# **Weekly Report – W12 Fall 2022**

## **Problems & Tasks**

1. Examination of the inertia matrix;
2. Identify what kind of ODE solver has been used in Chase’s model, if it hasn’t been specified, find out if it has been mentioned in other related papers;
3. Self-study about energy shaping and PDE solving;
4. Modifications about the part “back thigh panel” of the exoskeleton.

## **Solution**

1. *Examination of the inertia matrix*

Since in the last week, I examined whether the results for inertia matrix derived using my method and that stated in the paper are the same, it turned out to be yes. But when I kept on to find if the matrix is positive definite or not by checking the eigenvalues, trace and determinant of it respectively, the result was not so promising. Firstly, it took over 10 hours for Mathematica to fully simplify the eigenvalues for a matrix, and they were so complex that we could not figure out if both of them are positive; for the trace, it is obvious to prove it’s positive; and for the determinant, it contains so many substration terms, we could not prove its positivity neither. But I believe if we use the way introduced in Dr. Ian’s paper, the matrix should be correct.

However, when I checked the correctness of the Coriolis (C) matrix, I found that no matter how I simplified the expression, the results of C matrix by using the method introduced in [1] and the ones derived in Dr. Ian’s paper [2] were not identical all the time, finally I found the reason behind. The formula in [1] is probably wrong due to the first edition (from page 202, Eqn. (6.62)), which is shown as follows,

But in [3] and [4], the element in C matrix can be expressed by the following formula (which can be found in page 173 in [3] and page 278 Eqn. (8.17) in [4]),

We can clearly find that the denominators for the partial derivatives are different, in Eqn. (W12-1), there are two repeated denominators , which could be the main reason for simulation failure.

1. *ODE solver used for general soft robot models*

To acquire a more accurate simulation result as well as minimizing the computation cost, a proper ODE solver must be chosen. To identify what kind of solver should be used in our falling SRA project, one possible way is to compare the different results of different ODE solvers, however, the code still runs with errors according to singularity, so probably we shall study what other researchers did for analogous simulation, potential directions can be summarized as follows,

* Since our structure of modelling was enlightened by Chase, we can try to find if the ODE solver has been specified in his dissertation;
* There are some toolboxes like Robotics System Toolbox in MATLAB/Simulink, to solve the dynamic governing equations of a complex robot system, sorts of numerical integration methods (ODE solvers) will be used, we probably can find the answer through the source code;
* Look for some papers about soft robotics simulation in MATLAB/Simulink, check if the ODE solver has been specified in the lines, or if the link of source code was provided, we can look through some details inside;
* Search for some other packages which include the ODE solver.

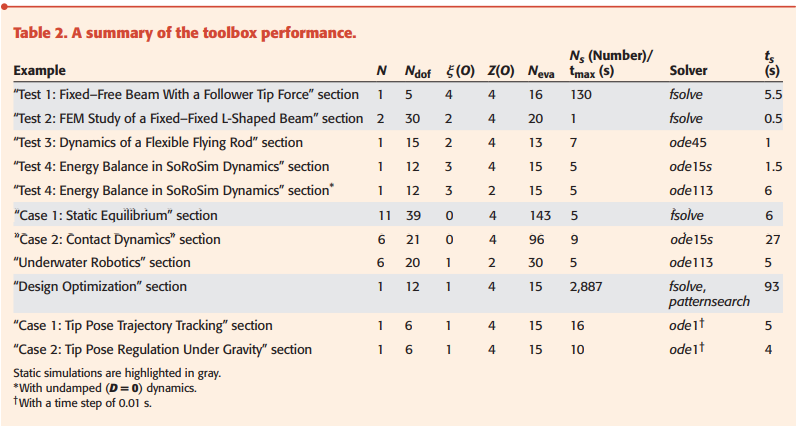
Given the directions above, the achievements are shown below:

(1). Back to Chase’s dissertation once more, I focused on Chapter 3, Continuum Robot Modelling to see if there exists some information about ODE solver, the answer is no, this chapter talks more about the dynamics itself; and Chapter 4 is mainly about the validation of the model on Gazebo.

(2). Indeed, there exist plenty of subsystems, like kinematics, dynamics, robot constraints, manipulator motion planning and so on in the Robotics System Toolbox, but when I checked each function inside, I found that they were highly integrated, the code of ODE solver at the bottom was hidden, which cannot be seen. My comment is that maybe we can use the C code generator to see if the highly integrated code will be expanded in C, in this case, the type of ODE solver will be found.

(3). The main difficulty for confirming the ODE solvers used in papers is that we could not guarantee the packages the researchers used is MATLAB/Simulink, it could be Gazebo, Sofa or anything else. Besides, the programming language they used is probably not in MATLAB, which could be Python or C instead, making it harder to trace what kind of ODE solver has been used. Even if it was specified in the paper that the package is based on MATLAB/Simulink, the source code was seldomly shared.

(4). The last possible solution is to find if there are some other packages are developed based on MATLAB language. It turned out there is something valuable, a package called TMTDyn was usually utilized for modelling soft, rigid and hybrid continuum robots. Solvers are applied for different scenarios, for example, for static equilibrium problems, *fsolve* function will be used; for dynamic problems, *ode15s* and *ode113* functions are suggested to be used (depends on whether the system is stiff or not) [5]. But in Dr. Ian’s paper, only these two ODE solvers have been briefly discussed, apparently they are not sufficient to solve any other problems, as they are a lot more solvers existing. More explicit information can be found in [6], a group of tests were conducted to compare the performance of different ODE solvers, the table below is the screenshot from this paper,



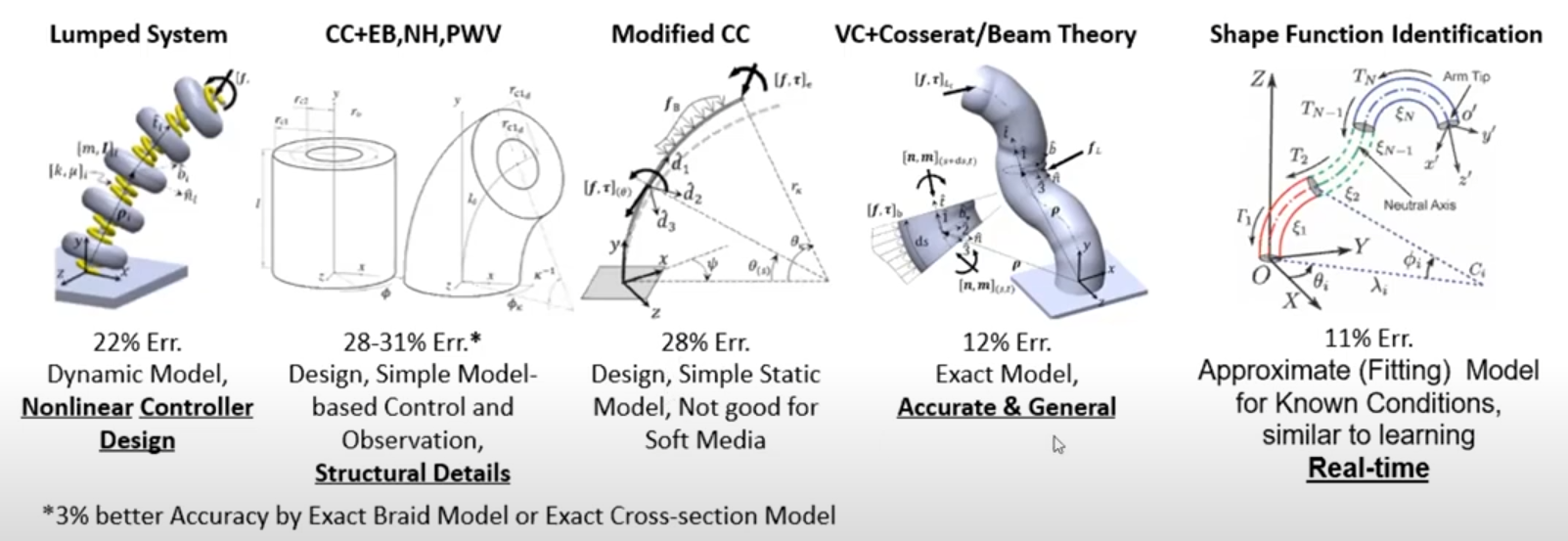
**Fig. W12-1** A summary of TMTDyn Toolbox performance based on the ODE solver selected

where is the total number of links, stands for the total degree of freedom for the model, is the maximum strain order, is the order of Zannah collection approximation, is the total number of points at which static or dynamic equations are evaluated (like FEM) and for static simulation, is total simulations performed; for dynamic one, the 6th column in the table represents the real-time duration and the last column stands for the simulation time. And all the tests are performed on a PC with the following specifications: Intel Core i7-1065G7 CPU at 1.3 GHZ and 1.5 GHz with 16 GB RAM.

There is one thing worth note that the TMTDyn Toolbox is more suitable for systems that its soft links can be modelled by applying Cosserat Rod Theory. And it can be concluded that the most frequently used ODE solvers in the package are *fsolve*, *ode15s* and *ode113*, among them, *fsolve* is just for static problems, which is exactly the same as the statement in [1]; *ode15s* is usually used for stiff systems, like “Case 2: Contact Dynamics” in the table above, the section contains very fast and slow variation terms (the velocity and displacement of the contact point, the theory to judge if a system is stiff or not has been introduced in Weekly Report 10 – Fall 2022); and *ode113* is used for non-stiff dynamic problems according to its high order tolerance error control.

1. *Energy shaping method for modelling soft robots*

With the development of soft robot control, there emerged lots of different methods for modelling, like lumped, constant curvature (CC), Cosserat rod theory and energy shaping function and so on. As shown in the figure below, modelling soft robot with Cosserat theory and shape function has a much better performance than conventional methods like lumped and CC in terms of the system errors. This might be due to the former two approaches mentioned previously are more appropriate for continuum robots, by computing the stress and strain of each point on the central line to identify its configuration space.

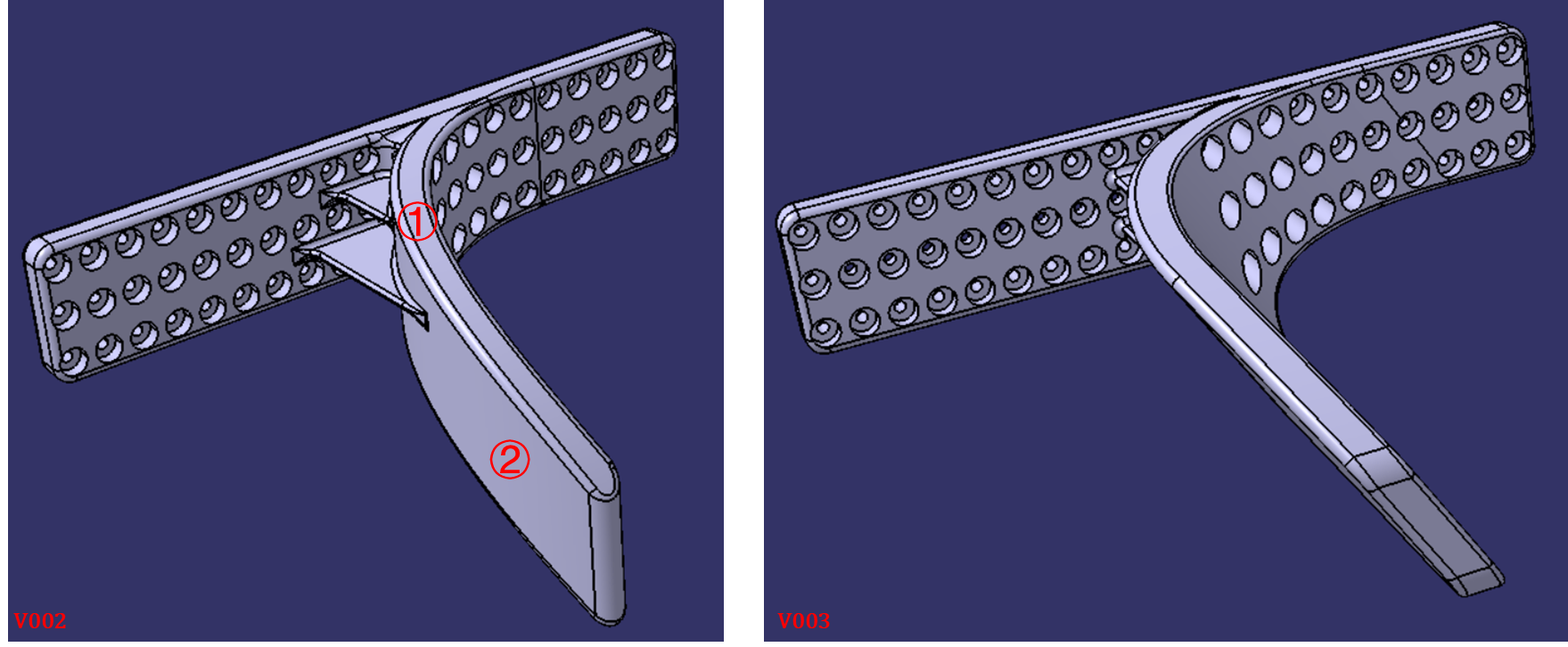


**Fig. W12-2** The system error of each modelling method of soft robot

Now I’m still in the entrance level in this field, I started from some lecture videos to get some rough ideas about this concept last week, I will keep looking in the near future to accumulate enough background knowledge to prepare model with energy shaping method when necessary.

1. *Modifications about the part “back thigh panel” of the exoskeleton*

Due to the feedback from the subjects (including me) recently, I was thinking about how to make some improvements afterwards. Since there were a few limitations about the version 2 (V002) which is the design on the left hand side in the figure below, the amendments accordingly are as follows:



**Fig. W12-3** The modifications from V002 to V003

(1). As the shape of human’s thigh is more like a cone rather than a cylinder, the portion marker as No.1 in V002 was originally designed to be perpendicular with the ground, now it has been modified with an adaptable angle to fit with the thigh;

(2). The portion marker with No.2 was designed to be an arc shape, the problem was that due to different body sizes of different subjects, this part cannot be guaranteed that it would 100% fit with the thigh, usually there will be a little gap between, which makes the subjects uncomfortable. To avoid this issue, the extended part was designed to be tangential to arc curve connected with the base.

## **Reference**

[1] Spong, Hutchinson and Vidyasagar (2006) *Robot Modeling and control*. New York: John Wiley & sons.

[2] Wang, C. *et al.* (2021) “Dynamic control of multisection three-dimensional continuum manipulators based on virtual discrete-jointed robot models,” *IEEE/ASME Transactions on Mechatronics*, 26(2), pp. 777–788. Available at: <https://doi.org/10.1109/tmech.2020.2999847>.

[3] Murray, R.M., Li, Z. and Sastry, S.S. (1994) *A mathematical introduction to robotic manipulation*. Boca Raton: CRC.

[4] Lynch, K.M. and Park, F.C. (2019) *Modern Robotics: Mechanics, planning, and Control*. Cambridge, United Kingdom ; New York, USA ; Port Melbourne, Australia ; New Delhi, India ; Singapore: Cambridge University Press.

[5] Sadati, S.M.H. *et al.* (2020) “*tmtdyn*: A Matlab package for modeling and control of Hybrid Rigid–continuum robots based on discretized lumped systems and reduced-order models,” *The International Journal of Robotics Research*, 40(1), pp. 296–347. Available at: <https://doi.org/10.1177/0278364919881685>.

[6] Mathew, A.T. *et al.* (2022) “SoRoSim: A matlab toolbox for hybrid rigid-soft robots based on the geometric variable-strain approach,” *IEEE Robotics & Automation Magazine*, pp. 2–18. Available at: <https://doi.org/10.1109/mra.2022.3202488>.