# **Weekly Report – W2 Spring 2023**

## **Problem & Task**

1. Schedule a more thorough test plan for ROM method simulation (e.g. applying force on the SRA tip in x, y, z directions and applying pressure in chamber 1, 2 and 3 respectively);
2. Think about how to add another SRA in the existing TMTDyn package, since in the future, we have to make this package as similar as possible to our project;
3. Think about how to add an additional object (could be a cube or sphere) into the TMTDyn package which could be useful for impact dynamics simulation;
4. Derive the impact dynamic process;
5. Apply a PD controller in the TMTDyn package to let the SRA tip gradually achieve the target position (could be the center point of the object in bullet point No. 3).

## **Solution**

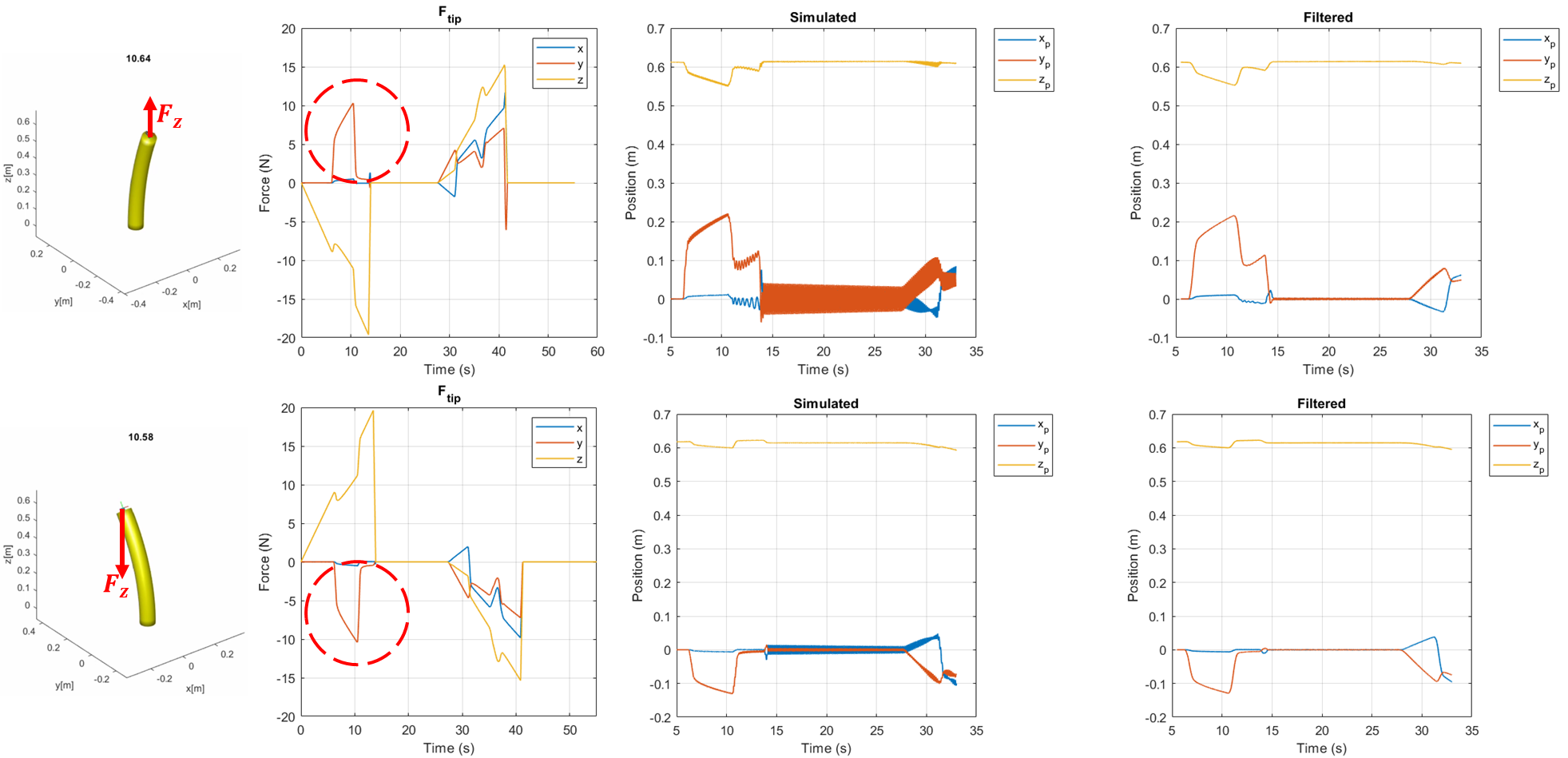
1. A more thorough test on ROM’s performance

As we have done quite a lot simulations based on EBR/EBA, to test the performance of ROM, the basic procedure is similar, to apply a single direction force exerted on SRA tip in x, y and z directions respectively, and for the pressure input, we can pump pressure gradually in the three chambers individually to see the response.

My plan is to design a linearly increasing force/pressure input, when the force/pressure achieves the maximum value, it will be removed immediately (like a step input) and let the SRA be in free motion for a period, then apply the same amount of input in the opposite direction and release the SRA.

(1). For force input, the maximum value is designed to be 10 N, 20 N, 30 N, 40 N, 50 N, 100 N, 200 N and 500 N for all the three directions, so there will be 8 groups of tests in total, here I would like to state that though we have already had a meeting and report the simulation results during the weekday this week, I think it’s still necessary to make some formal documentations for tracking in the future, more importantly, some of the key details haven’t been covered in the meeting.

* A rough conclusion can be made based on the simulation results that generally these datasets can prove that the ROM method is satisfied with our project’s requirements and all the results looked reasonable except for some minor issues;
* No matter for which kind of modelling method or what kind of parameters we set, there always will be some of the points that the package will fail (error emerged), for example, when I set the input force for x and y direction as 50 N, the simulation will abort automatically due to errors in the newly derived inertia matrix file and the external load file, after changing all the sqrt() into sqrt(complex()), the issue was solved, and later for even larger force input, there is no such error any more, the reason behind is that the shear force is relatively large; for the force applied on z axis, things become a little different, no matter for positive or negative z direction, there always existed an unexpected motion in y direction, I could understand that when the force is towards down to the ground, due to the nonuniformity of elastic/soft materials, the bending direction can be arbitrary, however, when the force in positive z direction, the SRA should be stretched only, there should not be any other motions in x or y direction, the SRA would be bended in y direction for just a short period, then back into rightly parallel with z axis; also the maximum force forged in y direction is always half of the maximum force applied in z direction, which also explained that when the maximum force input was set to be 100 N in z direction, the same error will emerge just like I did for x and y with 50 N input. I have reported this issue to the author along with any other problems I found about this package, I hope next week I will get some feedback;



**Fig. W2-1** The problem when force is applied on z direction

* When I did test for EBA/EBR, there were also some errors but due to large number of segment setting, I have never run the package again for those inputs didn’t cause any errors after amendments. So this time to be more rigorous, I decided to run the package again after modification for 10 N, 20 N, 30 N and 40 N force input additionally to find if it would affect the final results. After comparing the results before and after, I found simulation results for small shear forces will not be affected.

1. Adding another SRA in the 3D space

Though this target was not fulfilled yet, I think I have a rough idea about it which can be shown as follows,

* Since there are too many nest functions to modify, the first thing we have to clarify is what are they. Because the two arms are exactly the same, the EOMs will be the same as well, only the force/pressure input and the base position are different, the animation function cannot be called twice due to the unique structure of the package, I have tried for several times, apparently there are much more need to fix, so I also consulted help from the author as well;
* Next thing is to find where we can fix the base position of the arm, and then we can establish another SRA with different base position with the same amount of input;
* At last, we can make up another force/pressure input and change the size of the file to store the intermediate variables.

1. Add another object in an arbitrary position in the 3D space

Different from adding another SRA, because the cube temporarily does not need to contact or collide with other objects (SRAs), there will not be any coordinates updated according to the results from the ODE solver.

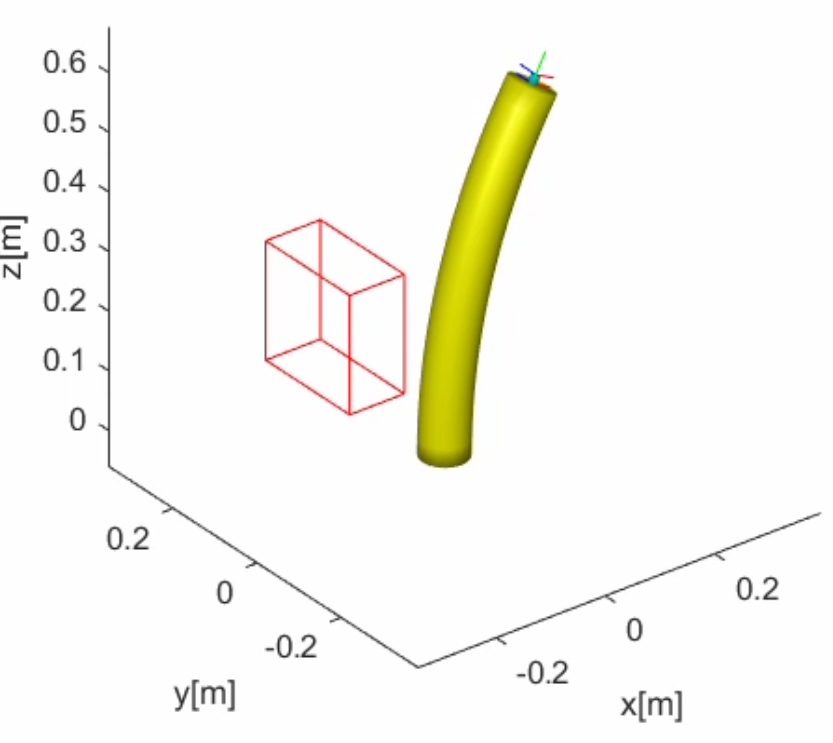


Fig. W2-2 The example of adding a cube in the 3D space

Currently I have no idea to draw a solid cube (just like the format of the SRA in the figure above), so I used the frame of the cube instead, meanwhile utilization of function patch will cause some unexpected effects, probably because patch and mesh cannot be plotted in the same frame, I have faced the same problem when I played with Chase’s simulation package (adding a pad at the base of SRA using patch).

Additionally, the cube is established about the center point, we can change the position easily as we desire in the future.

1. Derivation of impact dynamics

According to the principle of virtual work and conservation of moment of the impact point, we can work out the impact force based on the impact position and its velocities, so the first several things we need to do are shown as follows,

* In my understanding, the first thing is to confirm the exact position the SRA impacts with the object (could be the cube in Fig. W2-2), ROM is a continuous modelling method based on theories like Cosserat rod theory, the basic principle is that we can select several (the order number “O” in ROM) arbitrary points in the backbone of the SRA, by computing the states of these point, we can confirm the shape of the pipeline (backbone), temporarily I don’t know if there exists a specific expression for the pipeline curve, if it does, since it is surrounded by a continuous tube whose cross section area does not change with bending according to one of the assumptions for ROM, we can confirm the 3D space of SRA at any time instant; similarly, for the object in the same frame, it can be composed of a set of inequalities or a single one like a sphere as follows,

If our code could detect the intersection set of the SRA space and the impacted object’s space at every time step, we can compute when and where they will collide with each other.

* Then we can work out the impact force, and use analytical Jacobian to assembly it onto the right hand side of the new derived EOMs, however, all the external forces are exerted on the SRA tip only, there must be a MATLAB file to make such transformation (obtain an input force vector according to the force on tip and Jacobian). This is also the question I need to ask the author, about how to add an external force in an arbitrary position of the surface of the SRA.
* After dealing with the two major problems above, we can move on for coding work.

1. Adding a PD controller in the package

Since we would like our end-effector (the tip) of the SRA, and states of the tip are all about exact position in the 3D frame and its linear velocities, so our actuated force should be in the format as follows,

Temporarily I haven’t tested out the proper values of the coefficients, I will continue working on it next week.

## **Plan**

1. Consult the author of the package for suggestions in terms of the questions listed above, by the end of next week, I think I should be able to successfully add another arm;
2. Write a trial MATLAB subscript to test if my guess about confirming the impact point could be realized by code;
3. Continue to work on adding a PD controller in the package and learn about how it works in control process.