

参考材料及实验任务说明

1、V-REP 与 MATLAB 联合仿真

V-REP 与 MATLAB 进行联合仿真时，V-REP 负责提供控制体对象模型、搭建仿真环境，以及解析来自 MATLAB 的 socket 控制命令，MATLAB 负责控制体对象控制算法的编写。由 MATLAB 输出 socket 控制命令，以控制 V-REP 中的控制体对象运动，从而完成任务。

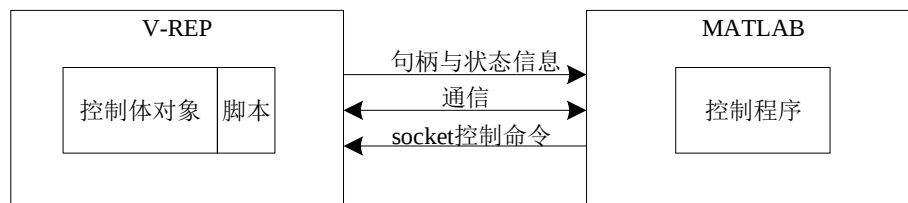


图 1.3-1 联合仿真时的通信控制流

仿真开始时，V-REP 通过 `simExtRemoteApiStart()` 函数，MATLAB 通过 `vrep=remApi('remoteApi')`、`vrep.simxStart()` 函数，共同建立控制体对象与控制程序之间的通信联系。

仿真过程中，控制程序通过 `simxGetObjectHandle()` 函数获取控制体对象各执行机构的句柄；通过 `simxGetObjectPosition()` 函数、`simxGetObjectOrientation()` 函数与 `simxGetObjectVelocity()` 函数分别获取各句柄在大地坐标系下的位置、方向与速度实时信息；经控制程序计算后，通过 socket 下发控制命令给 V-REP，V-REP 内部对 socket 控制命令进行解析后，驱动控制体对象实时运动。

仿真结束时，控制程序通过 `simxFinish()` 函数中断其与控制体对象的通信联系。

下面以驱动 **Manta 模型** 运动为例，说明使用 V-REP 与 MATLAB 进行联合仿真的方法。

1) V-REP 场景

在 V-REP 仿真界面内添加 Manta 模型，清空其脚本内所有的控制命令后，添写以下的控制命令（内含注释与说明）。

-----以下为脚本-----

```
if (simGetScriptExecutionCount()==0) then
    --获取 Manta 模型各执行机构的句柄
    vehiclebody_handle= simGetObjectHandle('body_dummy') --后轴中心的句柄
    steer_handle= simGetObjectHandle('steer_joint') --转向电机的句柄
    motor_handle= simGetObjectHandle('motor_joint') --驱动电机的句柄
    fl_brake_handle= simGetObjectHandle('fl_brake_joint') --左前轮制动电机的句柄
    fr_brake_handle= simGetObjectHandle('fr_brake_joint') --右前轮制动电机的句柄
    bl_brake_handle= simGetObjectHandle('bl_brake_joint') --左后轮制动电机的句柄
    br_brake_handle= simGetObjectHandle('br_brake_joint') --右后轮制动电机的句柄
    --变量初始化
    steer_angle=0 --初始化期望前轮偏角
    motor_velocity=0 --初始化期望电机转速
    brake_force=0 --初始化期望制动力
    --变量定义
    max_steer_angle=0.5235987 --最大期望前轮偏角 30°
    motor_torque=60 --电机最大转矩
    simServer=simExtRemoteApiStart(5050,20,false,false) --建立 V-REP 与 MATLAB 之间的通信联系
end
```

```

    print(" starting server..... ")
end
--接收 socket 信息并解包
data=simGetStringSignal("StringNamestr")
if (data) then --接收到新的 socket 信息
    ctrlvalue=simUnpackFloats(data)
    steer_angle = ctrlvalue[1] --期望前轮偏角控制量
    motor_velocity = ctrlvalue[2] --期望电机转速控制量
    print("steer_angle = ",steer_angle,"motor_velocity = ",motor_velocity);
    simClearStringSignal("StringNamestr") --清除本 socket 信息
else --未接收到新的 socket 信息
    steer_angle = 0 --期望前轮偏角控制量
    motor_velocity = 0 --期望电机转速控制量
    print("smessage no received ");
end
--电机转速与电机制动力的关系
if (math.abs(motor_velocity)< 0.1) then
    brake_force=100 --制动力
else
    brake_force=0
end
--设置期望前轮偏角的域值
if (steer_angle> max_steer_angle) then
    steer_angle=max_steer_angle
end
if (steer_angle< -max_steer_angle) then
    steer_angle= -max_steer_angle
end
--下发期望前轮偏角控制命令给 Manta 模型
simSetJointTargetPosition(steer_handle, steer_angle)
--下发期望电机转速与转矩给 Manta 模型
if(brake_force>0) then
    simSetJointForce(motor_handle, 0)
else
    simSetJointForce(motor_handle, motor_torque)
    simSetJointTargetVelocity(motor_handle, motor_velocity)
end
--下发制动力给 Manta 模型
simSetJointForce(fr_brake_handle, brake_force)
simSetJointForce(fl_brake_handle, brake_force)
simSetJointForce(bl_brake_handle, brake_force)
simSetJointForce(br_brake_handle, brake_force)

```

-----以上为脚本-----

2) MATLAB 程序

(1) MATLAB 配置方法

首先新建 MATLAB 项目，然后对 MATLAB 进行配置，以方便其与 V-REP 建立连接。配置方法如下：

将 V-REP 安装目录中 matlab 文件夹 (...\\V-REP3\\V-REP_PRO_EDU\\programming\\remoteApiBindings\\matlab) 下的 remoteApiProto.m、remApi.m、remoteApi.dll (Windows) (或 remoteApi.dylib (Mac) 或 remoteApi.so (Linux)) 三个文件添加到 MATLAB 新建项目中 (即将以上三个文件与自定义程序文件置于同一项目文件夹中)。

(2) MATLAB 程序编写

MATLAB 自定义程序如下：

-----以下为 MATLAB 自定义程序-----

```
close all
clear all
clc

%%建立本控制程序与 V-REP 中 Manta 模型之间的通信
disp('Program started');
vrep=remApi('remoteApi'); % using the prototype file (remoteApiProto.m)
vrep.simxFinish(-1); % just in case, close all opened connections
clientID=vrep.simxStart('127.0.0.1',5050,true,true,1000,20);%connectionAddress,connectionPort,
waitUntilConnected, doNotReconnectOnceDisconnected, timeOutInMs, commThreadCycleInMs
if (clientID>-1) %%如果本程序与 V-REP 仿真场景通信成功
    disp('Connected to remote API server');
else %%如果本程序与 V-REP 仿真场景通信失败
    disp('Failed connecting to remote API server');
end

%%变量初始化
deSpeed=0;%%初始化期望电机转速
deSteeringAngle=0;%%初始化期望前轮偏角

%%主控制程序
while(vrep.simxGetConnectionId(clientID) ~= -1)
    deSpeed=13; %%期望电机转速
    deSteeringAngle=0.1; %%期望前轮偏角
    %%通过 socket 把控制命令发送到 V-REP,控制命令在 V-REP 内再解包、执行
    CtrlValue=[deSteeringAngle,deSpeed];
    CtrlValueString=vrep.simxPackFloats(CtrlValue);
    [errorCode]=vrep.simxSetStringSignal(clientID,'StringNamestr',CtrlValueString,vrep.simx_o
pmode_oneshot);
end
fprintf('control loop ended\n'); %%控制回路结束
```

-----以上为 MATLAB 自定义程序-----

3) 联合仿真

首先在 V-REP 场景界面内启动仿真，然后在 MATLAB 内运行程序，即可驱动 Manta 模型进行运动。需要注意的是：V-REP 3.4 以前版本均是 32 位的，所包含的与 MATLAB 相连的库也是 32 位的，如果 MATLAB 是 64 位的可能会连不上，需要使用 64 位的库。当 MATLAB 与 VREP 连接上以后，在 MATLAB 中可以直接调用 VREP 中的 API 函数读取对象的位置、速度、航向等信息。

1-vrep 与 matlab 联合仿真示例 文件夹下的例程 在 vrep3.4+matlab2016 版，已经测试，运行正常。

2、Lua 介绍

Lua 的详细介绍可参考 <http://www.lua.org/pil/index.html>

3、实验任务

以 2-控制算法测试 文件夹中提供的 Manta 模型为控制对象，在 matlab 中编写纯跟踪控制程序(在提供的 code.m 文件中添加即可)，实现对阿克曼转向车辆的路径跟踪，并分析控制参数（如预瞄距离、初始速度）对控制效果的影响。

