# **Weekly Report – W9 Fall 2022**

## **Problem**

1. Use the properties of determinant and trance of a matrix to examine if the inertia matrix in the governing equation is positive definite via Mathematica;
2. Specify where the ODE solvers have been used in Simulink;

## **Solution**

1. Examination of inertia matrix

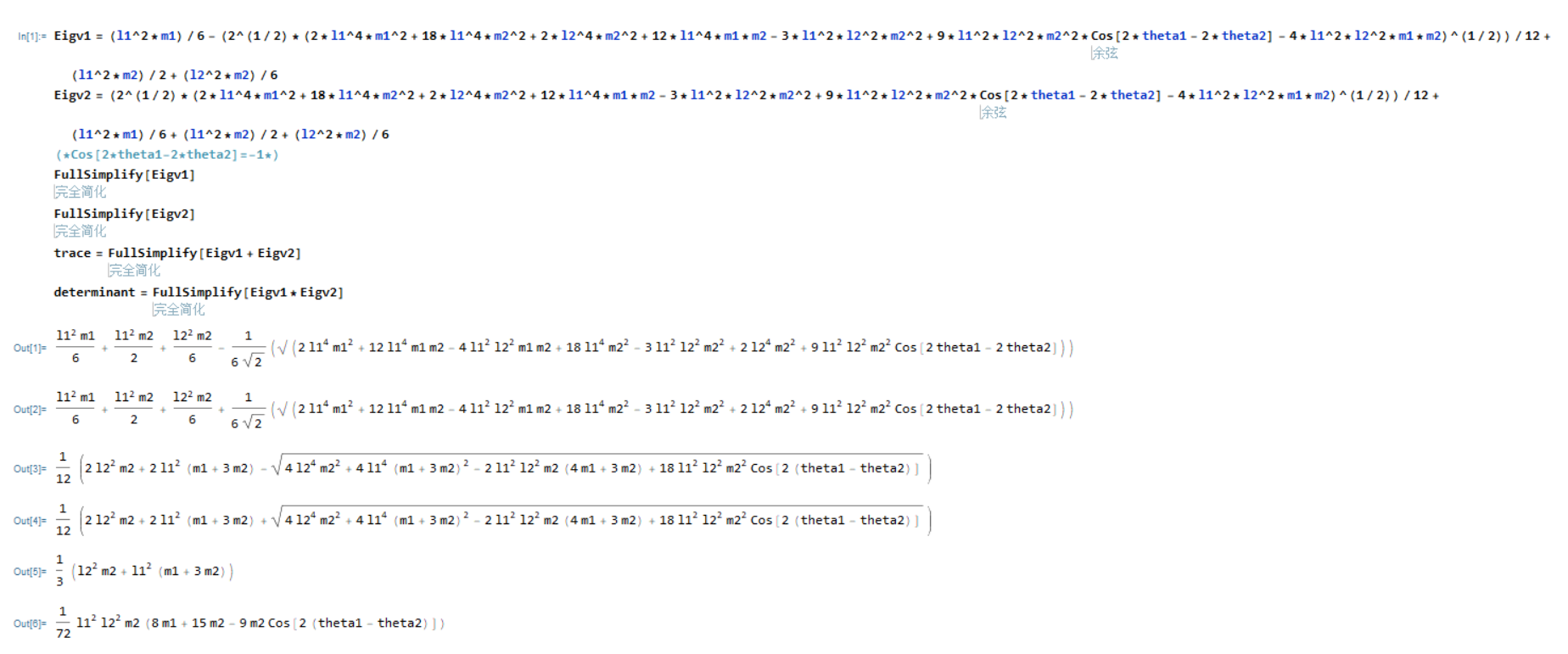
Since it was hard to examine the correctness of inertia matrix in the governing equation just by judging the sign of each eigenvalue, we can try to use the following two properties of a matrix:

(1). Trace of matrix: The sum of all eigenvalues of a matrix.

(2). Determinant of matrix: The product of all eigenvalues of a matrix.

For a inertia matrix, if it is positive definite, the necessary and sufficient conditions are that both the trace and determinant of the inertia matrix should be greater than zero (or positive).

The results showcased by Mathematica is as follows,



**Fig. 1** The computation results of trace and determinant from Mathematica

Since to judge the sign of each eigenvalue of the inertia matrix can be hard when some of the parameters cannot be determined, calculating the trace and determinant of it will be an alternative, and the results of them are:

Apparently, the trace will always be positive, for the determinant, we can plug in the upper and lower boundary of the term , and correspondingly the two boundaries of the determinant will be

It turns out both of them are positive, so we can say the inertia matrix is positive definite. However, back to the falling SRA simulation, I tried to evaluate the inertia matrix by the same method in Mathematica, the inertia matrix contains too many high order terms of the variables like , which made it even harder to check the sign of the trace and determinant. The only thing I can guarantee is that I utilized the same way to derive the inertia matrix, so there shouldn’t be any issues.

1. Several questions for solving ODEs in Simulink

During this week, I have read a lot of tutorials, blogs and lecture videos about the following topics, how to solve ODEs in Simulink, how ODE solver works in Simulink, how integrator block works for solving ODE is Simulink and so on, though I have already known most of them, the aim is to find if there are any direct answers or indirect indications to the concern for which step the ODE solver has been applied. But it turned out few of the resources are useful.

(1). Integrator block

Firstly I will copy down exactly the same information as provided in MATLAB official help center, which introduces the basic function of this block.

The block dynamics are given by

Where is the block input, is the block output, is the block state and is the initial condition of .

*“While these equations define an exact relationship in continuous time, Simulink uses* ***numerical approximation methods*** *to evaluate them with* ***finite precision****. Simulink can use several different numerical integration methods to compute the output of the block, each with advantages in particular applications.”*

*“The* ***selected solver*** *computes the output of the Integrator block at the current time step, using the current input value and the value of the state at the previous time step. To support this computational model, the Integrator block saves its output at the current time step for use by the solver to compute its output at the next time step.”* [1]

From the official documentation above, we can deduct that one of the locations for ODE solver been applied in the interval from the function block (which contains the dynamics) to the integrator blocks at least, since some of the integrator block’s inputs are computed by the dynamics of the system. But this is just a very general deduction, which needs to be proved furthermore.

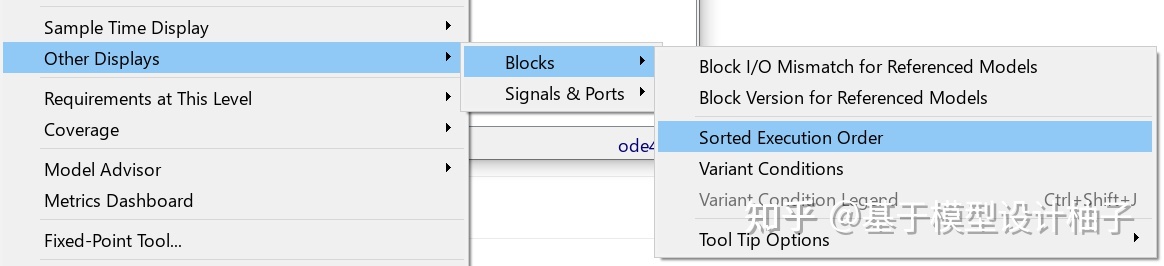
(2). From the specifications of ODE solvers

Facially the concepts of “how the ODEs are solved?” and “different ODE solvers” are basically the same, however, the first concept concerns more about a more general method in the big picture, using numerical integration and approximation is the key feature for Simulink to solve ODEs; ODE solvers focus more about the trade-off between simulation precision and computation cost.

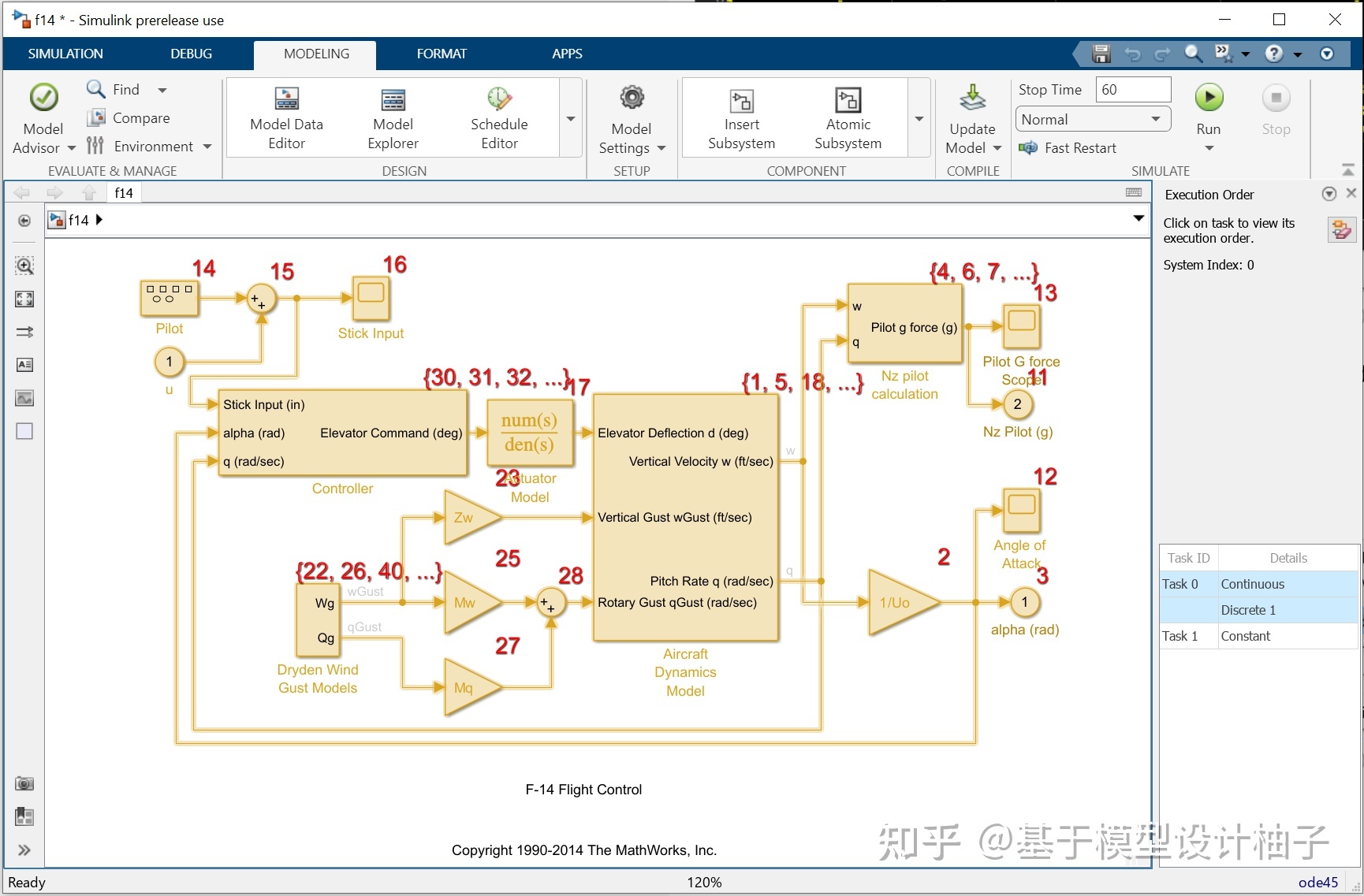
There are plenty of examples that can prove ODE solvers have truly been used in simulation process even there are no ODEs inside, and using different ODE solvers can lead to slightly different results, which makes it hard to locate the specific section the solver has been used.

(3). From the execution order of Simulink model

The subsequent example [2] comes from Zhihu, which is about analysis of how the ODE solver works by observing the data flow.



**Fig. 2** The procedure to display the execution order of the Simulink model



**Fig. 3** The data flow in the example Simulink model

When we select the sorted execution order, we will find that all the Simulink blocks and links turn yellow, which means that the ODE solver has been used throughout the whole model. In Fig.2 we can see that the procedure was not strictly performed as what we thought, that is because some hidden blocks are inside. This example just indicates that the ODE solver does not act on a single block or section, instead it works for everywhere of the model.

1. Progress on SRA simulation

* The symbolic variables stored in struct cannot be directly substituted even in MATLAB, it could be one of the reasons that the Simulink model cannot run.

## **Reference**

[1] *Discrete-time integrator* (no date) *Integrate signal - Simulink*. Available at: https://www.mathworks.com/help/simulink/slref/integrator.html (Accessed: October 24, 2022).

[2] *How Simulink solver computes the model*, *Zhihu*. Available at: https://zhuanlan.zhihu.com/p/550096542 (Accessed: October 24, 2022).