基于Syslab和Sysplorer的机理数据融合仿真建模操作手册



|  |  |  |  |
| --- | --- | --- | --- |
| **编制** | **曹宇宇** | **生效日期** |  |
| **审核** |  | **批准** |  |

**文件变更摘要**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **日期** | **版本** | **变更说明** | **修订** | **审核** | **批准** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

目录

[1. 引言 1](#_Toc130567604)

[1.1 概述 1](#_Toc130567605)

[1.2 参考资料 1](#_Toc130567606)

[2. 总体设计 1](#_Toc130567607)

[2.1 总体架构 1](#_Toc130567608)

[2.2 技术路线 3](#_Toc130567609)

[3. 应用流程 4](#_Toc130567610)

[3.1 操作步骤 4](#_Toc130567611)

[3.1.1 Train 4](#_Toc130567612)

[3.1.2 Build 5](#_Toc130567613)

[3.1.3 Infer 10](#_Toc130567614)

[3.2 结果展示 16](#_Toc130567615)

# 引言

## 概述

基于Syslab和Sysplorer的机理数据融合仿真建模操作手册旨在通过构建简单伺服翼模型，将操作流程可视化，方便用户对该工具包的使用。

该工具包使用多种语言、多种平台构建而成。首先，在Syslab平台训练深度学习模型，并将其保存；接着，对保存的模型进行编译，使其成为可以被机器识别的二进制可执行文件；最后，在Sysplorer平台搭建仿真模型，并通过接口调用已写好的函数，实现模型推理，并将推理结果用于联合仿真或可视化。

## 参考资料

1. Sysplorer官方文档

<https://www.tongyuan.cc/SysplorerHelp.html#/>

1. Syslab官方文档

<https://www.tongyuan.cc/help/SyslabHelp.html>

# 总体设计

## 总体架构

通过在Syslab平台上的数据预处理、模型训练、模型保存、模型导出操作将已训练好的深度学习模型保存并导出，再通过Sysplorer平台的Modelica API导入模型并调用，实现使用数据模型与物理模型的融合建模。

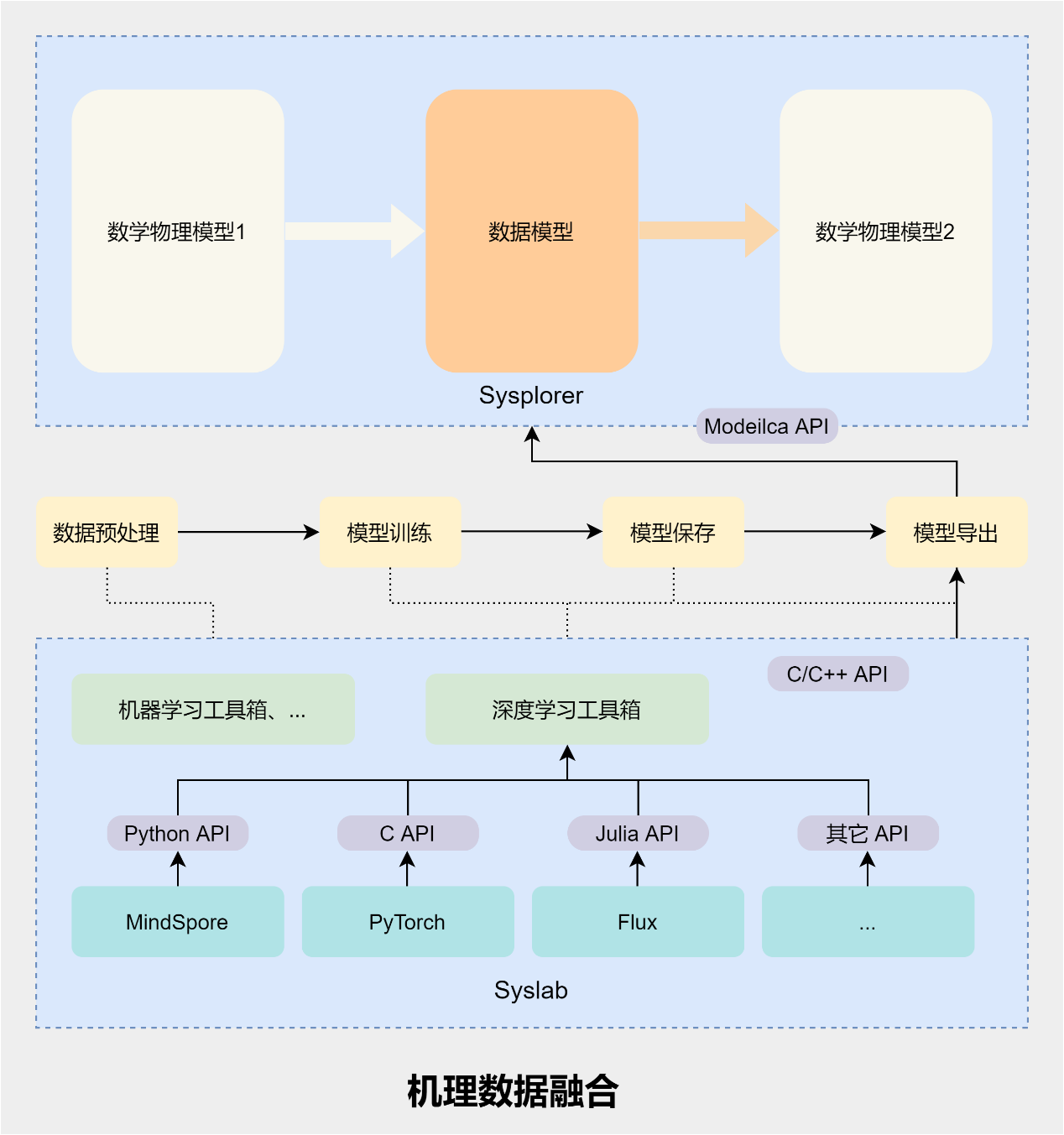


图2.1 总体架构

该工具包主要实现了Syslab和Sysplorer两个平台的联合建模。

Syslab平台主要使用了深度学习工具箱，该工具箱目前仅支持导出MindSpore框架的模型，即.mindir格式的深度学习模型。在未来的新版本中，将支持更多的框架以及更多格式的模型。例如，PyTorch框架的C函数模型、Flux框架的Julia函数模型等。

Sysplorer平台目前仅支持Modelica调C函数操作，即将训练好的模型通过C开放接口，再使用Modelica调用C实现模型推理。

## 技术路线

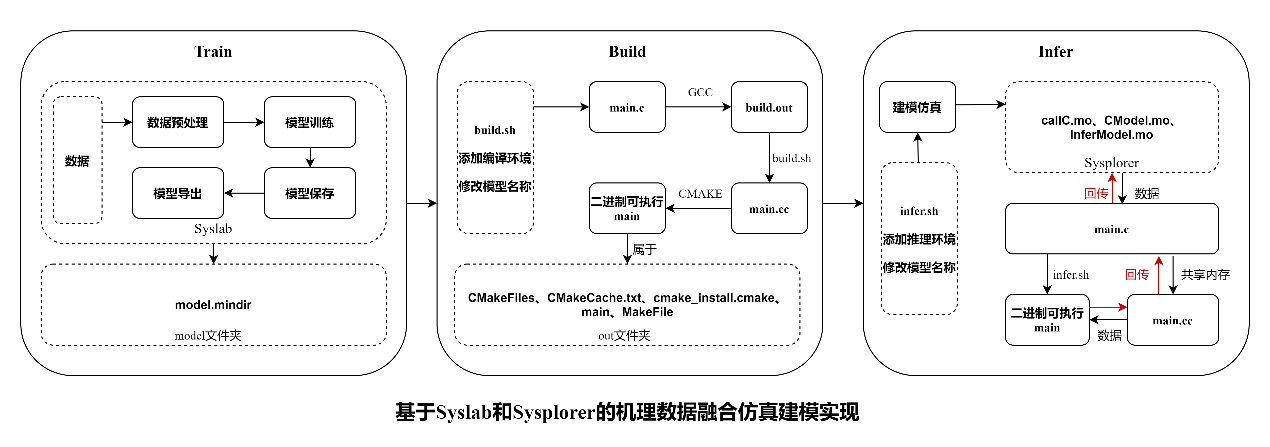


图2.2 技术路线

基于Syslab和Sysplorer的机理数据融合仿真建模实现主要包含Train、Build和Infer三个模块，具体实现逻辑如下：

Train模块主要负责深度学习模型的训练，该模块主要在Syslab中进行。首先将采集到的数据使用机器学习工具箱进行数据预处理，处理完成的数据使用深度学习工具箱进行模型训练，并将训练好的模型通过save函数保存为.mindir格式，最后将模型导出到指定model文件夹中用于接下来模型编译。

Build模块主要负责模型编译，该模块主要在用户Linux环境下进行。第一次加载模型需进行编译，之后推理阶段将不再进行编译，但每次更换模型需重新编译新的模型。首先用户需要在build文件夹中的build.sh脚本中添加mindspore的编译环境和已保存的模型名称（每个用户不尽相同）。修改完成后，对build中的main.c进行编译生成输出文件build.out。执行build.out后，将自动调用用户已设置的编译环境、模型以及mindspore开放的C++接口main.cc，使用Cmake对模型进行编译生成二进制的可执行文件main，该文件将被存放在out文件夹下，以便后续推理时使用。

Infer模块主要负责模型推理和联合仿真建模的实现，该模块主要在Sysplorer平台中进行。首先用户需要在infer文件夹中的infer.sh脚本中添加mindspore的编译环境和已保存的模型名称（每个用户不尽相同）。然后用户需要根据自己仿真模型的输入输出构建自己的modelica模型，并将C函数接口传入modelica中。构建完成模型后即可进行联合仿真，数据模型会将上一个数学物理模型传出的数据读入，通过modelica调用main.c，将数据传递给C函数，C函数一方面会通过调用infer.sh将用户已设置的编译环境、模型传递给可执行二进制文件main，另一方面会通过共享内存将仿真数据传递给mindspore开放的C++接口main.cc，main.cc再将数据传递给main以供推理。在main收到C函数和C++函数传递的所有信息后，将进行推理，并把模型推理的结果通过main.cc、main.c回传给Sysplorer数学模型，该流程同样使用了共享内存技术。最后，Sysplorer数学模型在收到推理结果后，将选择继续传递给下一个数学物理模型还是结果展示（取决于用户搭建的联合仿真模型）。

# 应用流程

本操作手册主要基于伺服翼系统进行机理数据融合仿真建模，将复杂的伺服翼系统替换为数据驱动的深度学习模型，即将下图中System02H机理模型替换为AIModel数据模型。

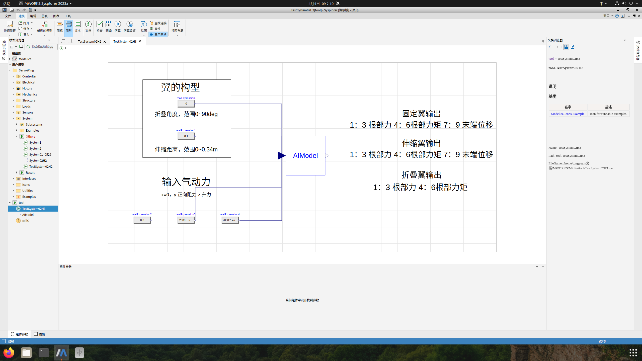
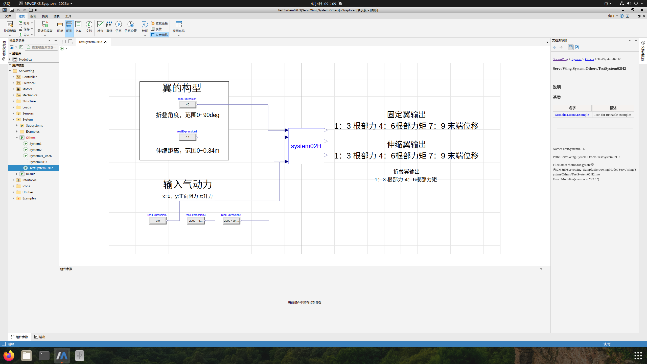


图3.1 System02H模型（左）和AIModel模型（右）

使用Sysplorer构建的System02H伺服翼机理模型如下图所示：

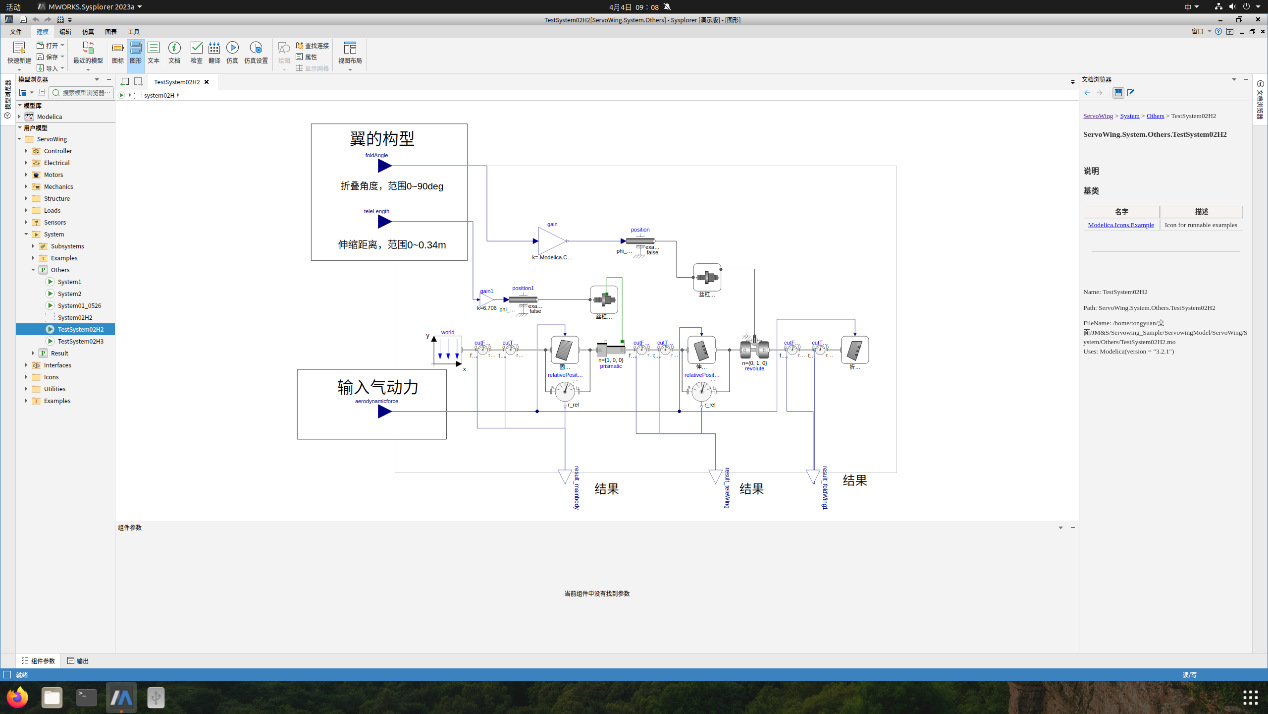


图3.2 System02H模型

## 操作步骤

基于Syslab和Sysplorer的机理数据融合仿真建模实现分为以下三步：

### Train

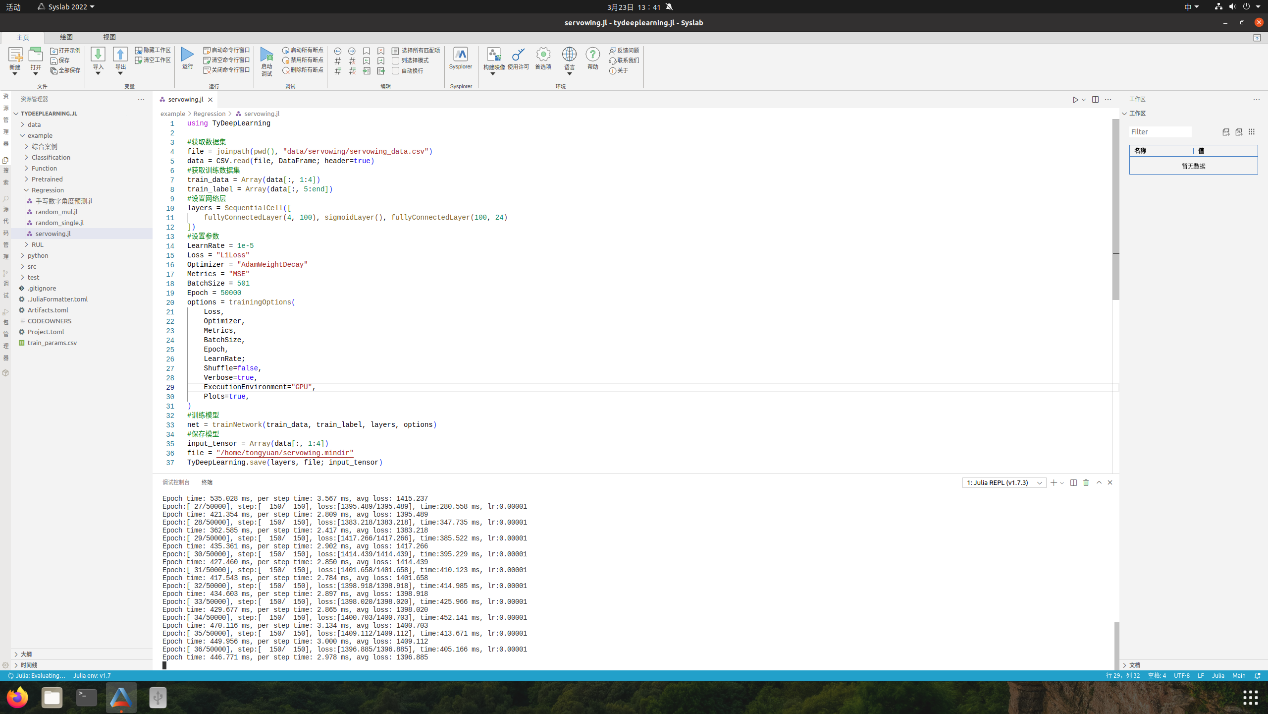


图3.3 模型训练

该系统有四个输入参数，二十四个输出参数，共75150条数据参与训练。模型采用传统的全连接网络，学习率为1e-5，损失函数为“L1Loss”，优化器为“AdamWeightDecay”，评估指标为“MSE”，批次大小为501，训练轮次为50000，采用GPU训练。将上述参数设置完成后进行训练并将训练完成的模型保存为.mindir格式，最后将其复制到指定model文件下。

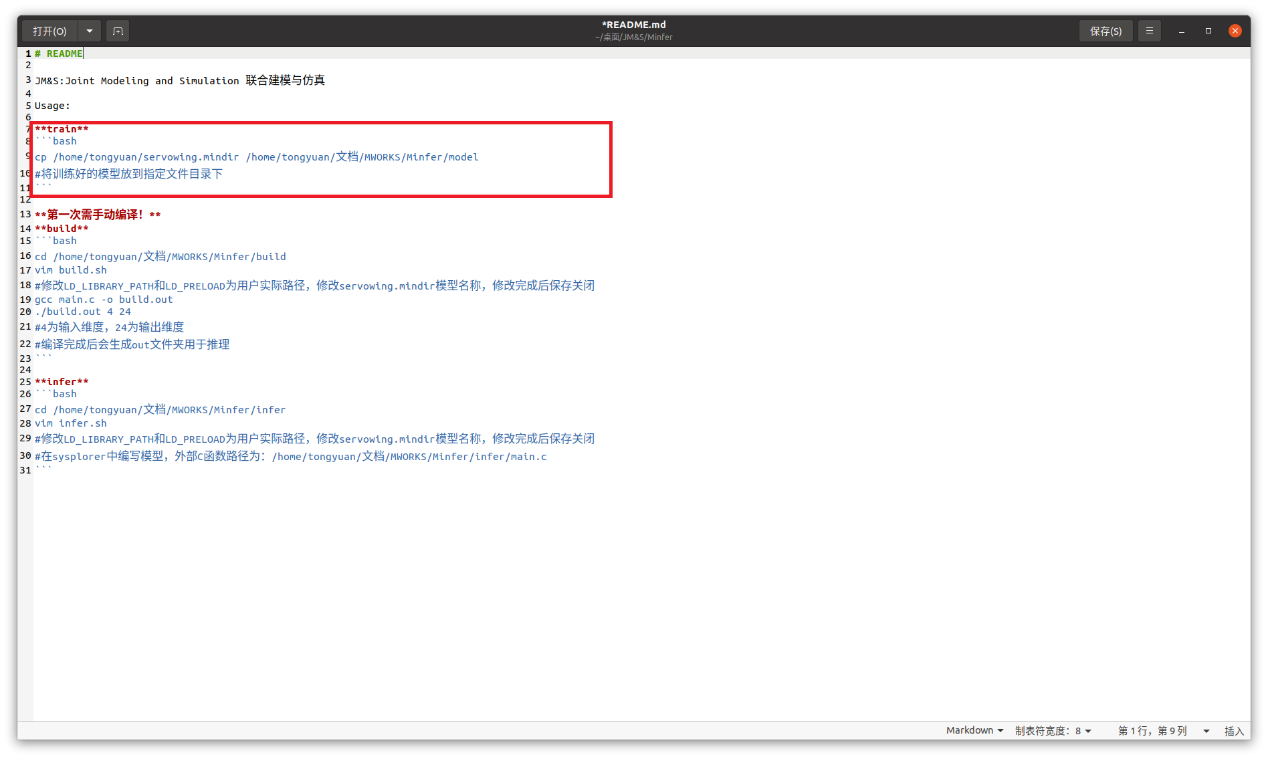


图3.4 模型保存

使用命令cp /{Home}/{model} /{Home}/文档/MWORKS/Minfer/model将模型保存到指定路径，{Home}为用户主目录，当前为home/tongyuan，{model}为用户保存的模型名，当前为servowing.mindir。

### Build

1. 基本概况

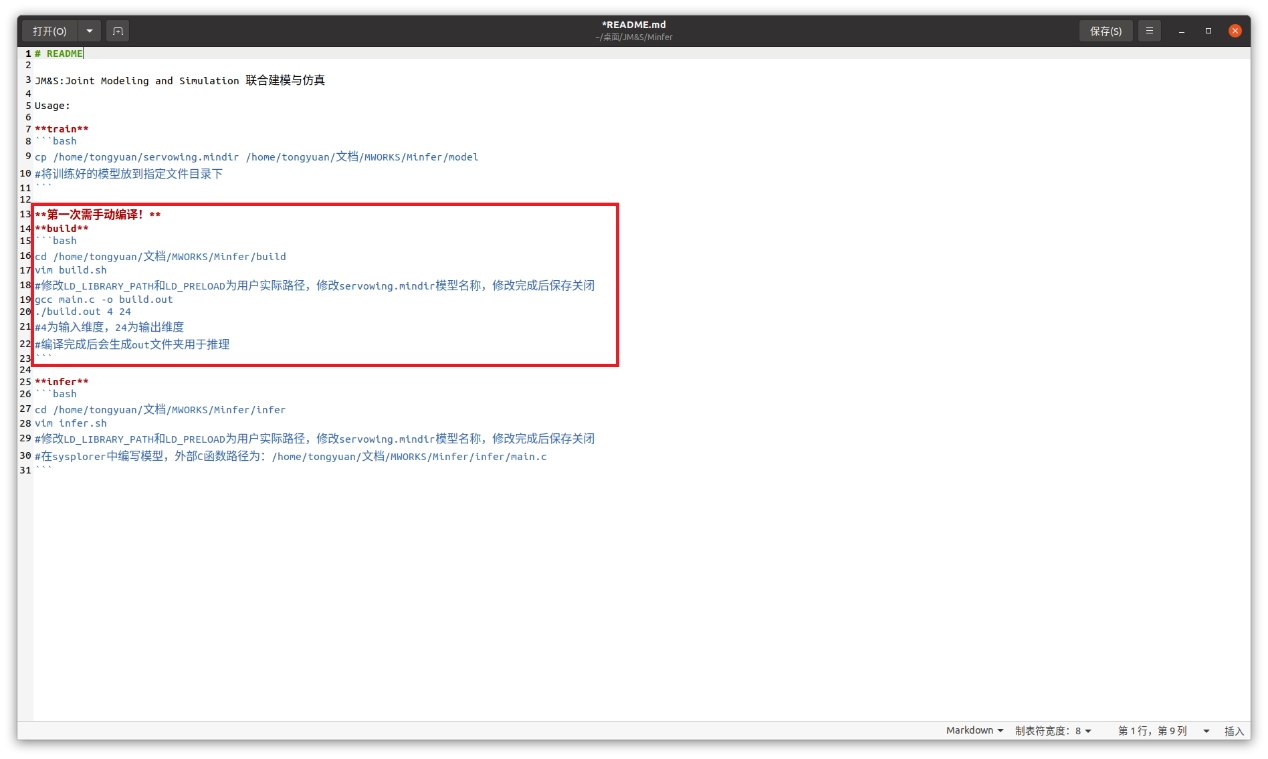


图3.5 Build流程

Build阶段分为三个步骤：

1. 切换到指定目录下。
2. 打开并修改build.sh文件。
3. GCC编译C函数。

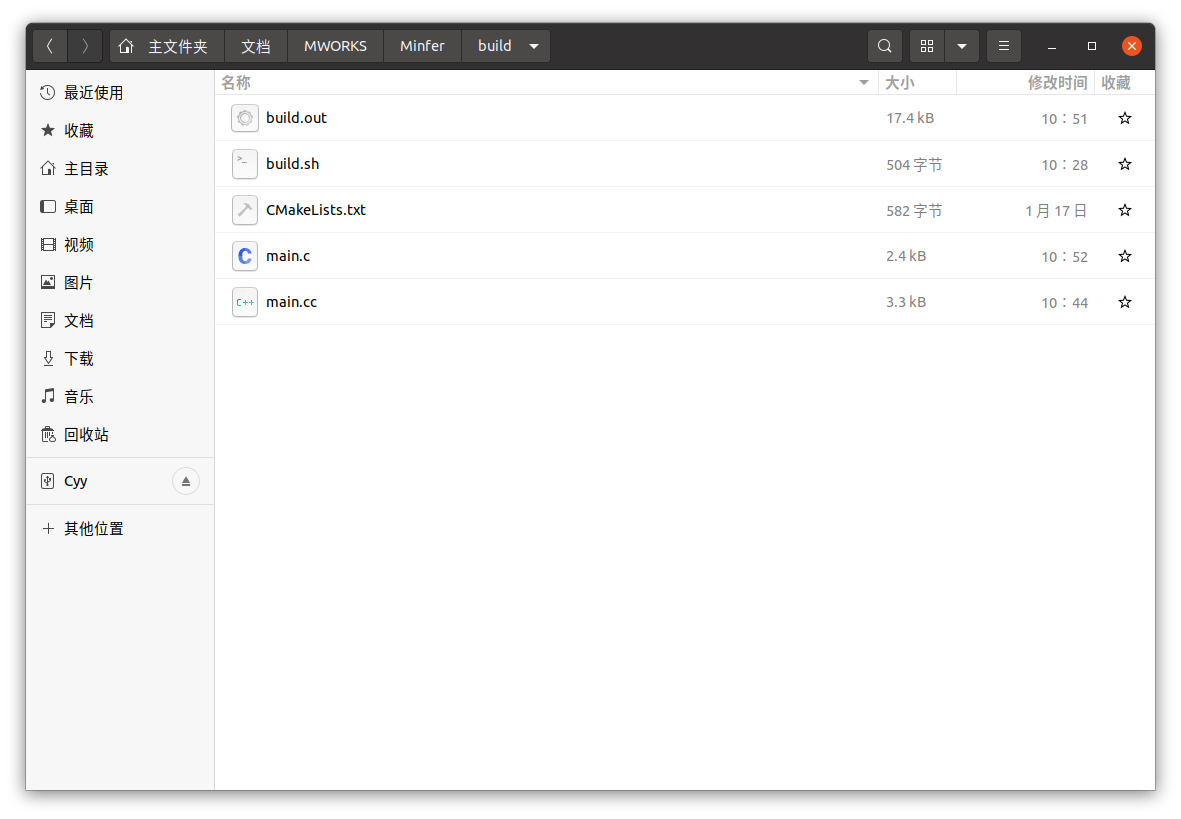


图3.6 build目录结构

build文件夹共包含以下文件：

* build.out：自动编译脚本的输出文件。
* build.sh：自动编译脚本，用户需手动修改。
* CmakeLists.txt：模型Cmake编译脚本文件。
* main.c：C函数文件，包含调用build.sh脚本和main.cc命令。
* main.cc：C++函数文件，模型推理源文件。

1. 操作步骤

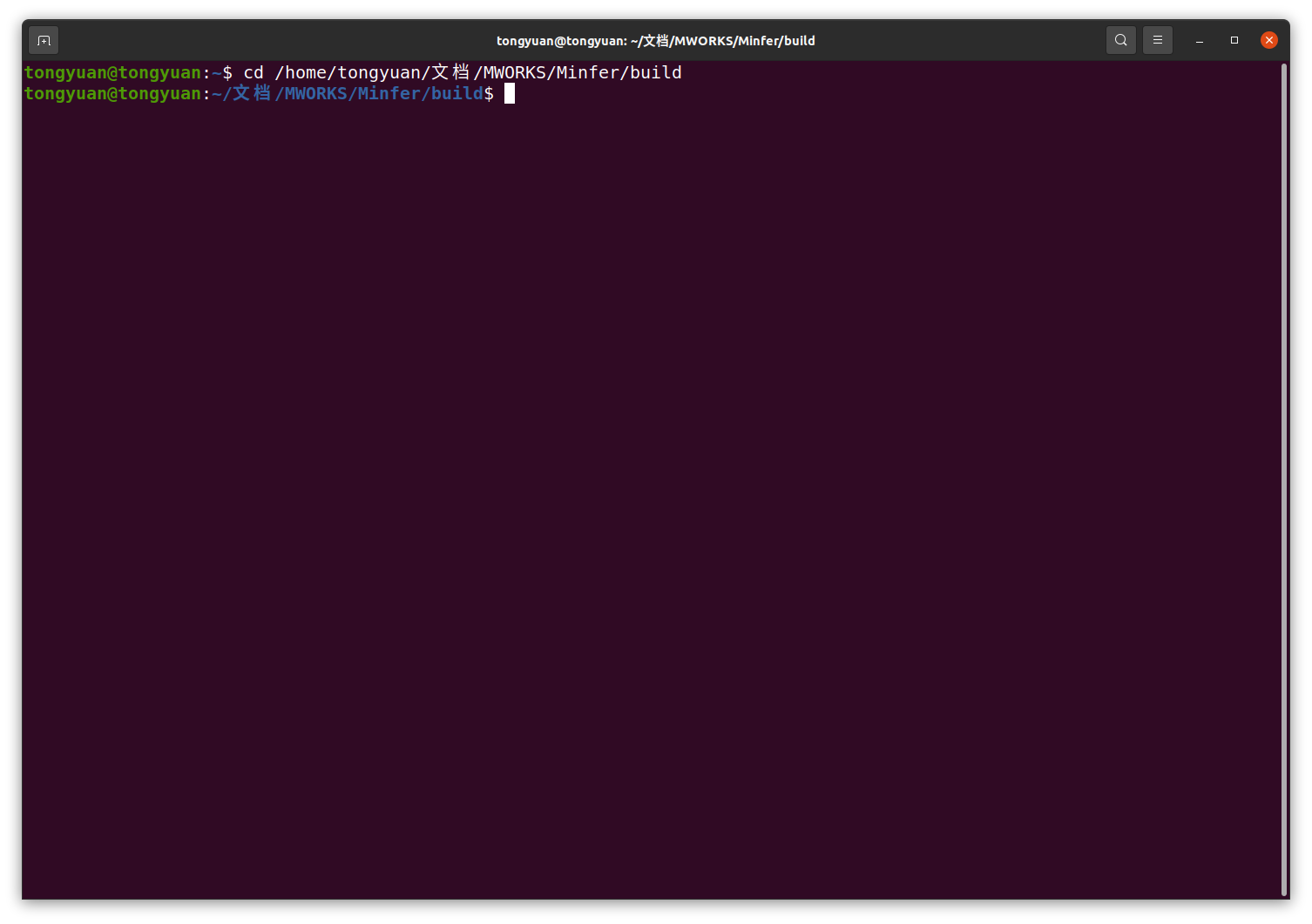


图3.7 模型编译(一)

该步骤为模型编译第一步，切换到用户MWORKS所在的build目录下，命令为：cd /{Home}/文档/MWORKS/Minfer/build。{Home}为用户自己的主目录路径，当前显示为home/tongyuan。

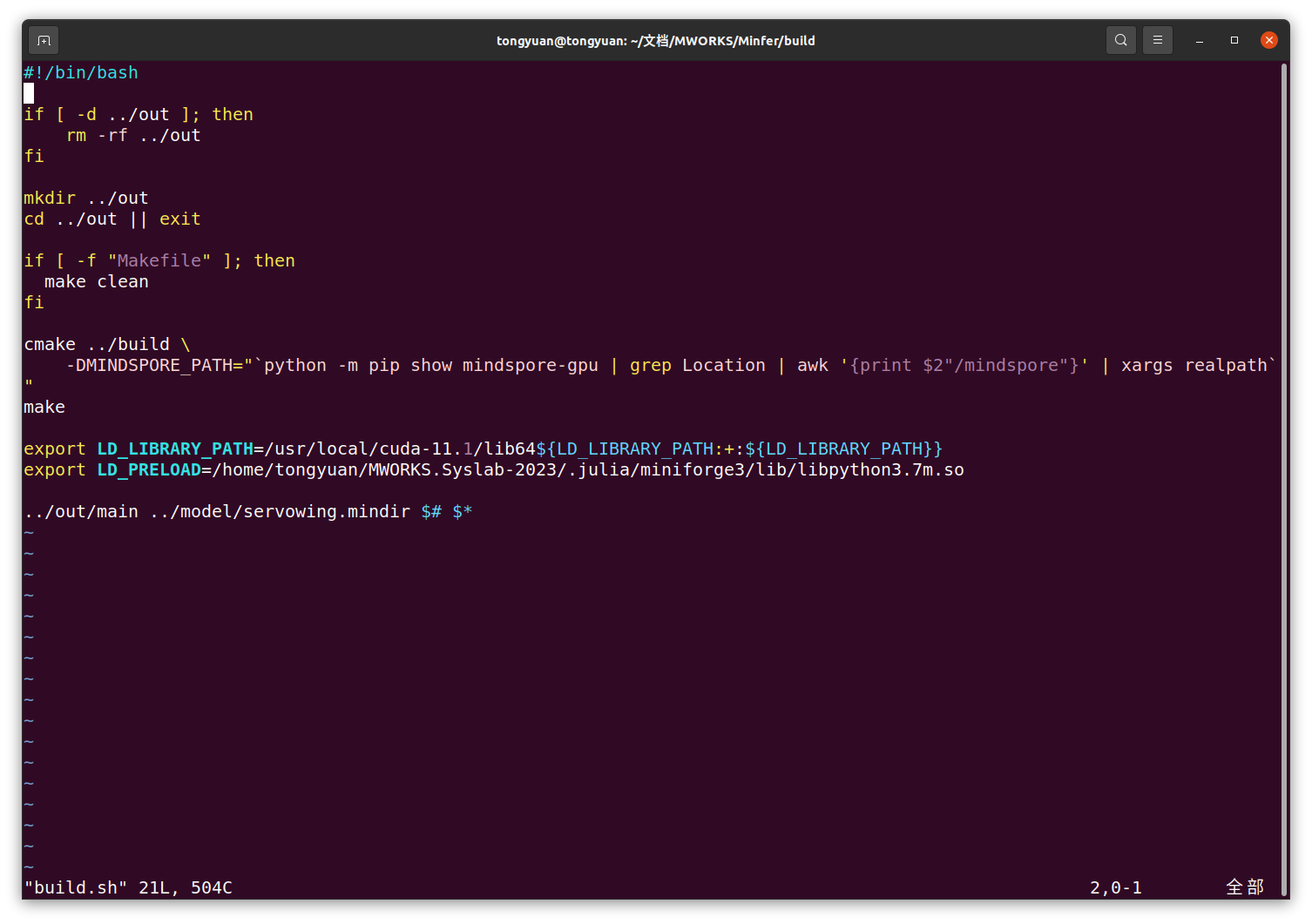


图3.8 模型编译(二)

该步骤为模型编译第二步，通过vim build.sh打开脚本文件，使用i命令打开编辑模式，修改LD\_LIBRARY\_PATH和LD\_PRELOAD为用户自己的路径，并对模型名称进行修改，当前为servowing.mindir，用户可根据自己命名修改为特定的名称。修改完成后，Esc推出编辑模式，:wq保存退出脚本文件。

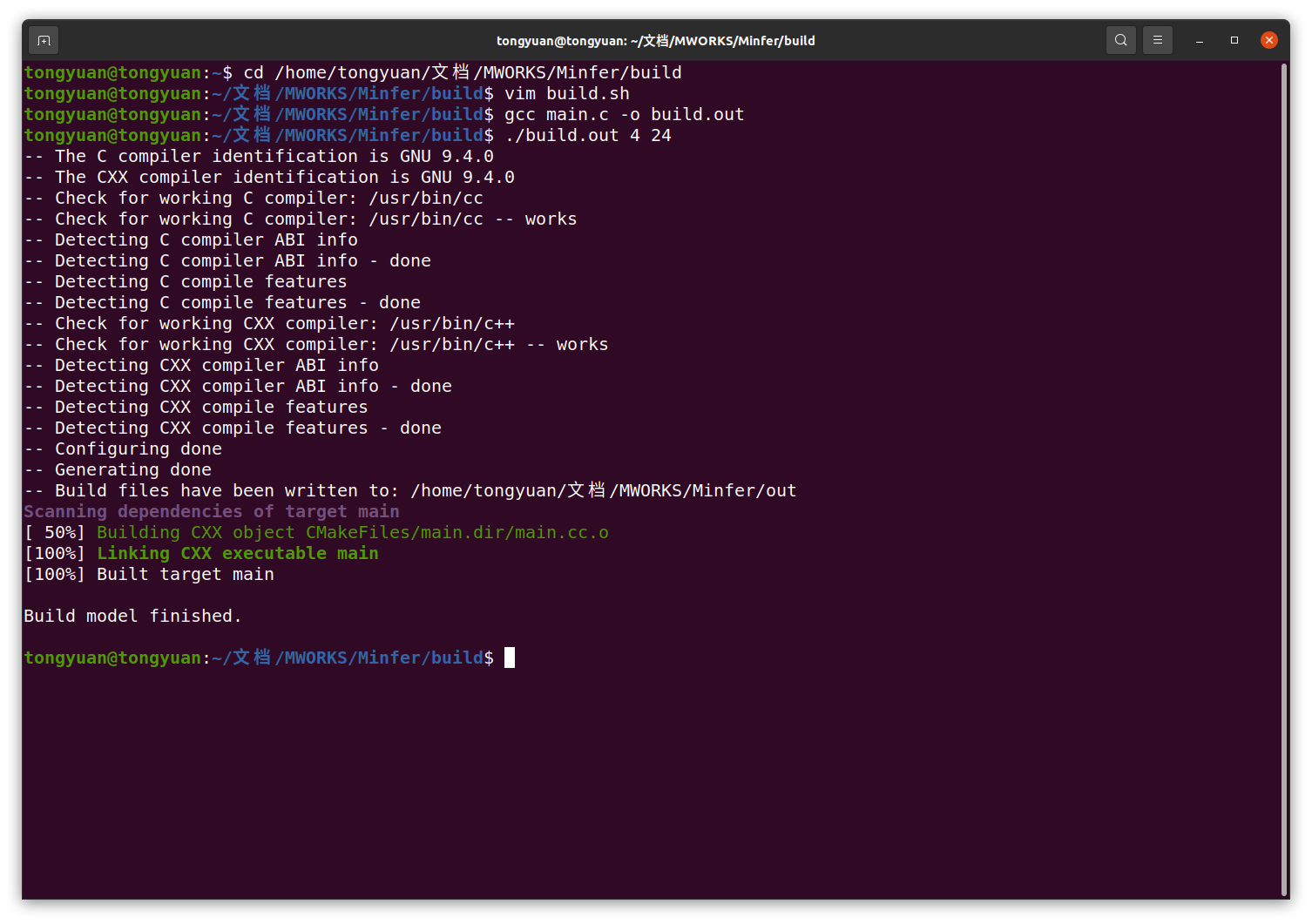


图3.9 模型编译(三)

该步骤为模型编译第三步，通过gcc main.c -o build.out命令将C函数文件编译为可执行文件build.out，执行该可执行文件./build.out {input} {output}，{input}为模型训练输入维度，当前为4，{output}为模型训练输出维度，当前为24。编译完成后即可生成out文件夹，里面包含了推理所需的所有文件。

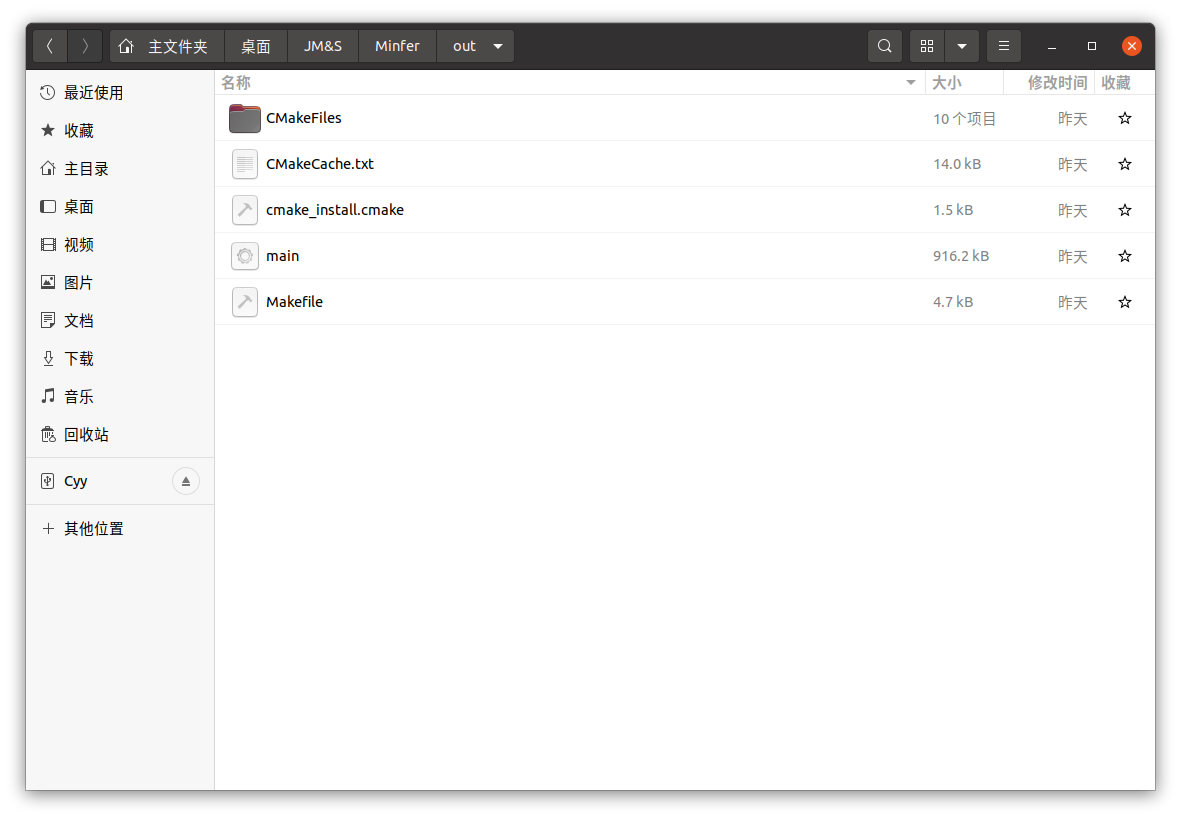


图3.10 out文件夹

### Infer

1. 基本概况

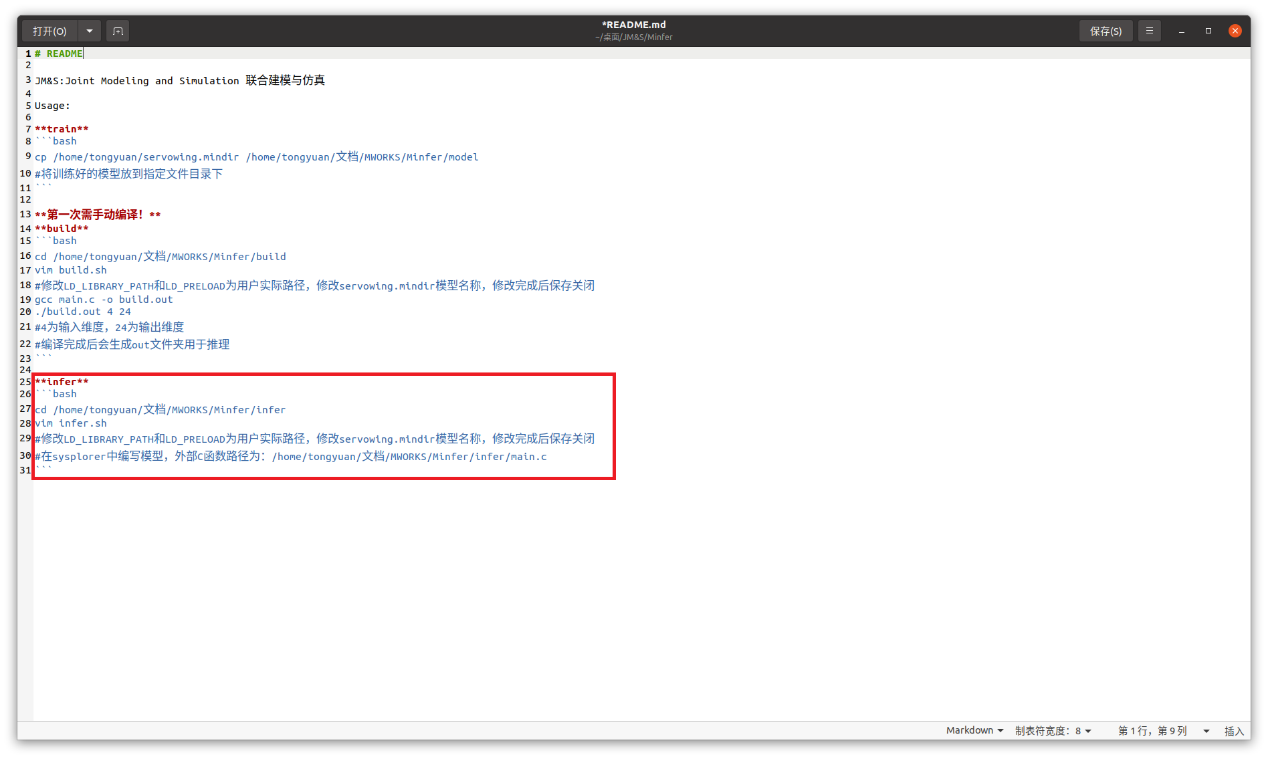


图3.11 Infer流程

Infer阶段分为三个步骤：

1. 切换到指定目录下。
2. 打开并修改build.sh文件。
3. 加载并修改modelica模型，添加用户自己的模型进行联合仿真。

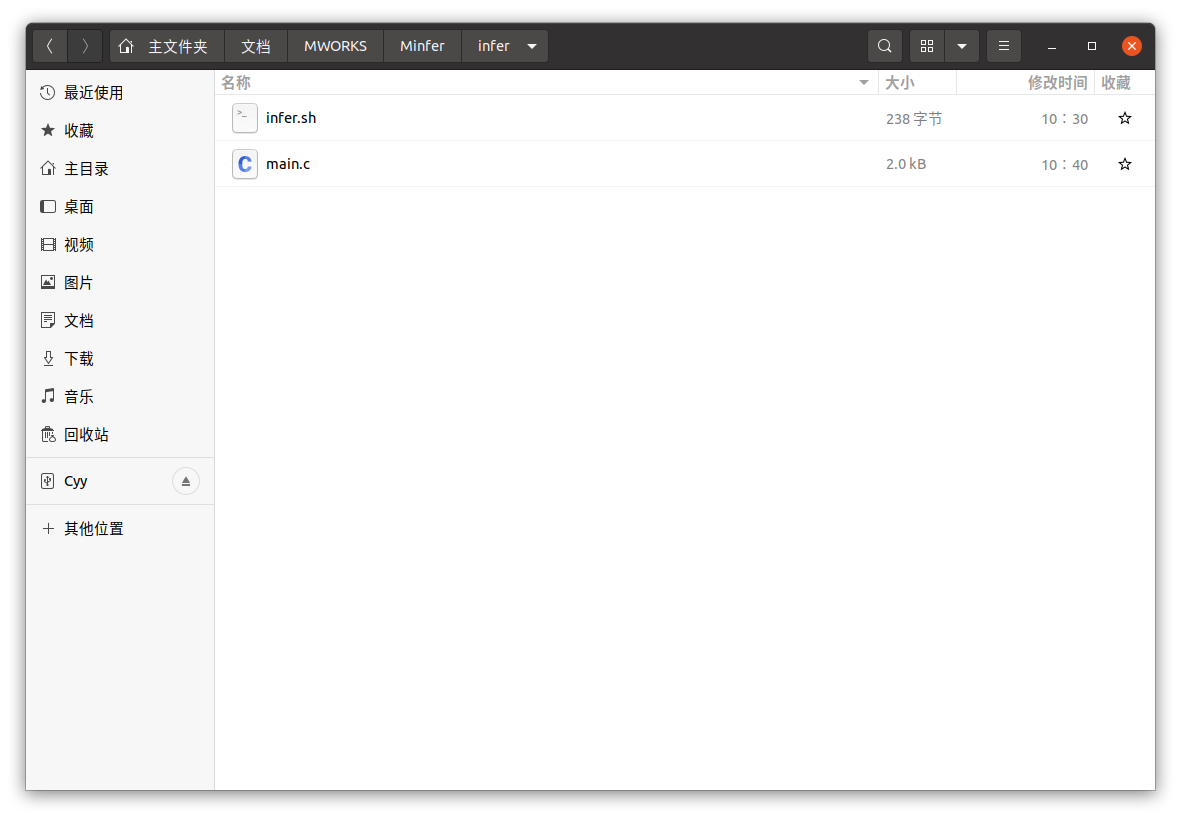


图3.12 infer目录结构

infer文件夹共包含以下文件：

* infer.sh：自动编译脚本的输出文件。
* main.c：C函数文件，包含调用infer.sh脚本和模型可执行main命令。

1. 操作步骤

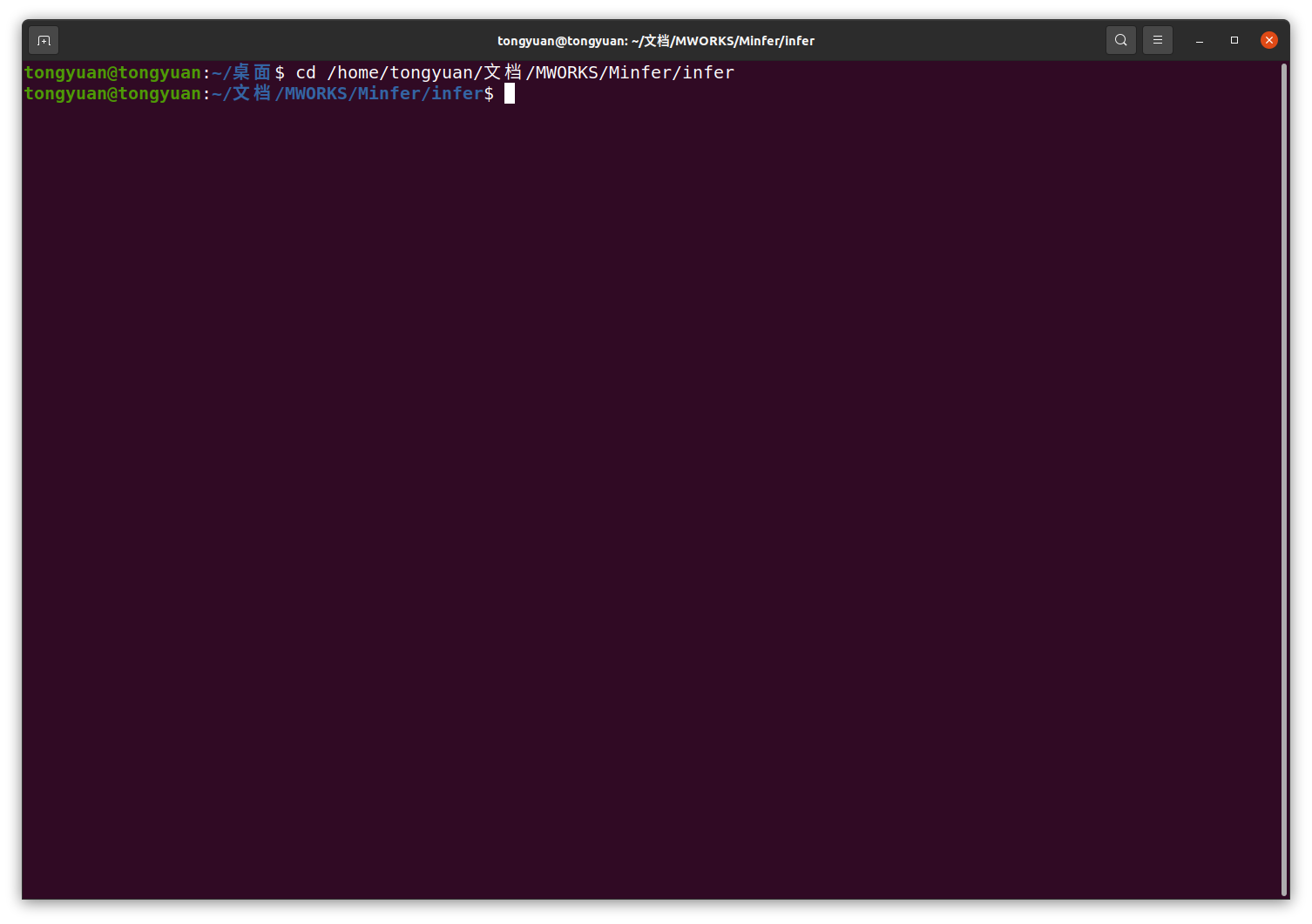


图3.13 模型推理(一)

该步骤为模型推理第一步，切换到用户MWORKS所在的infer目录下，命令为：cd /{Home}/文档/MWORKS/Minfer/infer。{Home}为用户自己的主目录路径，当前显示为home/tongyuan。

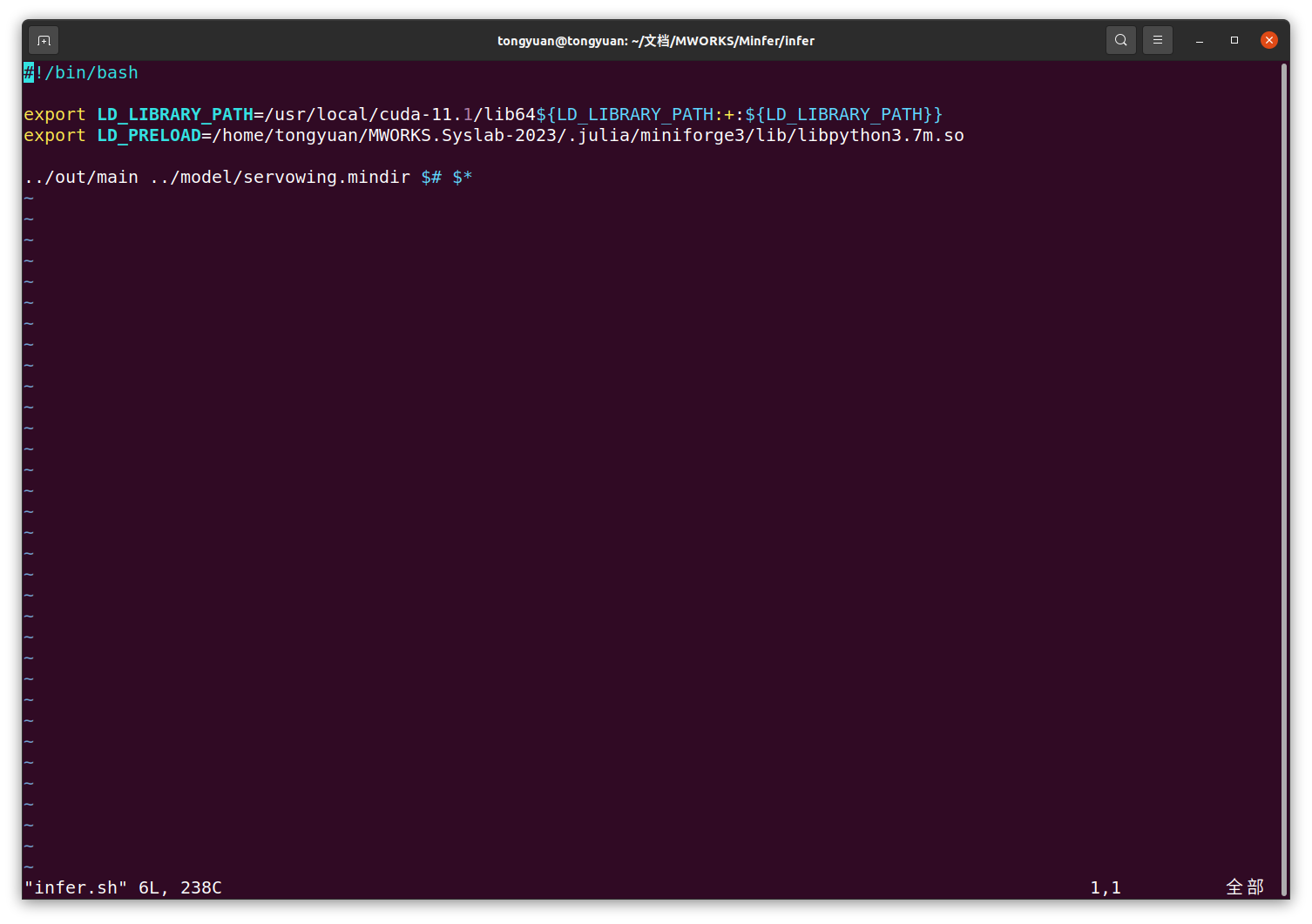


图3.14 模型推理(二)

该步骤为模型推理第二步，通过vim infer.sh打开脚本文件，使用i命令打开编辑模式，修改LD\_LIBRARY\_PATH和LD\_PRELOAD为用户自己的路径，并对模型名称进行修改，当前为servowing.mindir，用户可根据自己命名修改为特定的名称。修改完成后，Esc推出编辑模式，:wq保存退出脚本文件。

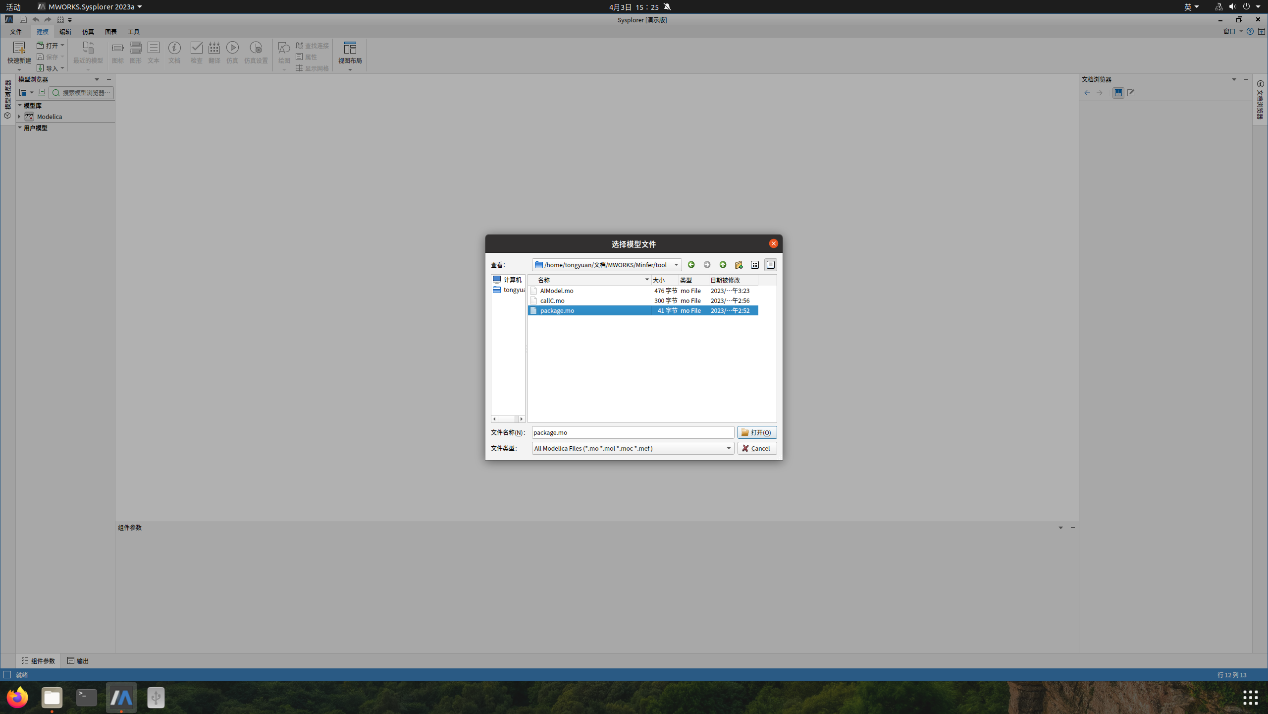


图3.15 加载模型(一)

该步骤在Sysplorer中进行，首先加载模型推理的modelica文件包，路径为/{Home}/文档/MWORKS/Minfer/tool/package.mo，{Home}为用户自己的主目录路径，当前显示为home/tongyuan。

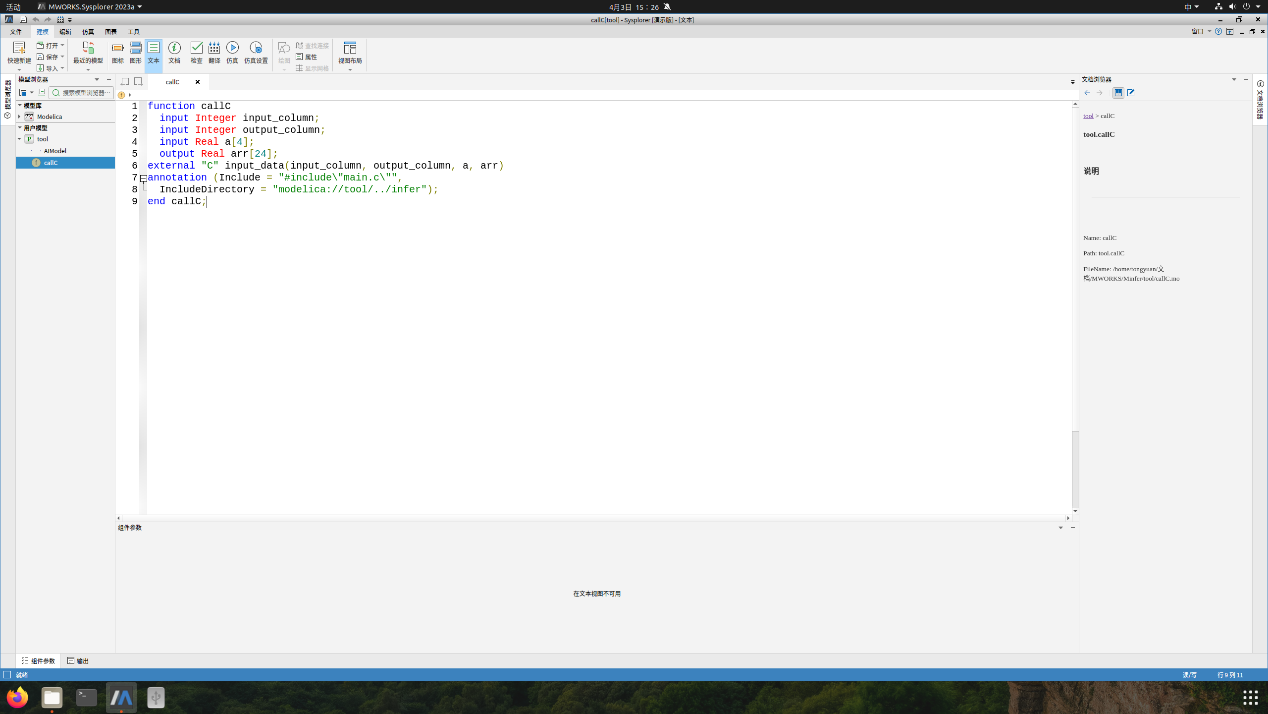


图3.16 加载模型(二)

选中callC函数，在菜单栏中选择文本模式，修改模型的输入输出维度，当前a[4]代表输入维度为4，arr[24]代表输出维度为24。

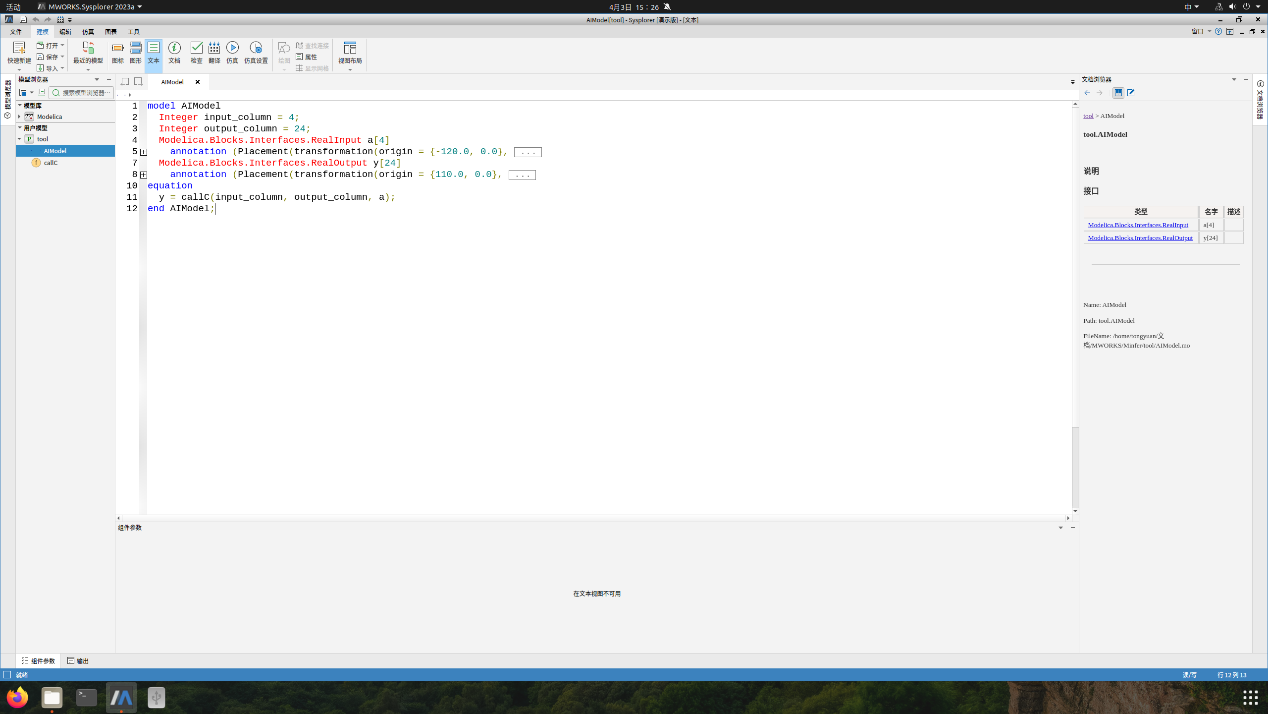


图3.17 加载模型(三)

同理修改AIModel中输入输出的维度，a[4]代表输入维度为4，y[24]代表输出维度为24，input\_column = 4代表输入维度为4，output\_column = 24代表输出维度为24。

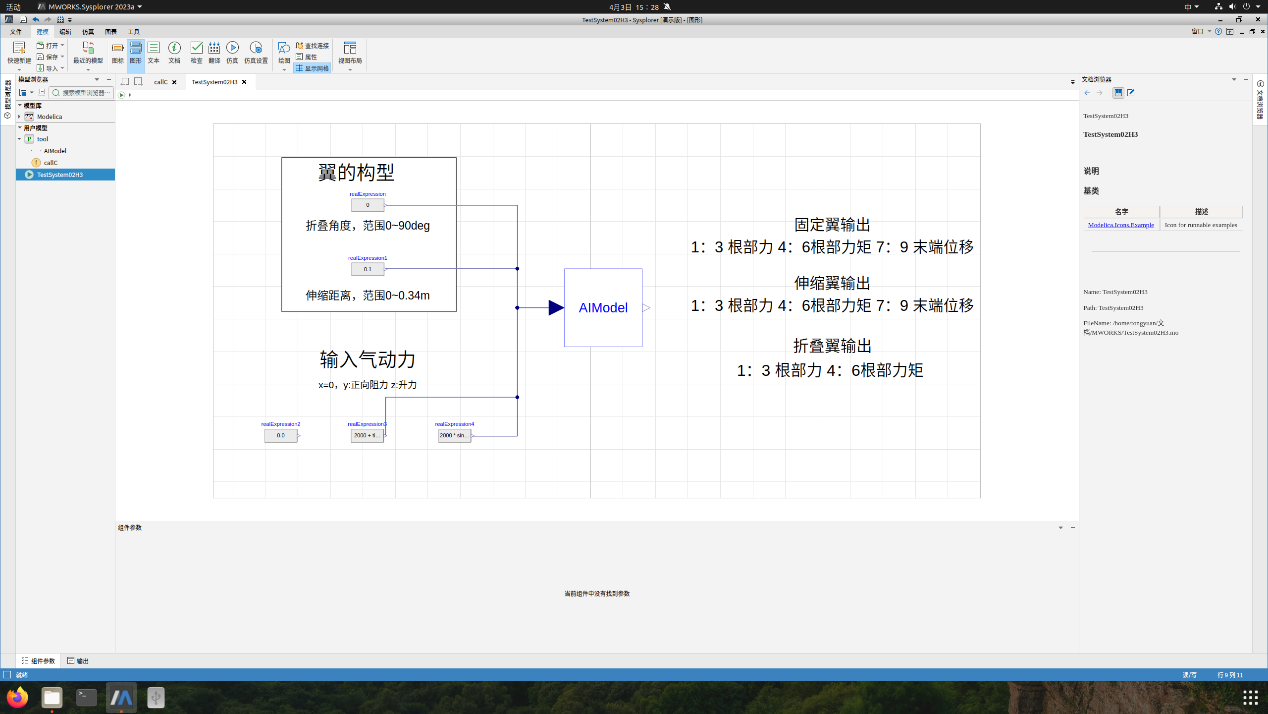


图3.18 构建用户模型(一)

上图为简易的伺服翼系统模型，模型名称为TestSystem02H3，该模型输入为四个，分别是翼的折叠角度、伸缩距离、输入气动力的y正向阻力、z升力，将信号数据传入AIModel中进行推理，推理结果分别为固定翼输出、伸缩翼输出和折叠翼输出，包含24个输出，需要对每条连线进行配置，如下图所示：

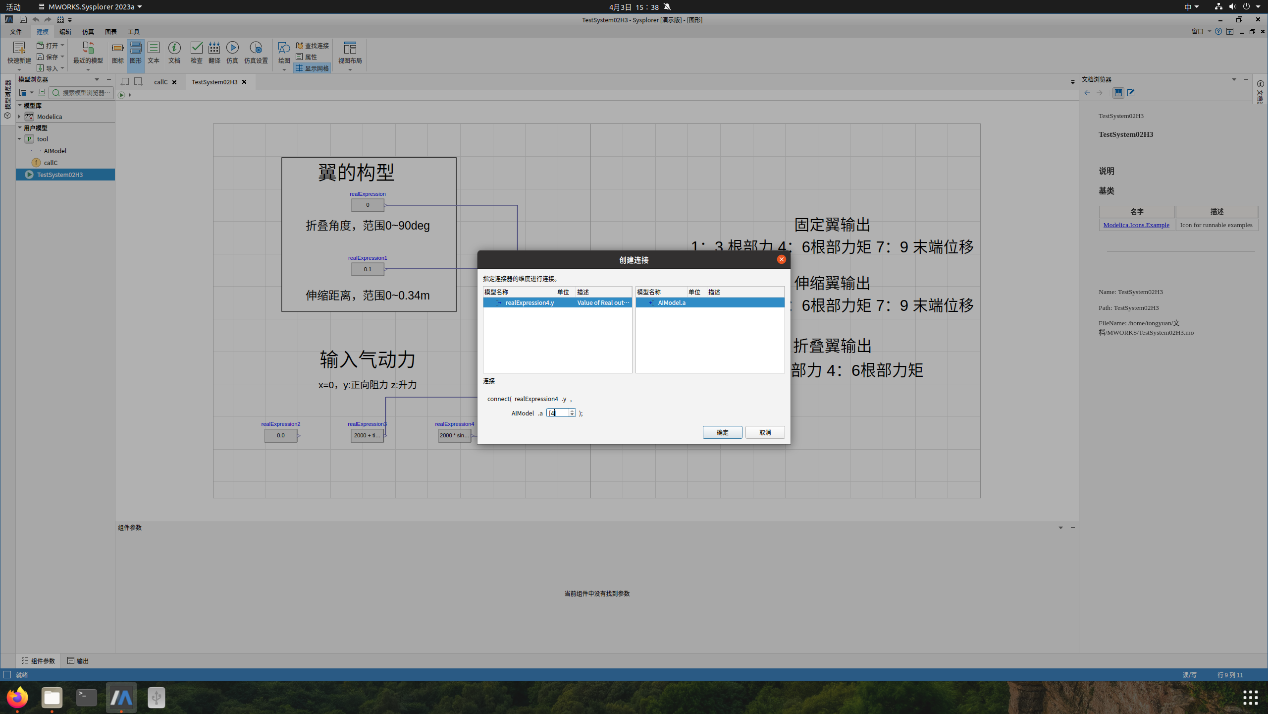


图3.19 构建用户模型(二)

构建完成后，设置仿真参数，即可开始进行仿真，仿真结果展示如下。

## 结果展示

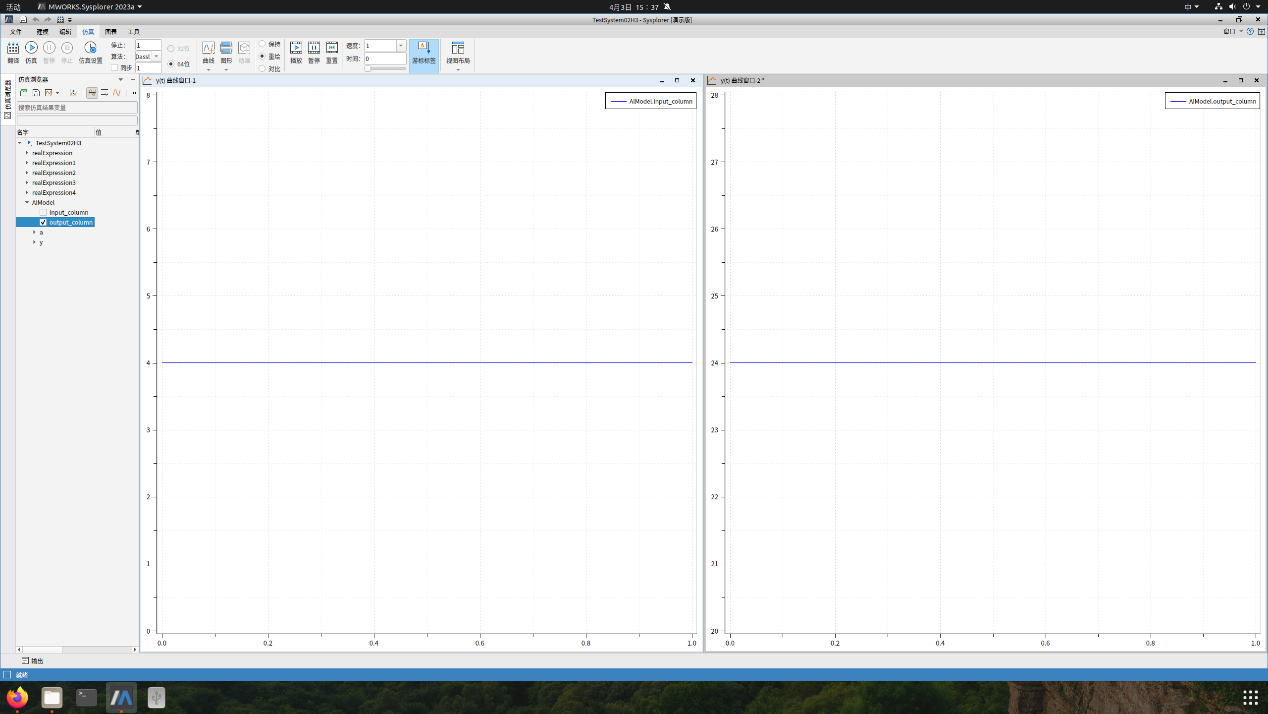


图3.20 模型输入输出维度

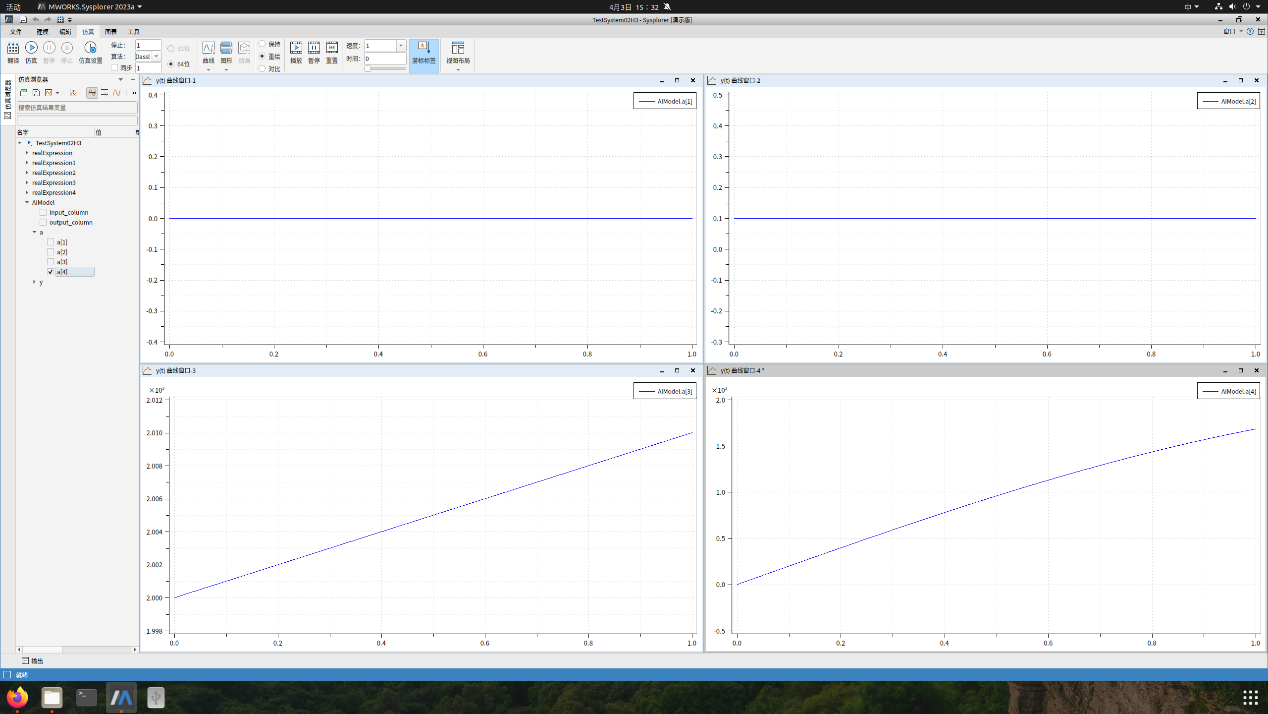


图3.21 伺服翼模型输入结果

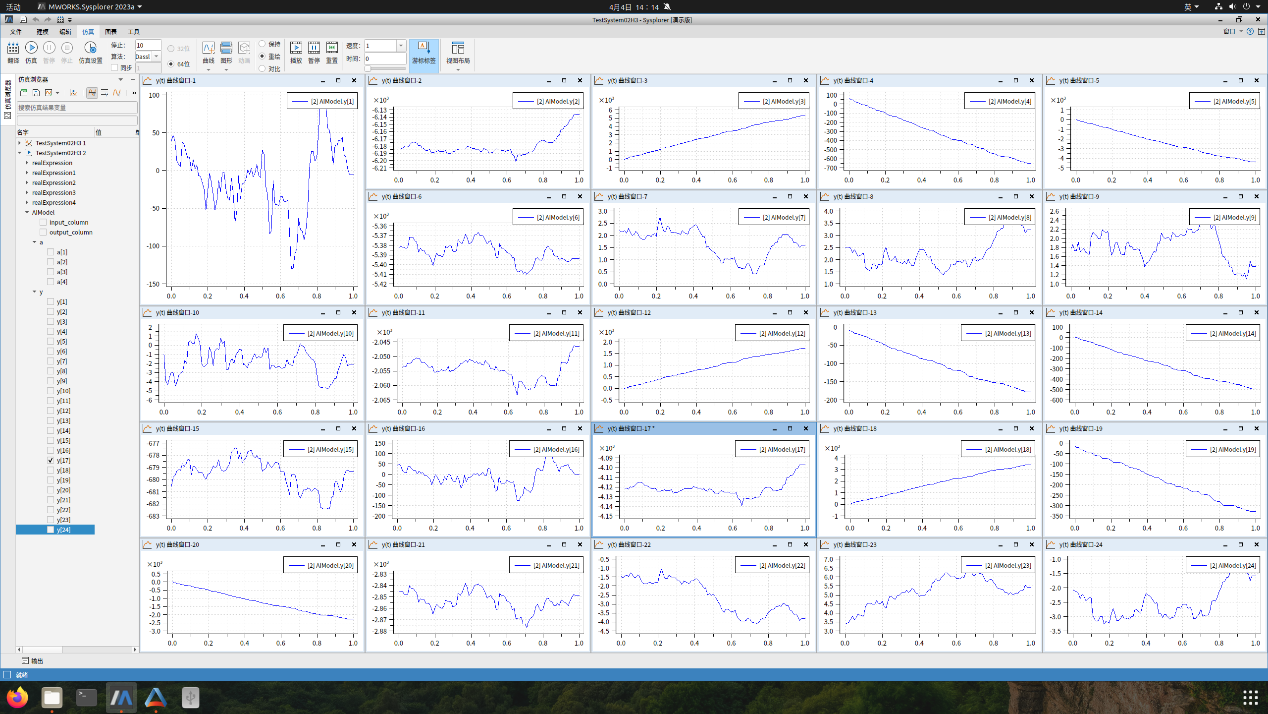


图3.22 AI模型推理结果

在联合仿真结束后，可对联合仿真的结果进行展示，图3.18为输入输出维度展示，输入为4维，输出为24维；图3.19为4个输入结果展示；图3.20为24个推理结果输出展示。