本章目录

平台架构与开发流程

第二章 实验平台配置

2.1 关键工具介绍
2.2 平台安装与首次仿真
2.3 平台架构与开发流程
2.4 API与SDK开发接口
2.5 例程体系与实验路线
2.6 习题与课堂讨论

本节 4 个教学单元

2.3.1 整体架构与核心组件
计算职责、闭环数据与观测来源

2.3.2 模型与算法开发
运动模型、控制程序与上层应用

2.3.3 仿真模式与实物迁移
软件、目标硬件、真实载具与集群迁移

2.3.4 对照测试与结果分析
参数对照、异常传播与结果解释

教材对应：2.2.1—2.2.2（架构与组件）、2.3.1—2.3.3（底层控制、故障测试、感知与集群）

2.3 平台架构与开发流程：本节导入

从运行系统到自己的开发任务

- **运行机制：**从运动目标出发，追踪控制器输出、模型运动和传感反馈，理解各软件怎样共同完成一次闭环计算。
- **开发位置：**改惯量属于运动模型，改控制律属于自驾仪程序，改识别方法属于上层应用；三者形成不同的运行产物。
- **验证路线：**模型测试观察运动规律，软件测试检查程序协作，目标硬件测试观察计算和通信时序。测试条件由研究对象决定。

定位对象 → **形成产物** → **接入闭环** → **对照结果**

任务示例：改惯量 → 生成DLL → CopterSim加载 → 对照角速度响应。

先定位开发对象，再确定产物、接入位置与验证方式。

2.3.1 整体架构与核心组件：整体框架

运行架构：四类计算共同形成闭环

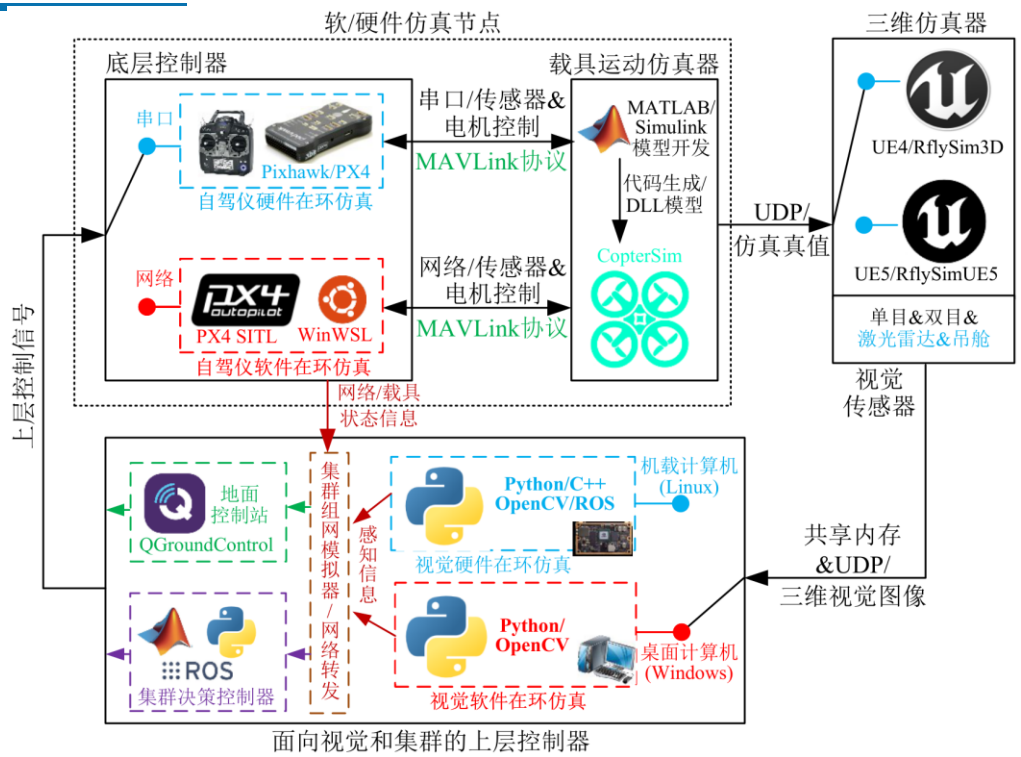


图2.9 RflySim工具链整体框架

- **载具运动仿真器：** CopterSim接收执行量，计算位姿与虚拟传感测量。
- **底层控制器：** 自驾仪融合测量，用目标与估计状态计算执行指令。
- **三维仿真器：** 按位姿更新场景，生成图像、深度等感知数据。
- **上层控制器：** 识别目标、规划运动，再将新目标交给自驾仪。

模型计算运动与测量，控制器依据反馈驱动下一步运动。

2.3 平台架构与开发流程

2.3.1 整体架构与核心组件：组件对应

软件窗口与后台计算一一对应



(a) CopterSim　(b) QGroundControl

(c) RflySim3D　(d) WinWSL/WslGUI

图2.11 核心软件界面预览

- **CopterSim**：加载动力学模型并显示状态；作为通信枢纽，连接自驾仪、三维端与上层程序。
- **QGroundControl**：接收自驾仪遥测，显示估计状态，配置参数与任务航线。
- **RflySim3D**：把位姿映射为场景运动，按传感配置输出图像等观测。
- **WinWSL**：提供Linux环境，运行PX4软件在环、ROS与OpenCV程序，并可编译PX4固件。

2.3.1 整体架构与核心组件：底层控制

底层闭环：执行量经模型变成新测量

闭环含义：当前测量返回控制器，参与计算下一轮执行输出。



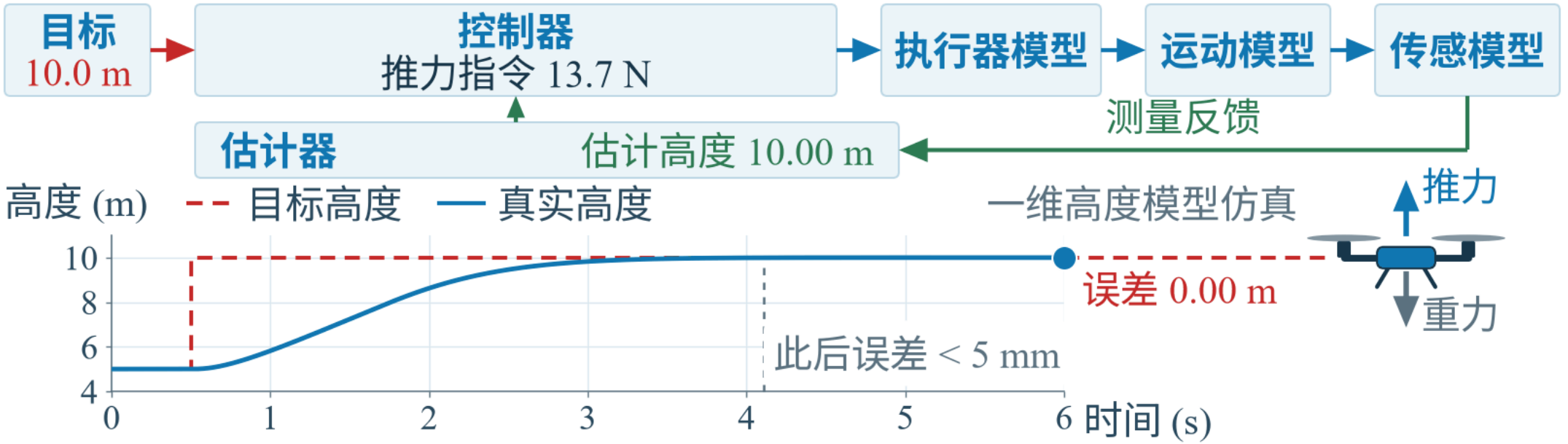
- **执行到运动：**电机或舵面模型把执行指令变成力和力矩；运动模型据此更新速度、位置与姿态。
- **测量到控制：**传感模型生成测量，估计器融合为状态；控制器根据目标与估计状态之差修正下一轮输出。

执行器改变运动，传感测量返回估计器，控制器据此修正输出。

2.3.1 整体架构与核心组件：底层控制

底层闭环：逐拍迭代与误差收敛

阶跃后约 900 拍（3.6 s）：误差 < 5 mm，推力回到悬停值 13.7 N

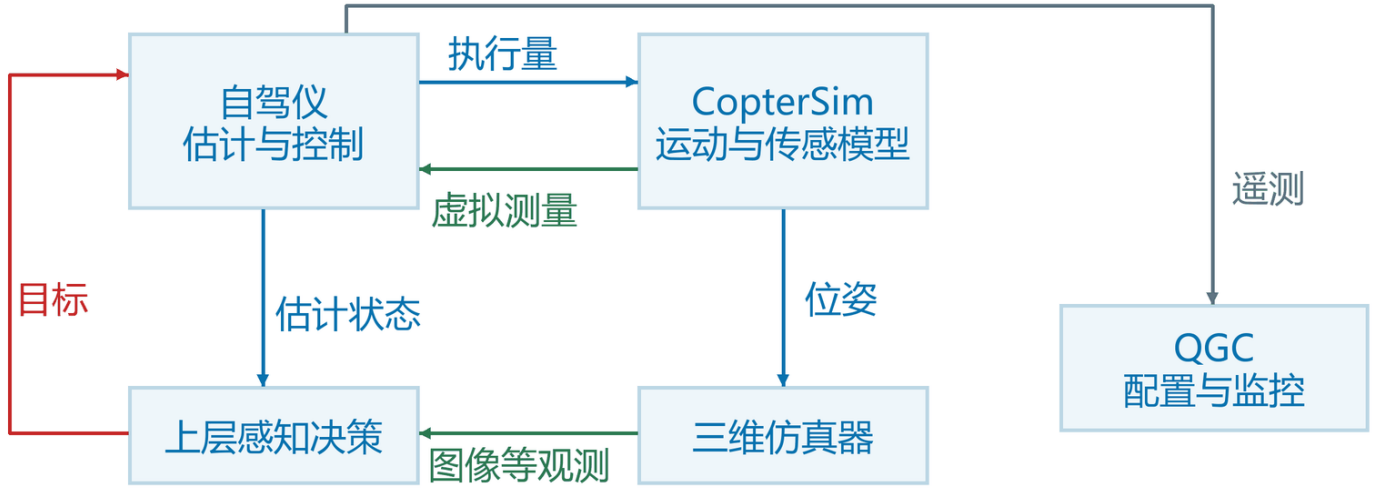


图：底层闭环逐拍收敛（一维高度简化数值仿真）

测量反馈参与下一拍计算，本示例的高度误差逐渐收敛。

2.3.1 整体架构与核心组件：感知决策

上层闭环：场景观测产生新的运动目标



图：底层控制与上层感知决策双回路

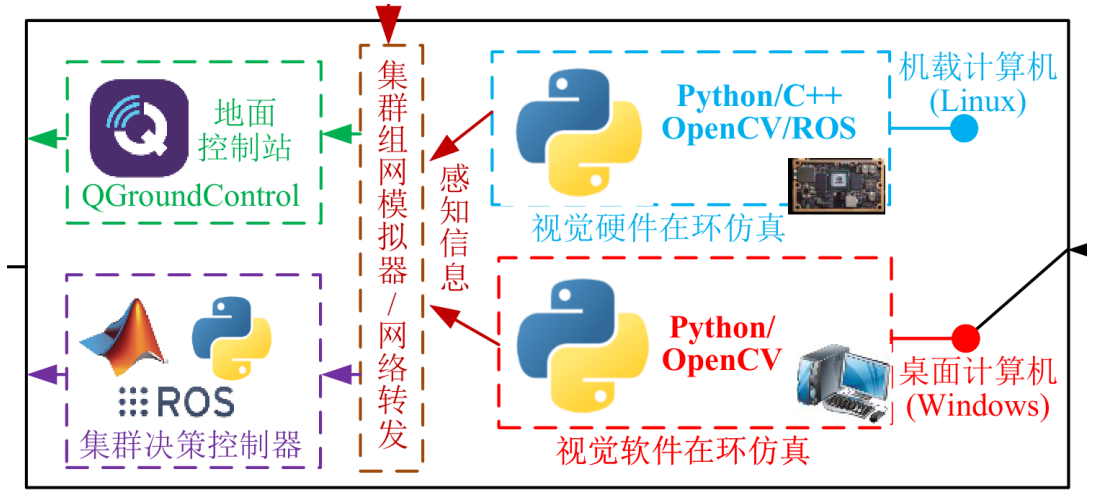
运动反馈：测量进入自驾仪，模型位姿驱动场景。

任务反馈：算法识别目标，规划结果返回自驾仪。

底层控制维持运动，上层感知决策通过目标接口影响运动。

2.3.1 整体架构与核心组件：上层控制器

上层控制器：单机感知决策、组网通信与集群控制



面向视觉和集群的上层控制器

图2.9 RflySim工具链整体框架（局部）

- **感知决策层：** 处理本机图像与点云，完成目标检测跟踪、定位建图、避障与任务决策。
- **组网通信层：** 集群组网模拟器在多机间转发状态与感知信息。
- **集群控制层：** 地面控制站与集群决策控制器组织多机协同，可采用集中式或分布式（下页）。

输入输出： 输入三维端的图像、点云和自驾仪的状态估计；输出解锁、起飞、期望速度或轨迹等指令。

上层控制器把观测变成运动目标，多机时经组网通信共享信息、协同控制。

2.3.1 整体架构与核心组件：集群控制

集群控制可采用集中式或分布式架构

比较项	集中式	分布式
决策位置	地面站汇总全部状态， 统一规划轨迹与任务	各节点独立运行决策逻辑， 按本机与邻机信息生成指令
优势与代价	便于全局统一优化； 依赖通信，易受干扰	减少中心依赖，扩展较灵活； 仍需满足协同与通信条件。
适用场景	精度要求高、 规模较小的任务	大规模集群或 通信受限的环境

选择依据：任务规模、通信条件与实时性要求；两种架构也可结合使用。

分布式减少对中心节点的依赖，也把更多协调工作交给机间通信。

2.3.1 整体架构与核心组件：模型分工

动力学、三维外观与传感器各有职责

对象	决定的行为	改动示例
动力学模型	执行量 → 受力、运动 与模拟测量	改变惯量，响应随模型方程变化
三维模型 与场景	几何、材质、环境 及可见外观	换外形，不自动改变动力学参数
传感器配置	安装位姿、视场、 分辨率及数据输出	转动相机，观测范围随之改变

例：CopterID标识实例，ClassID选三维外形，二者与动力学模型须对应同一载具。

2.3.1 整体架构与核心组件：数据来源

同一次运动，真值、观测与估计不同

数据	产生位置	主要用途
仿真真值	运动或场景模型中的状态	驱动显示，作为误差分析参照
传感观测	传感模型及其采样过程	模拟算法在给定条件下读到的数据
状态估计	自驾仪或上层估计器	融合观测，供控制与任务程序使用

例：模型位置 → 含噪定位测量 → 融合后的估计位置。

测量由模型状态产生，估计融合测量；同源运动可以得到不同数值。

本章目录

平台架构与开发流程

第二章 实验平台配置

2.1 关键工具介绍
2.2 平台安装与首次仿真
2.3 平台架构与开发流程
2.4 API与SDK开发接口
2.5 例程体系与实验路线
2.6 习题与课堂讨论

本节 4 个教学单元

2.3.1 整体架构与核心组件
计算职责、闭环数据与观测来源

2.3.2 模型与算法开发
运动模型、控制程序与上层应用

2.3.3 仿真模式与实物迁移
软件、目标硬件、真实载具与集群迁移

2.3.4 对照测试与结果分析
参数对照、异常传播与结果解释

教材对应：2.2.1—2.2.2（架构与组件）、2.3.1—2.3.3（底层控制、故障测试、感知与集群）

2.3.2 模型与算法开发：工具与产物

开发成果有不同的构建与运行位置

开发与构建		交付产物		运行位置
运动建模 Simulink模型与参数	→	**模型DLL** 动态链接库及参数	→	**CopterSim** 运动与模拟测量
控制设计 Simulink或C/C++模块	→	**控制软件／固件** 生成、集成并构建	→	**自驾仪** 估计、控制与调度
感知决策 Python／C++／ROS	→	**应用程序与配置** 程序、权重和接口配置	→	**桌面／机载计算机** 观测处理与任务目标

运行示例：CopterSim直接加载模型DLL；修改模型后重新构建，再加载新文件。

2.3 平台架构与开发流程

2.3.2 模型与算法开发：开发总览

开发流程连接原型、运行产物与实物

- **模型与控制（1–3）：** 在Simulink中建立模型与控制器，闭环仿真，按跟踪误差与超调调参。
- **代码生成（4）：** 控制器代码编入PX4软件，运动模型生成DLL交给CopterSim。
- **硬件仿真（5）：** 程序下载到自驾仪，经USB与CopterSim交换传感数据和控制信号。
- **真机实验（6）：** 装机后依次做台架、室内和外场测试，再按实物响应调参。

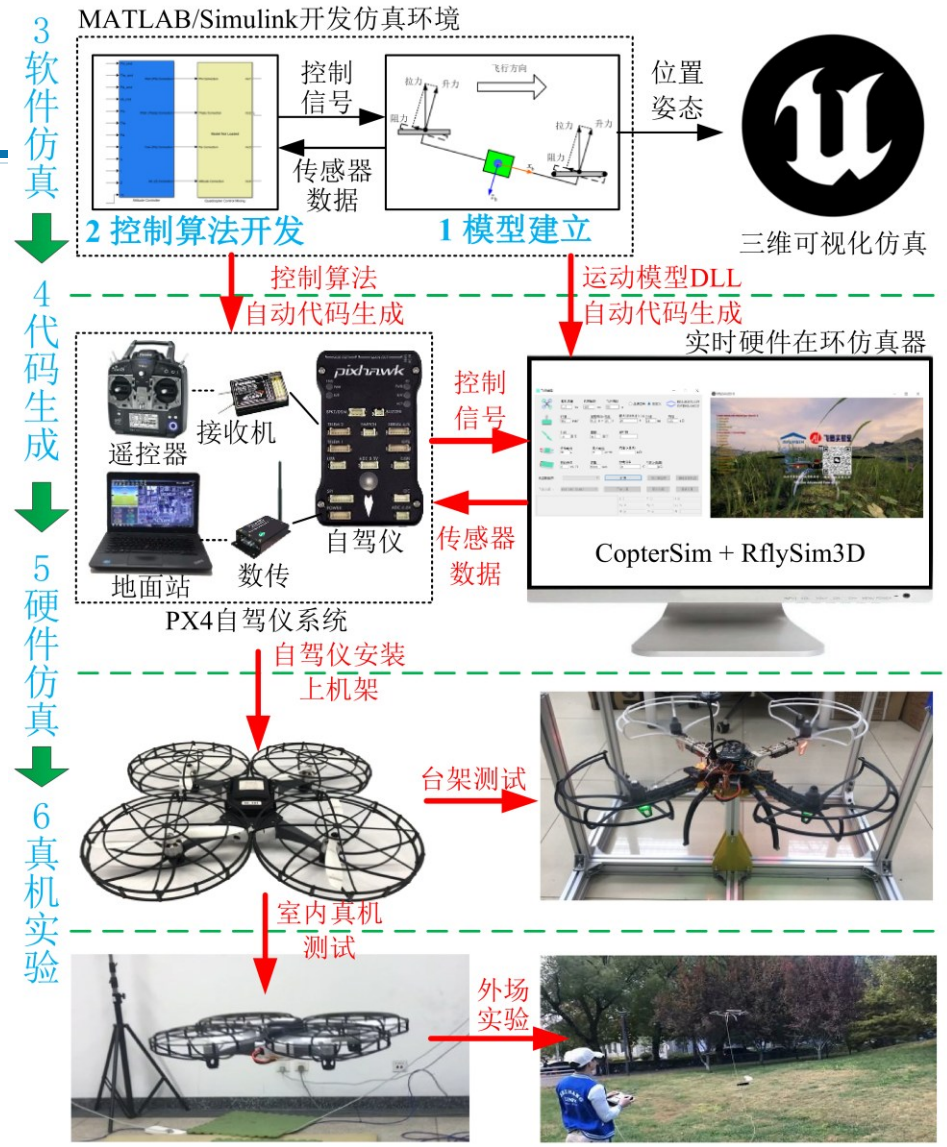


图2.17 底层控制算法开发到实验流程

先用模型验证设计，再把对应产物接入软件、硬件与实物测试。

2.3.2 模型与算法开发：开发内容

运动、场景、控制与程序集成分别开发

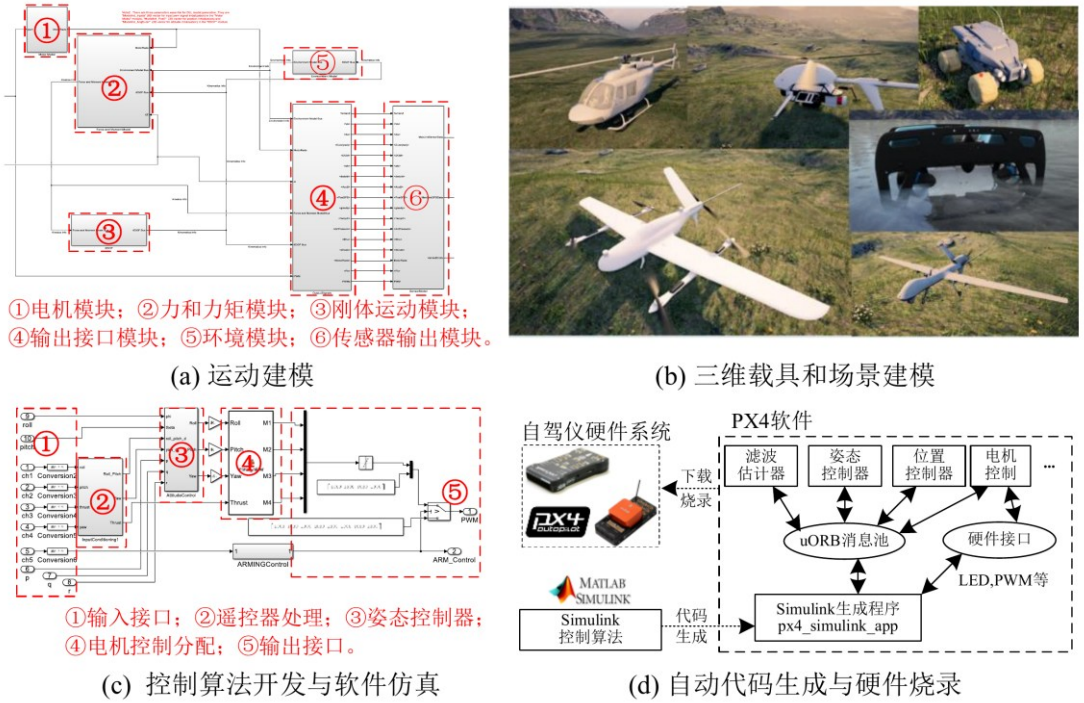


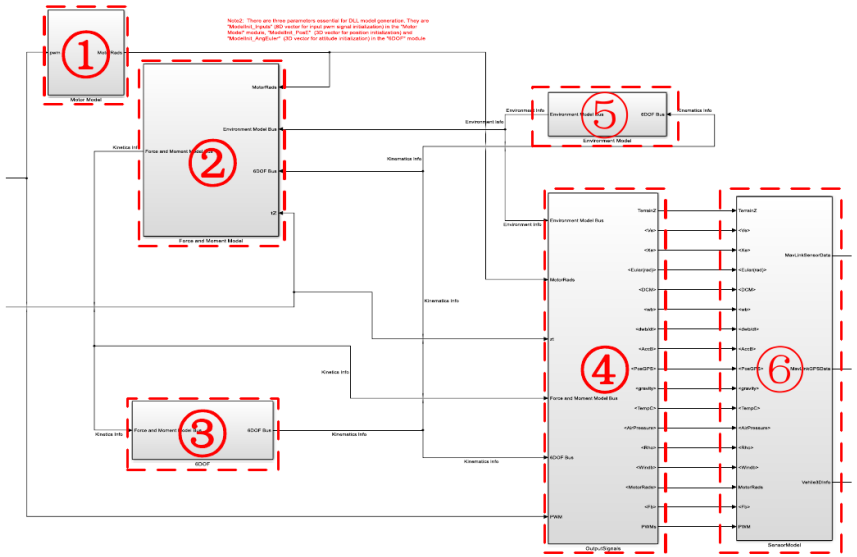
图2.18 底层控制算法开发关键过程图

- **运动建模(a)**：描述执行器、受力和运动方程，输出载具状态与模拟测量。
- **场景建模(b)**：配置几何、材质与环境，形成三维显示和传感观测。
- **控制设计(c)**：由目标和估计状态计算控制量，再分配到各执行器。
- **程序集成(d)**：连接算法输入输出，生成代码并编入目标软件或固件。

四项工作分别形成运动、观测和控制能力，再通过接口组成系统。

2.3.2 模型与算法开发：运动模型

运动模型把执行输入变成状态与观测



①电机模块；②力和力矩模块；③刚体运动模块；
④输出接口模块；⑤环境模块；⑥传感器输出模块。

图2.18(a) 运动建模

- **执行与受力：**电机①产生转速，力和力矩②合成拉力、重力等。
- **运动与环境：**③更新位姿与速度，环境⑤按高度给出空气密度等。
- **输出与运行：**接口④打包状态，⑥生成传感测量；编译成DLL后由CopterSim周期调用。

对应例程：

e2-UavModeling（动态建模）

e2_DLL-Load（DLL模型导入）

模型方程决定响应；编译后的DLL把同一模型带入CopterSim。

2.3.2 模型与算法开发：场景与感知

场景观测为感知算法提供可控输入



图2.18(b) 三维载具与场景建模

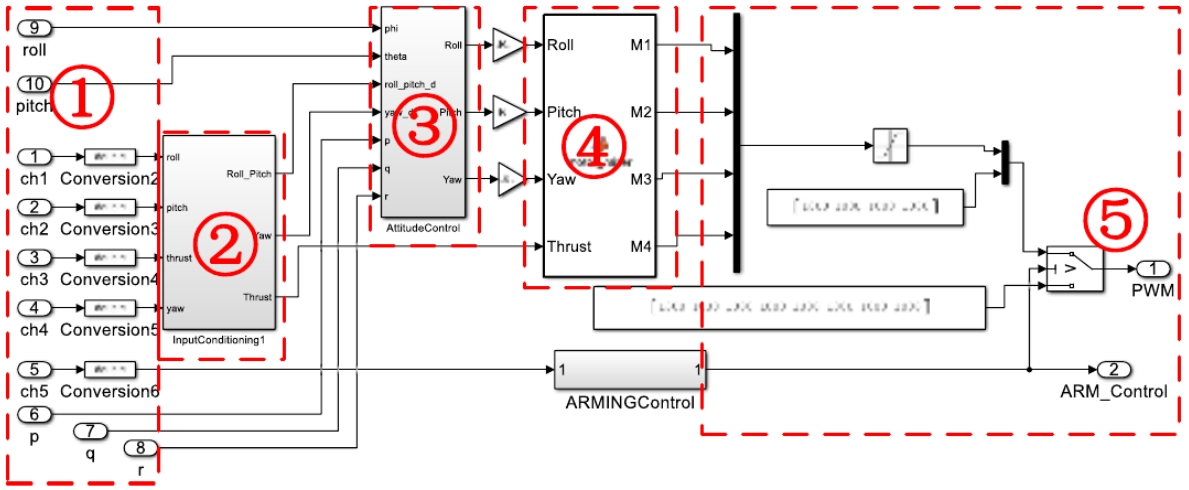
- **可控环境：**几何、材质、光照和遮挡共同决定传感器看到的内容。
- **多模态输入：**按传感配置输出图像、深度和点云，供识别与定位。
- **闭环接入：**识别结果进入决策程序，形成位置或速度目标，由自驾仪执行。
- **对照示例：**固定轨迹与控制器，只改遮挡，比较漏检率和任务完成。

对应例程：e2-UavModeling
（在RflySim3D中添加三维模型）

固定控制与运动模型，改变场景或传感条件，就能对照感知算法。

2.3.2 模型与算法开发：控制设计

控制算法从目标与反馈计算执行量



①输入接口；②遥控器处理；③姿态控制器；④电机控制分配；⑤输出接口。

图2.18(c) 控制算法开发与软件仿真

- **输入与目标：**①接收遥控与状态，②把遥控通道变为期望姿态。
- **姿态控制：**③用比例-积分-微分（PID）控制，由期望与估计姿态之差计算力矩。
- **分配输出：**④把力矩与总拉力分给各电机，⑤封装为执行指令，解锁后生效。

对应例程：e5-AttitudeCtrl（姿态控制器设计实验）

目标进入反馈控制，再经执行器分配转成各通道的指令。

2.3.2 模型与算法开发：程序集成

控制程序集成后参与自驾仪闭环

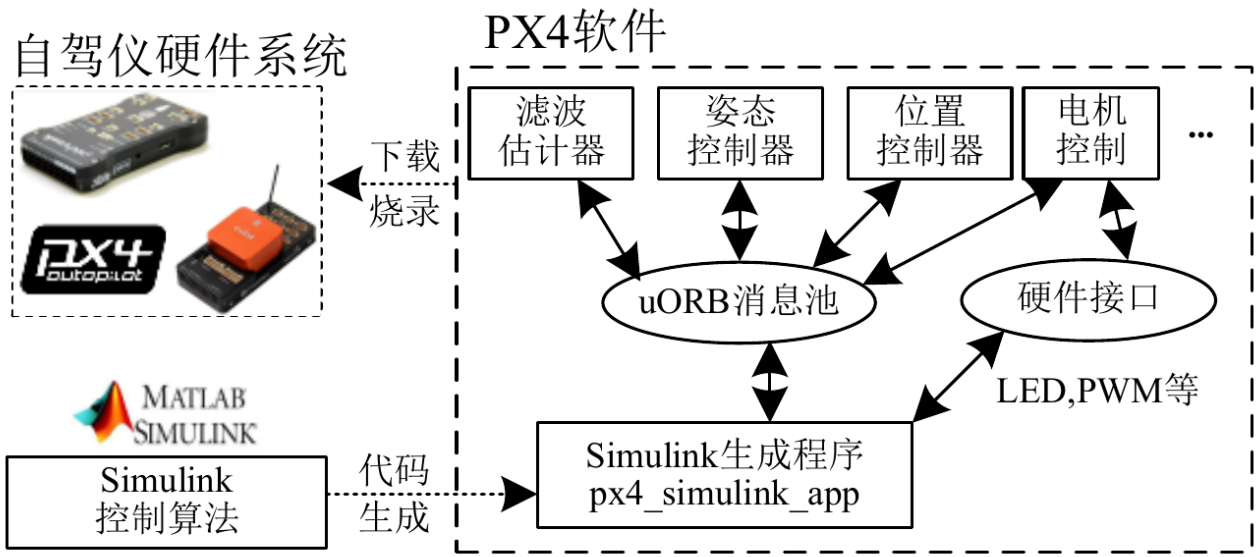


图2.18(d) 自动代码生成与硬件烧录

- **消息连接：** PX4模块经微对象请求代理器（uORB）订阅输入、发布结果。
- **目标构建：** 生成的C/C++代码编入软件或固件，运行目标须匹配仿真环境或控制板。
- **控制接替：** px4_simulink_app接管执行器输出，替代原姿态或速度控制器。

对应例程：e7_Code-Generation（MATLAB自动生成飞控固件实验）

生成程序编入目标软件并取得对应输出职责后，算法才进入运行闭环。

本章目录

平台架构与开发流程

第二章 实验平台配置

2.1 关键工具介绍
2.2 平台安装与首次仿真
2.3 平台架构与开发流程
2.4 API与SDK开发接口
2.5 例程体系与实验路线
2.6 习题与课堂讨论

本节 4 个教学单元

2.3.1 整体架构与核心组件
计算职责、闭环数据与观测来源

2.3.2 模型与算法开发
运动模型、控制程序与上层应用

2.3.3 仿真模式与实物迁移
软件、目标硬件、真实载具与集群迁移

2.3.4 对照测试与结果分析
参数对照、异常传播与结果解释

教材对应：2.2.1—2.2.2（架构与组件）、2.3.1—2.3.3（底层控制、故障测试、感知与集群）

2.3.3 仿真模式与实物迁移：模式总览

仿真模式由真实与虚拟的替换位置区分

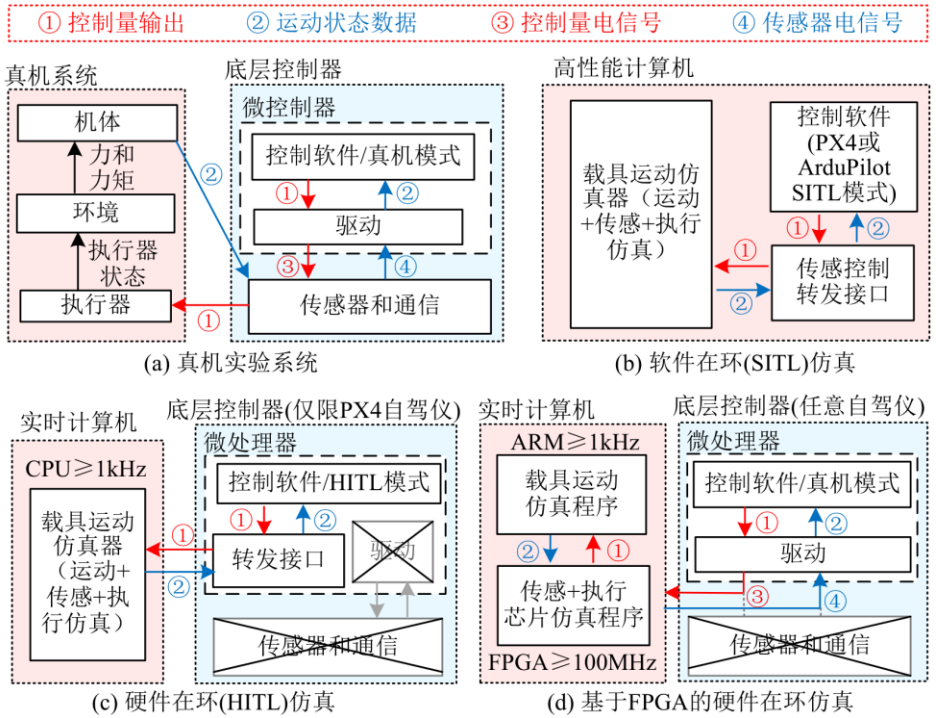


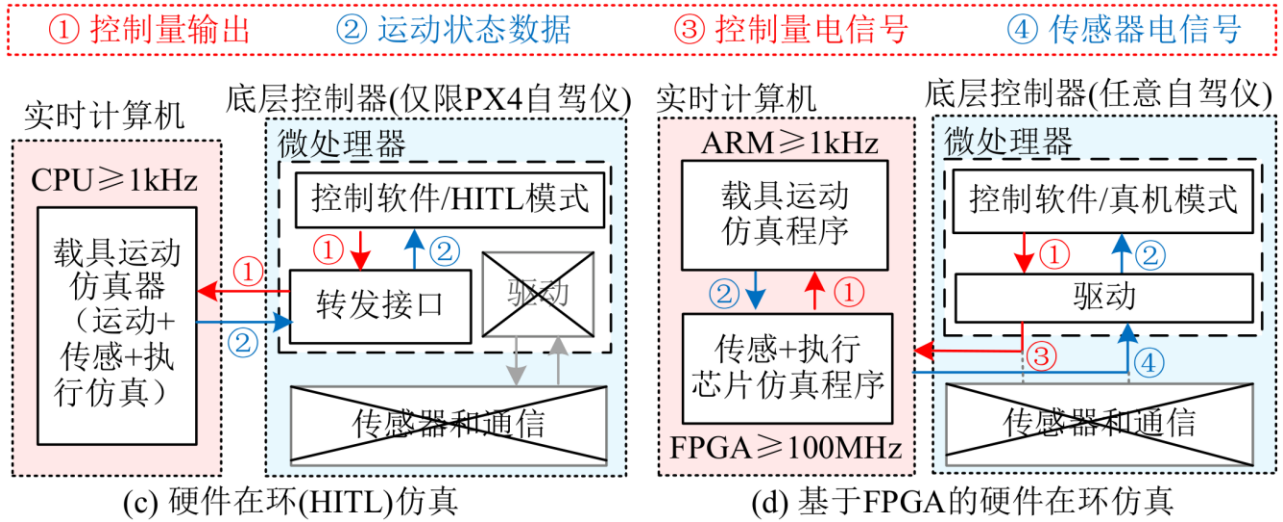
图2.10 RflySim支持的仿真模式

- **真实试验：**真实传感器测量实物运动，输出驱动真实执行器。
- **软件在环：**控制软件与载具模型都在计算机中，经网络交换消息。
- **硬件在环：**PX4运行于真实自驾仪，经USB串口收测量、回传执行量。
- **芯片接口模拟：**
由现场可编程门阵列（FPGA）
按原协议、电平与时序
模拟传感芯片，驱动照常参与。

软件在环运行控制软件，硬件在环加入目标硬件，实物测试加入真实载具。

2.3.3 仿真模式与实物迁移：替换位置

硬件在环与FPGA方案：在哪一层替换真实信号



(c) 硬件在环(HITL)仿真　　(d) 基于FPGA的硬件在环仿真

图2.10(c)(d) RflySim支持的仿真模式（局部放大）

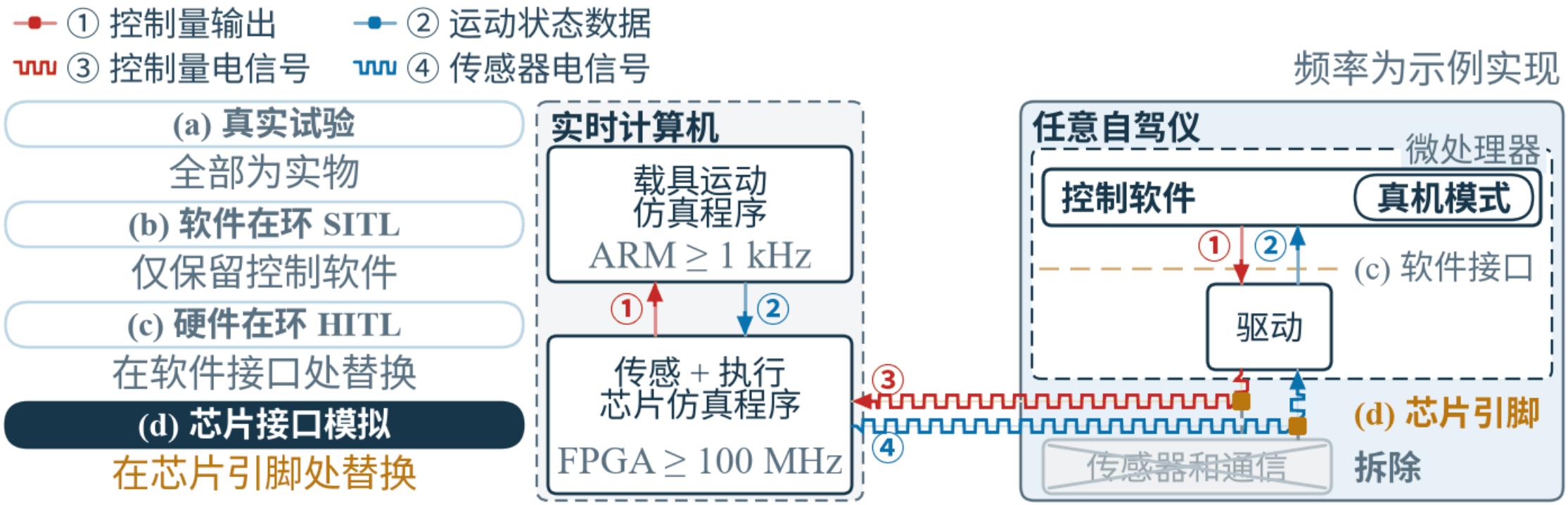
读图：①②是软件数据，③④是引脚电信号；
打叉框是被替代的真实硬件。

- **硬件在环(c)：**HITL模式下，转发接口绕过驱动，经USB串口与仿真器收发①②；PX4原生支持该模式。
- **FPGA方案(d)：**保持真机模式，驱动照常，拆除传感芯片，由FPGA读取③、生成④；芯片接口适配后，可不改固件。
- **代价：**要按芯片型号和通信校验逻辑建立芯片模型。

硬件在环在软件层替换测量与执行，FPGA方案在芯片引脚层替换电信号。

2.3.3 仿真模式与实物迁移：替换位置

仿真模式：真实信号在哪一层被替换



图：四种模式的替换位置（依据图2.10重绘的原理示意）

硬件在环替换软件数据，FPGA方案替换引脚电信号。

2.3.3 仿真模式与实物迁移：模式选择

模型、软件和硬件验证不同层次的实现

验证方式	闭环中的被测实现	新增的验证内容
模型在环 **MIL**	控制模型与载具模型 在建模环境中闭环	验证控制律、响应曲线 与数值求解行为
软件在环 **SIL / SITL**	实际控制软件与载具模型 在计算机中运行	检验模式切换、模块集成 和消息数据交互
硬件在环 **HIL / HITL**	目标控制硬件接入模型， 以虚拟测量驱动	测量板上计算周期、运行负载 与通信时序

SITL与HITL运行同一套PX4控制逻辑，差别在处理器、时序与接口。

算法关系、软件集成和板上时序，分别需要对应层次的验证。

2.3.3 仿真模式与实物迁移：硬件闭环

HITL由虚拟测量和执行量往返构成

消息	方向与数据	示例频率
HIL_SENSOR	模型 → 自驾仪 惯性、气压等测量	≥250 Hz
HIL_GPS	模型 → 自驾仪 定位测量	≥10 Hz
HIL_ACTUATOR_CONTROLS	自驾仪 → 模型 16路归一化执行量	≥200 Hz
HEARTBEAT	自驾仪 → 模型 系统与模式状态	1 Hz

表2.2 CopterSim硬件在环协议

传感测量送入自驾仪，执行量返回模型，心跳报告运行状态。

2.3.3 仿真模式与实物迁移：机载部署

机载程序可以先接仿真输入，再接真实输入

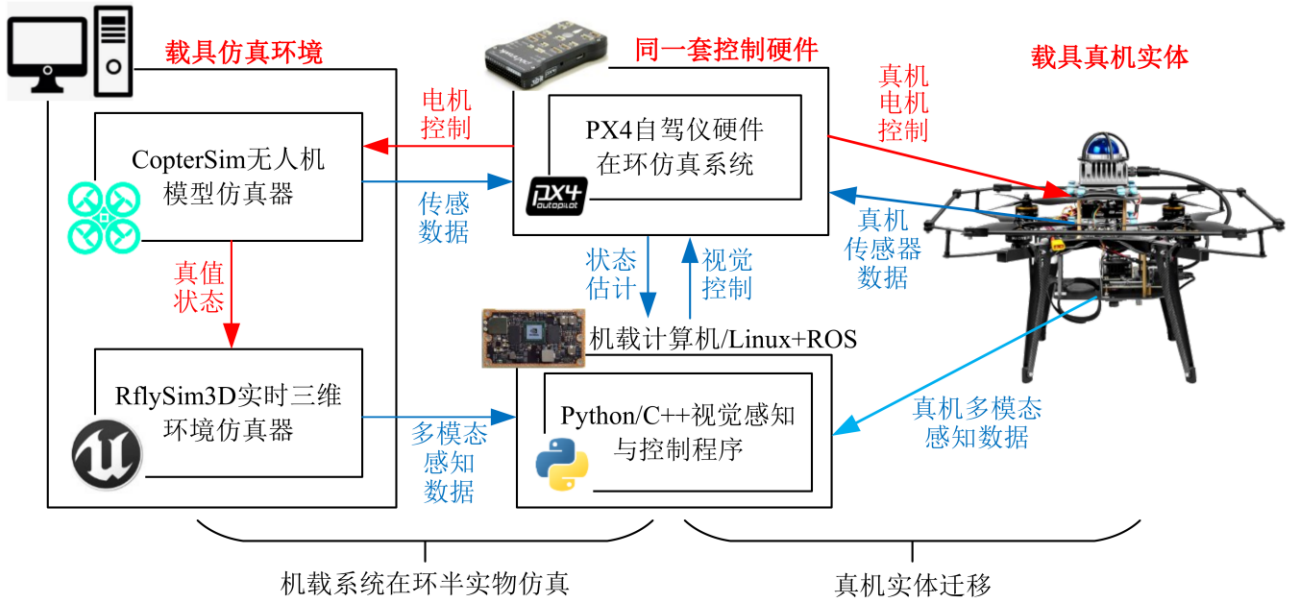


图2.20 机载系统硬件在环仿真

- **保留硬件：** 自驾仪与机载计算机先接仿真闭环，观察同一算法的板上计算与通信。
- **切换实物：** 模拟测量和图像改由真实传感器提供，执行量转接电机或舵面，显现噪声与时延差异。
- **部署条件：** 程序匹配处理器架构、系统库与驱动，时间戳和坐标定义保持一致。

保留控制硬件与算法程序，替换传感输入和执行对象，可以比较迁移差异。

2.3.3 仿真模式与实物迁移：实物验证

实物测试逐步加入真实传感、执行与环境

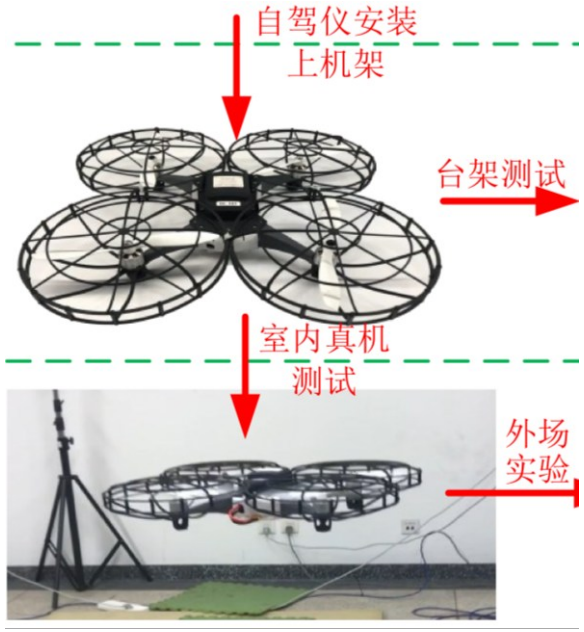


图2.17（局部）
真机实验阶段



安装与台架：接入电机、舵面和结构，观察输出方向、执行响应与振动。

任务与环境：在受控范围比较轨迹误差，观察传感噪声和扰动的影响。

台架暴露执行与结构差异，任务试验进一步暴露环境与运动差异。

2.3.3 仿真模式与实物迁移：虚实组合

虚实协同：部分载具虚拟，控制硬件仍真实

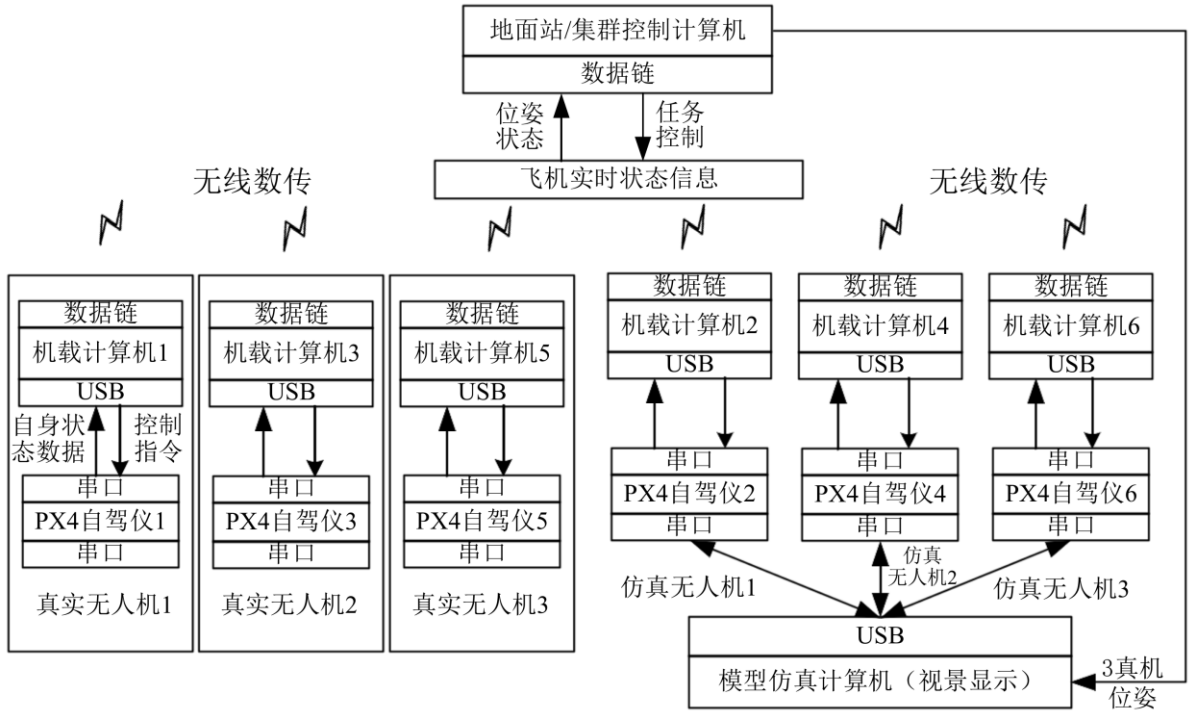


图2.21 固定翼半实物仿真实验架构图

- **真实硬件：**六套自驾仪、机载计算机和无线数传都是实物。
- **三实三虚：**3架真机实际飞行，另3架由模型计算运动与测量；真机位姿经地面站回传三维场景。
- **同步与测试：**时钟同步和运动预测抵消传输延迟；虚拟飞机主动靠近真机，检验密集编队中的避障与安全响应。

虚拟载具可与真机共享任务状态，真实硬件和运动仍受实际条件约束。

2.3.3 仿真模式与实物迁移：集群开发阶段

感知决策与集群控制分阶段迁移到真机

①原型验证与机载移植 → ②集群控制简化感知模型 / ③单机决策精细感知模型 → ④虚实结合逐步加真机 → ⑤良好条件真机集群 → ⑥恶劣或故障条件真机

- **解耦验证：**集群控制用简化感知模型，直接取三维引擎给出的目标位姿真值，省去图像生成与处理；单机决策用图像、点云等精细感知模型；两类可并行开发。
- **逐步换真：**先让算法在真实自驾仪与机载计算机上处理仿真数据（图2.20），再从“1实5虚”起逐步增加真机比例（图2.21为3实3虚）。
- **真机两步：**先在天气稳定、通信畅通、定位准确时检验流程与参数，再在恶劣天气、干扰或注入故障时检验安全模式与应急处置。

集群控制与精细感知可并行验证，再逐步接入真机检验。

本章目录

平台架构与开发流程

第二章 实验平台配置

2.1 关键工具介绍
2.2 平台安装与首次仿真
2.3 平台架构与开发流程
2.4 API与SDK开发接口
2.5 例程体系与实验路线
2.6 习题与课堂讨论

本节 4 个教学单元

2.3.1 整体架构与核心组件
计算职责、闭环数据与观测来源

2.3.2 模型与算法开发
运动模型、控制程序与上层应用

2.3.3 仿真模式与实物迁移
软件、目标硬件、真实载具与集群迁移

2.3.4 对照测试与结果分析
参数对照、异常传播与结果解释

教材对应：2.2.1—2.2.2（架构与组件）、2.3.1—2.3.3（底层控制、故障测试、感知与集群）

2.3.4 对照测试与结果分析：对照设计

参数对照从固定基线开始

- **固定基线：** A、B采用相同目标输入、初始状态、控制参数与记录时长，避免把任务变化误认为模型参数的影响。
- **明确改动：** 只改变偏航惯量，分别保存模型版本与实际加载参数；控制器、其他模型参数和数据来源保持一致。
- **解释响应：** 同步记录目标角速度、实际角速度与执行量，对照上升过程、峰值、稳态误差及收敛情况。

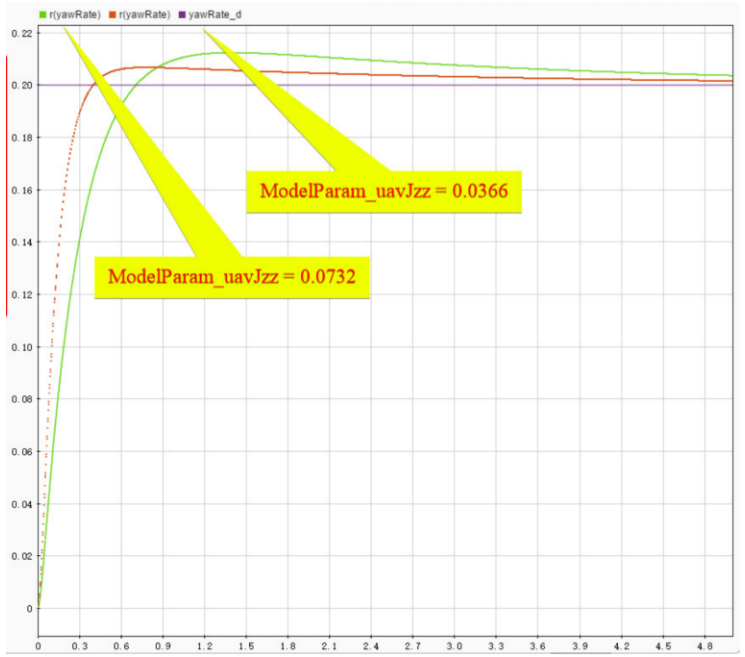
保留基线A → **只改惯量B** → **运行同一任务** → **比较响应**

记录归属： 模型版本＋实际参数＋输入条件＋日志，使每组曲线对应一次运行。

只改变研究对象，保留相同输入、初态和记录条件，才能解释响应差异。

2.3.4 对照测试与结果分析：结果解释

响应曲线说明模型改动产生了什么影响



图：e2-UavModeling例程的偏航角速度对照

- **同一输入：**目标为0.2 rad/s，初态和控制参数相同，使两组响应可比。
- **改变惯量：**A（橙线）为0.0366，B（绿线）加倍为0.0732 kg·m²。
- **曲线结论：**B组上升更慢、峰值略高（0.212对0.207），均趋近目标。

动态建模与参数对照：https://rflysim.com/doc/zh/2.Course/ch5/index.html

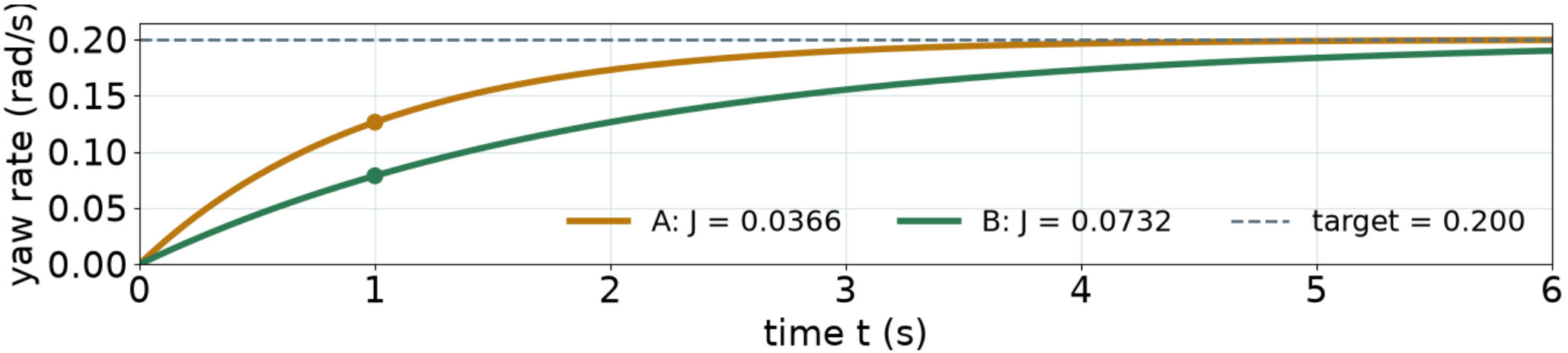
同一目标下比较响应速度、峰值与误差，并把结论限定在对应模型和配置。

2.3.4 对照测试与结果分析：可复算算例

同一力矩下，偏航惯量加倍使响应变慢

$$J\dot{\omega} + b\omega = \tau, \quad \tau = b\omega_{\mathrm{ref}}, \quad \omega(t) = \omega_{\mathrm{ref}}(1 - e^{-bt/J})$$

静止起步；目标0.2 rad/s；阻尼0.0366 N·m·s/rad；惯量A/B：0.0366/0.0732 kg·m²。



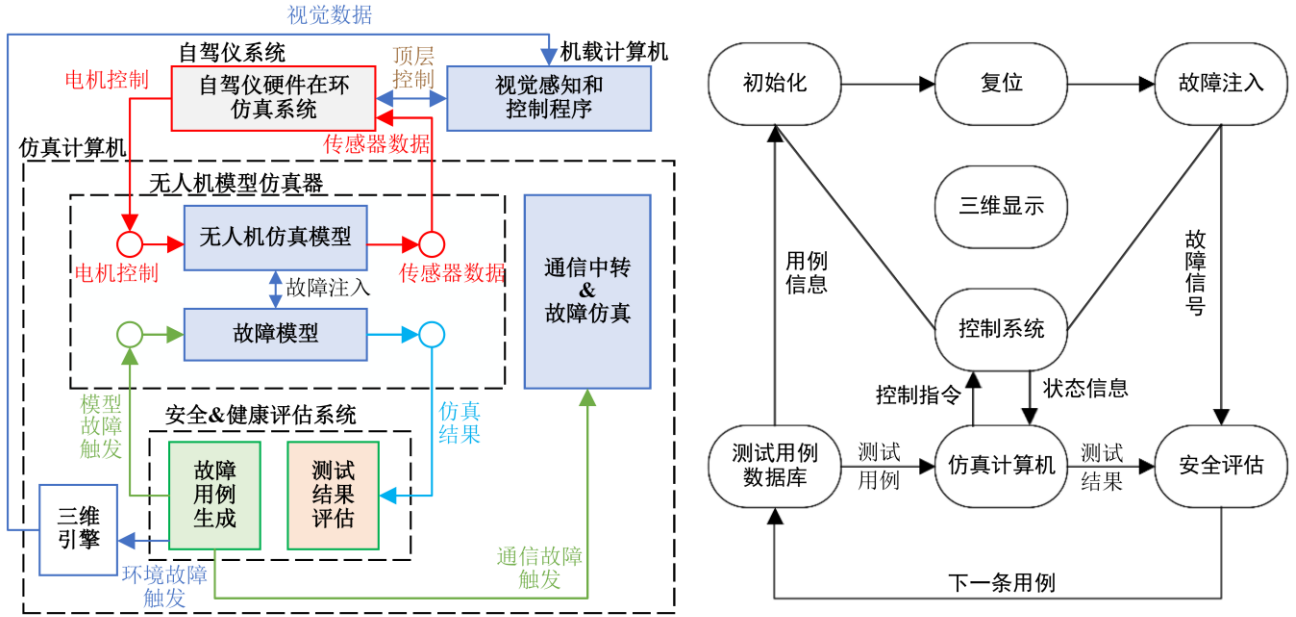
图：同一目标与力矩下的偏航角速度对照（教学简化模型，非平台实测）

复算t=1 s：A组0.1264，B组0.0787 rad/s；本模型无超调，不能推断原图峰值。

同一目标、力矩和阻尼下，惯量加倍使时间常数由1秒增至2秒。

2.3.4 对照测试与结果分析：故障测试

故障测试把异常条件加入同一任务



(a) 故障建模与注入原理图　　(b) 自动化故障仿真流程图

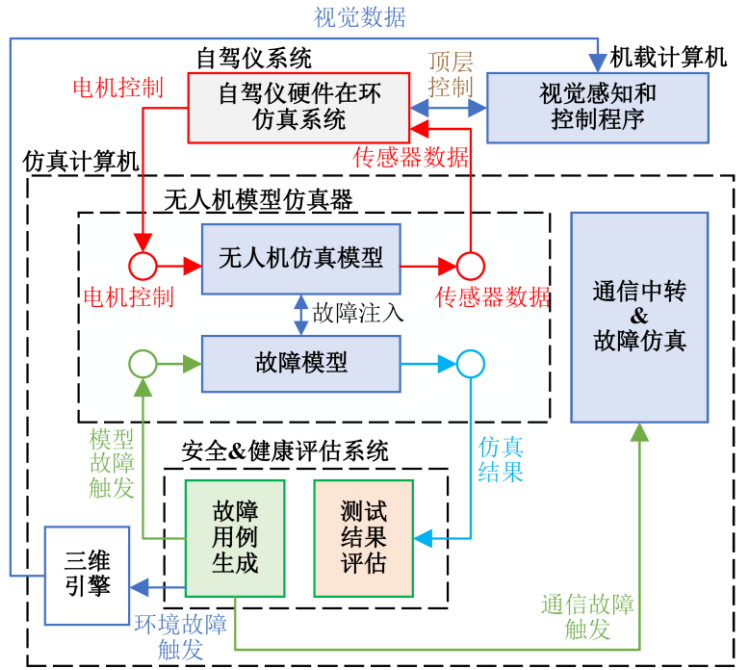
图2.19 故障注入系统设计框架图

- **测试目的：** 检验安全机制和容错策略，看发现故障要多久、能诊断哪些故障、处置是否正确。
- **用例条件：** 以无故障运行为基线，再对同一任务设置故障对象、时刻、幅值与持续时间。
- **注入与评估：** 故障可加在模型、三维环境、自驾仪或通信链路；结合误差、恢复时间等指标，评为可容忍、可恢复或不可恢复。

正常与故障工况采用相同任务，对照误差、检测延迟和恢复过程。

2.3.4 对照测试与结果分析：异常传播

故障注入位置决定首先变化的数据



(a) 故障建模与注入原理图

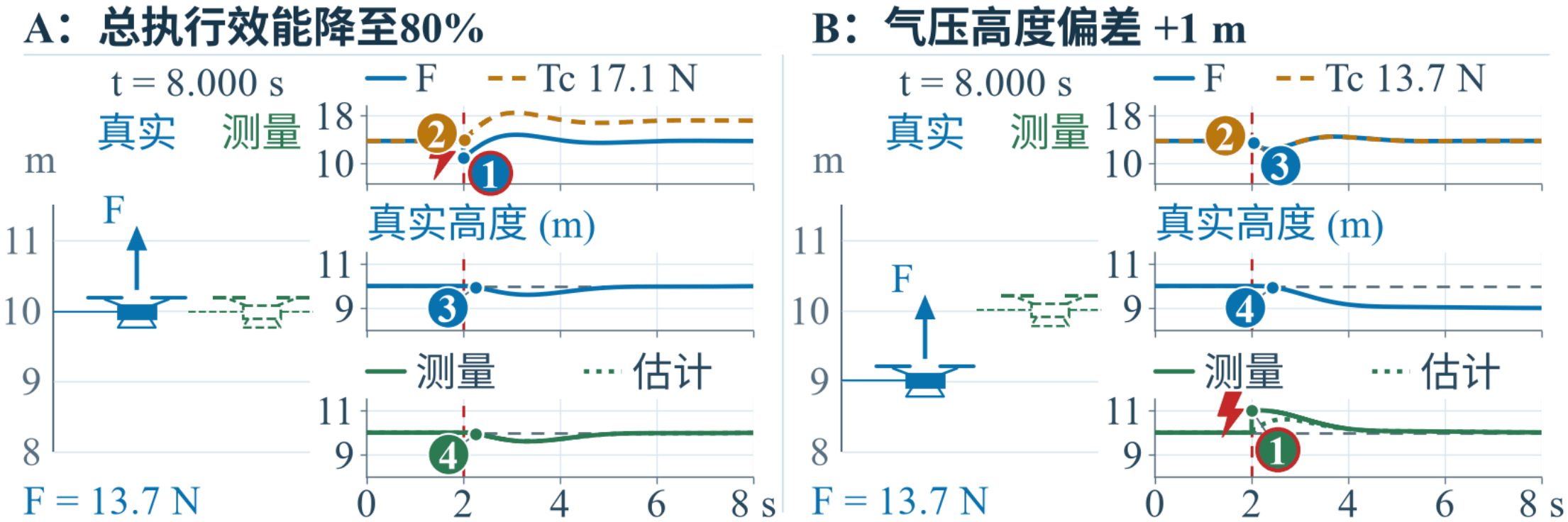
图2.19(a) 故障建模与注入原理图

- **动力异常：** 由“故障模型”注入，如电机效能下降先减小推力，随后运动、测量与控制输出变化。
- **观测异常：** 传感偏差先改变送往自驾仪的测量，再经状态估计和反馈控制影响实际运动。
- **通信异常：** 延迟或丢包由“通信中转&故障仿真”加入；日志记下注入时刻，便于分段对比。

动力故障先改变受力，观测故障先改变测量，后续响应沿闭环传播。

2.3.4 对照测试与结果分析：异常传播

先变的信号揭示故障从哪里进入闭环

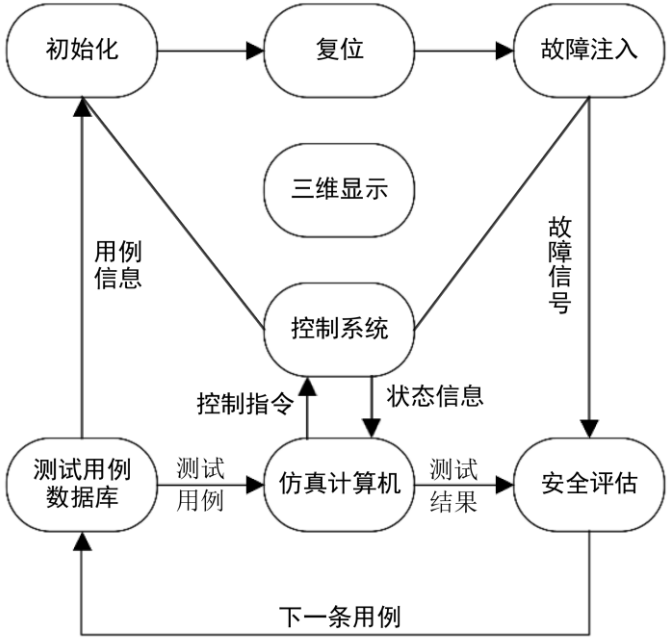


图：故障注入后的首变顺序（一维高度教学数值仿真）

执行效能下降先改变实际推力，气压偏置先改变测量高度。

2.3.4 对照测试与结果分析：自动测试

自动测试组织复位、执行、记录与判定



(b) 自动化故障仿真流程图

图2.19(b) 自动化故障仿真流程图

- **初始化与复位：**建立各软件通信并读入本轮用例，清除遗留故障，回到悬停等稳定状态。
- **注入与评估：**经UDP把故障参数送入仿真节点；结束后计算稳态误差、恢复时间和误差增长率。
- **记录与迭代：**配置、日志和结果按轮保存，再加载下一条用例；启动失败和安全终止单独统计。

每轮恢复约定初态并保存条件与日志，批量结果才具有可比性。

2.3.4 对照测试与结果分析：结果归因

沿目标、执行与响应解释结果

已观察到的现象	定位范围	能区分原因的记录
目标变化，执行输出未变化	目标接收、模式与控制职责	接收目标、当前模式、控制输出
执行变化，模型响应异常	通道、量纲与模型参数	执行量、实际参数、模型真值
模型已动，画面未对应	实例标识与三维状态映射	载具编号、发送位姿、显示对象
画面正常，算法没有新观测	传感配置与数据接收链	相机标识、帧时间与接收记录

从最先变化的量追踪后续响应，可以区分模型、控制和观测链的问题。

2.3 平台架构与开发流程：课堂练习

练习：为一项开发任务选择改动与验证

- **换载具：**将四旋翼改为固定翼，并保留同一任务航线。列出需改变的运动模型、执行器通道、控制分配与三维资源，说明各自作用。
- **改算法：**保持原有控制与模型，只替换视觉识别方法。说明算法输入输出，并设计同一场景下的识别精度和任务效果对照。
- **选验证：**研究控制板上一次计算需要多久、结果何时送出。选择有目标硬件参与的测试方式，说明应记录哪些时间与消息。

回答结构：改动对象 → 交付产物 → 运行环境 → 对照数据，并说明每一步的选择依据。

开发对象、运行产物和比较结果必须对应同一项任务。

2.3 平台架构与开发流程：练习要点

参考要点：改动对象、产物、环境与数据

- **换载具：**换成固定翼动力学模型，执行器通道改为电机加副翼、升降舵和方向舵，控制分配按固定翼构型，三维资源用固定翼外形；
只换外形时，物理响应仍是四旋翼。

- **改算法：**输入为相机图像，输出为目标位置或运动目标；离线用相同图像比较识别精度；闭环固定场景、初态、模型与控制器，观察轨迹和任务完成情况。

- **选验证：**用硬件在环让目标控制板参与运行，记录计算起止时刻、消息收发时间戳和处理器负载。

2.3 平台架构与开发流程：本节小结

架构定位对象，开发形成产物，测试解释效果

- **架构分工：**自驾仪计算估计与控制，模型计算运动，场景生成观测，上层程序给出任务目标；数据往返使这些计算形成闭环。
- **开发交付：**运动模型形成DLL，底层算法进入控制软件或固件，感知决策形成上层应用；每种产物都有明确的加载与运行位置。
- **验证选择：**模型、软件和硬件测试分别观察控制关系、程序协作与板上时序，实物测试进一步引入真实传感、执行和环境。
- **结果分析：**固定输入和初态，保存实际参数与日志，沿目标、执行量、运动和观测追踪变化，解释改动为何产生相应效果。

后续：2.4 API与SDK开发接口 → 2.5 例程体系与实验路线

沿模块输入输出追踪数据，才能把接口调用与实验现象联系起来。