. W0-6** The screen shot of error emerged for large shear force setting for ROM

To test more about the property when the SRA suffers from larger shear force, I set the following tests as shown in the table,

**Table W0-1.** Additional tests for larger input force for ROM

|  |  |  |
| --- | --- | --- |
| Input force | Error emerged time | Simulation time |
| 10 N | 0 | 353.183484 s\* |
| 20 N | 4 | 203.061510 s |
| 30 N | 4 | 178.219929 s |
| 40 N | 5 | / |
| 100 N | 4 | 206.880327 s |

\*For 10 N input, since there is no error emerged, the EOMs derivation will take much more time, for the other tests, if we derive the EOMs for each time we run the package, the modified code will be rewritten again and again, thus the time will be shorter.



**Fig. W0-7** The accidental error emerged for 40 N input

Especially, when the force was set to be 40 N, although we have fixed the code for four times, the final error seems that there is still some bugs with the animation process, which haven’t figured out a way to deal with (I think I need to report it to the author), but the animation can be normally recorded even with error.

In summary, from the table above, we can make a rough conclusion, after a certain value of input, the increasing of the input force will not affect the time for error emerging any more (the error emerged in massF1.m, massF2.m, sprdmpF10.m and LoadsF1.m files), my guess is that associated with our past experience, the time error emerged should be relevant to the complexity (the number of segments setting) of the system and the input force given.

However, the most exciting part of the tests is that the simulation results and animation looked pretty reasonable even for 100 N input force, which is a good sign.

#### (2). Debugging for more segments in ROM

To verify the performance of ROM for multiple segments, we are going to start from 5 N input force, because the package can run without any modifications or errors. My test started from 4 segments (since I remember there will be some problems with EBR for 4 segments), then increase the force to 10 N to see if the time for error emerging will be increased (because the 4 segment simulation will take a long time, to save time temporarily we just test for 10 N and 5 N and then make a comparison), and the results are shown as follows.

**Table W0-2.** The error emerged time for large number segment setting with ROM modelling (5 N input)

|  |  |  |
| --- | --- | --- |
| No. of segment | Error emerged time | Simulation time |
| 4 | 0 | 1087.879439 s |
| 5 | 1  (due to animation process) | 5477.835278 s  (just animation) |

It can be seen that compared with EBR, there is no error emerged in the EOM derivation process, which indicated that ROM is much more stable than EBR; besides, in terms of the simulation time, I remember the time for EBR with 4 segments was about 3900 s, ROM is much more efficient. However, the animation problem emerged again just as shown in Fig. W0-7 when the number of segment was set to be 5 for ROM, I think it’s necessary for me to repeat the simulation and find the reason behind next week.

And for 10 N input with 4 segments, the error emerged time is still 4, which means it is only related to the magnitude of input force setting rather than the complexity of the system, nevertheless, I reckon I still need to do more tests for more segments simulation to verify this.

## **Plan**

1. Prepare relative literature review paper for our project in advance, and classify them into control and modelling alike;
2. Continue to do more tests on the package if needed, evaluate the results of our own project’s simulation and make corresponding amendments.