**NATIONAL UNIVERSITY OF SINGAPORE**

**CEG5303: Intelligent Autonomous Robotic Systems**

**Lab: Guidance for Autonomous Vehicle**

Instructions: Simple Example

Simulation of Control Design for Assigned Systems

***1.1 Motion Control Design for Ground Vehicle***

In this part, the motion control problem is implemented for ground vehicle with model predictive control.

You are required to:

* Familiar with the model types of autonomous vehicles, establish any one of the vehicle models, i.e., basic 2-DOF, 3DOF considering yawing moment, etc. In addition, explain the model that you have established.
* Establish the simulation model/environment using MATLAB/Simulink, which should clearly show the plant model part, controller part and input/output channel.
* Realize the provided MPC control algorithm on motion planning for ground vehicle and choose the suitable control parameters.
* Take different autonomous vehicle structures into consideration in your dynamic modeling and optimize the control performance.
* Realize different driving functions or implement other kind of controllers of your choice.

***1.2 Required Software Platform***

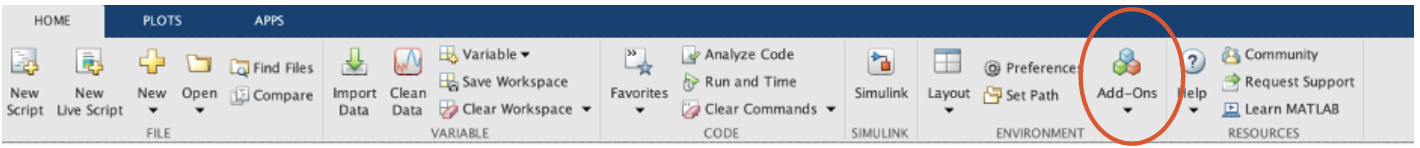
In this project, you need to use MATLAB and Simulink to realize software-in-the loop control. You need to familiar with basic language in MATLAB coding (M language), basic blocks, like gain, scope, etc. in Simulink modeling and some specific functional blockset, like Vehicle Dynamics Blockset. Then you can realize some control algorithms by simulation.

You can use **Vehicle Dynamics Blockset** in MATLAB/Simulink, which is only published in version from 2018a. The example model is established with version 2023b. So please install the suitable version on your computer. The detail of this blockset can be find in the Help Center, you can search to download the PDF documentation.

A screenshot of a web page

Description automatically generated

Click on Add-Ons in MATLAB as shown below:



Search for “Vehicle Dynamics Blockset” and install.

A white background with black text

Description automatically generated

We’ll need to use the optimization functions later, so also install “optimization\_toolbox” in the Add-Ons as well.

A screenshot of a white page

Description automatically generated

***1.3 Simulation Model Establishment***

The model of ***Simulation*** is established in **Simulink**. You can find the entrance in the highlight red square and create a blank model. Set parameters for this model.

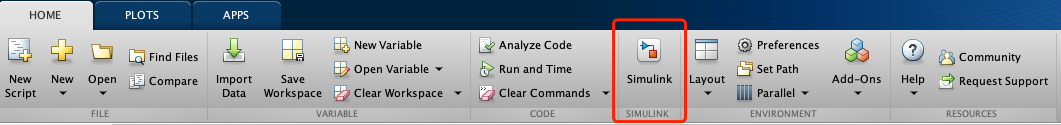


Fig1.2

Click Model Settings to set solver information.

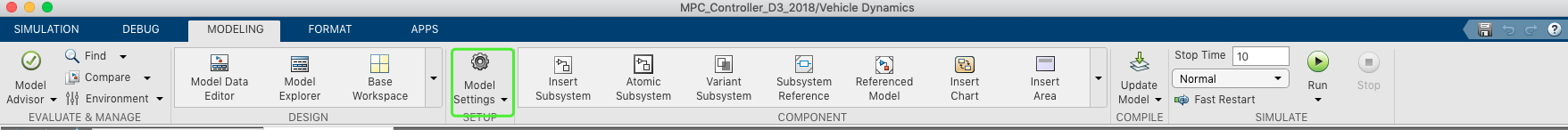


Fig1.3

**Solver selection:** Type: Fixed-step / Solver: auto/ Fixed-step size: 0.01

A screenshot of a computer

Description automatically generated

Fig 1.4

Then create the simulation! Here’s an example to realize ground vehicle control:

The diagram of ***Simulation*** model is shown as follow.

A diagram of a machine

Description automatically generated

Fig 1.5

The area of red square is the controller. Use Interpreted MATLAB Fcn block, which can be found in Simulink Library Browser on the top of the interface.

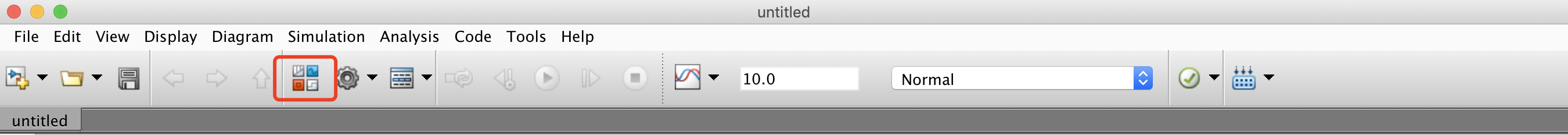


Fig.1.6

Double click this block and assign the name of .m document for the coding of algorithm. In this example, we will use “MPC\_controller.m”. Set Output dimensions to fit your algorithm.

A screenshot of a computer

Description automatically generated

Fig 1.7

The area of green square is the scope to show the results of simulation. Use Scope block to show the simulation curve. Use To workspace block to save variables to the MATLAB workspace.

The area of blue square is the subsystem of plant model. The inside of blue square is shown as follow.

A diagram of a machine

Description automatically generated

Fig 1.8

Use 3DOF Vehicle Body as the plant model from Vehicle Dynamics Blockset. All the blocks for Plant Model Establishment will be added from Simulink Library Browser.

If the signal fluctuates drastically, try to use **low pass filter** (as shown in the figure) and adjust the value of filter time constant and gain to obtain better performance.

A picture containing diagram

Description automatically generated

Fig 1.9

Double click on the “Vehicle Body 3DOF Single Track” to see the details of the vehicle setting. The plant model has block option and vehicle parameters. Use external longitudinal velocity and Front wheel steering to set the model as a basic model. You can also try to add other options to obtain a more complicated plant model.

A screenshot of a computer

Description automatically generated

Fig 1.10

Here is an example about the main parameters of vehicle.

Table: vehicle parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Symbol** | **Description** | **Value** | **Unit** |
|  | Vehicle mass | 1270 | kg |
|  | Front tire corner stiffness | 1.4324e+05 | [N/rad]: |
|  | Rear tire corner stiffness | 8.6517e+04 | [N/rad]: |
| ­ | Yaw polar inertia | 1536.7 | [kg\*m^2] |
|  | Longitudinal distance from center of mass to front axle | 1.015 | m |
|  | Longitudinal distance from center of mass to rear axle | 1.895 | m |

Some output signal of the plant model subsystem that will use in algorithm design is summarized as

Table: output signal

|  |  |  |
| --- | --- | --- |
| **Source-Symbol** | **Description** | **Unit** |
| Info- InertFrm-Disp- | Vehicle CG displacement along the earth-fixed X-axis | m |
| Info- InertFrm-Disp- | Vehicle CG displacement along the earth-fixed Y-axis | m |
| ­ | Vehicle CG velocity along the vehicle-fixed x-axis | m/s |
|  | Vehicle CG velocity along the vehicle-fixed y-axis | m/s |
|  | Rotation of the vehicle-fixed frame about the earth-fixed Z-axis (yaw) | rad |
|  | Vehicle angular velocity, r, about the vehicle-fixed *z*-axis (yaw rate) | rad/s |

Select these signals from the plant model.

After your build your Simulink model, save it as “MPC\_Controller.mdl”.

Set the parameters in “VehicleDynamicData.m” and run the file to load the parameters into your workspace.

Customized simulation model, set“v0\_ego”with your desired velocity like 20m/s, and decide front wheel angle signal with step input, sin input signal or other input signal with appropriate value (best below 0.15 rad).

***1.4 Control Model Establishment***

For motion planning, there are two main characteristic, kinematic and dynamic, should be considered in the model of controller.

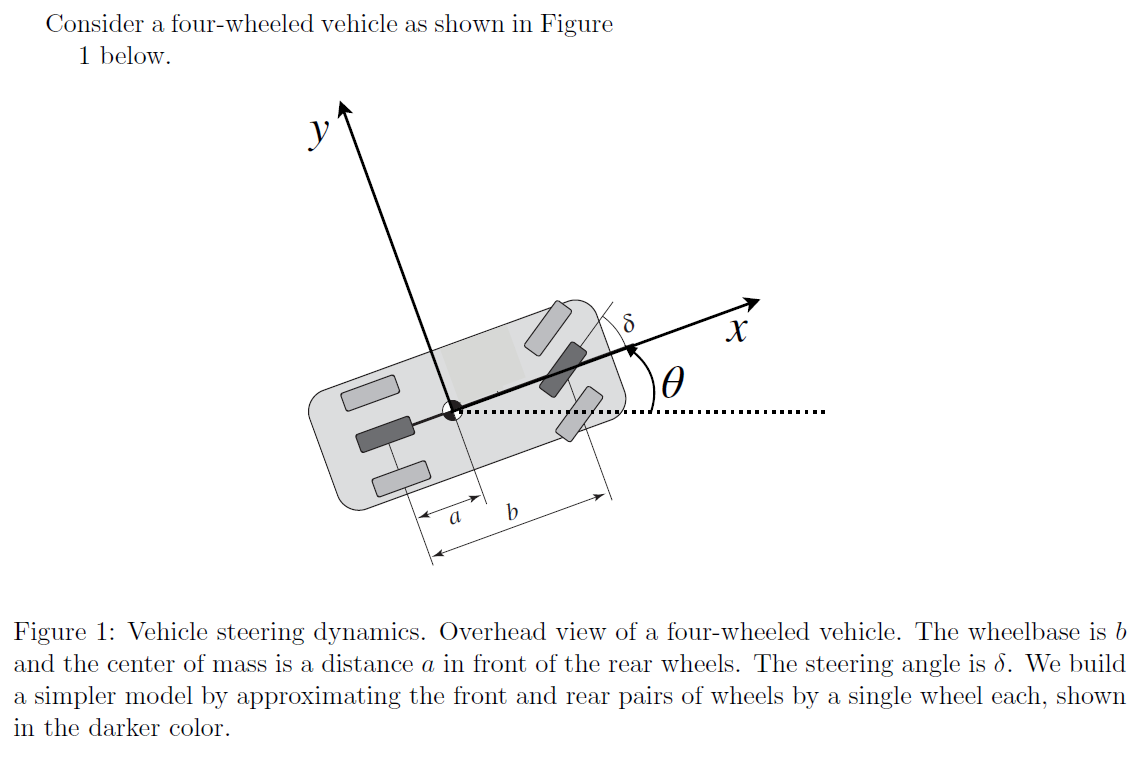


Fig 1.11

For the kinematic part, the equation can be expressed as



where is the coordinates of directions in the global coordinate system; is the longitudinal velocity, is the lateral velocity, is the heading angle in the global coordinate system; is the yaw rate.

For the dynamic part, the equation can be expressed as



where , , is the mass of the vehicle; is the yaw rate; is the moment of inertia of the vehicle about the -axis, and and are the distances from the center of gravity (CoG) to the front and rear axles, respectively; and are the lateral force with front and rear tires; and are the longitudinal force with front and rear tires,



is the steering-wheel angle; and are the cornering stiffness values of the front and rear tires, respectively.

For simplification, the longitudinal velocity can be assumed as a constant. This is a simple example about ground vehicle model. There are also many complicate models can be used to establish controller.

***1.5 Control Algorithm Design and Coding***

The controller in this example is designed with model predictive control. The state-space model used in this example can be written as

The controller in this example is designed with model predictive control. The state-space model used in this example can be written as



This state-space model is a nonlinear model. For model predictive control, you can linearize the nonlinear model or directly discrete the model to formulate an optimal problem. Here, we directly discrete the model to formulate an optimal problem. The discrete time step is . Then, the optimal control problem is to find the optimal control sequence such that



You can code with MATLAB to solve this optimal control problem with fmincon solver in Optimization Toolbox. Learn its function in help center of MATLAB.

fmincon is a nonlinear programming solver, which can find the minimum of a problem specified by

such that

You need to initialize control variables values, which can be set as 0. Set the option for optimization and the objective function and constrained function in separated functions, like ‘myobj1’, ‘mycon1’.

Open “MPC\_controller.m”, locate the TODO\_1, which is the constrained function. Implement the function. An example code is shown below:

function [c,ceq]=mycon1(u)

%%%%%%%%%%%TODO1%%%%%%%%%%%%

global N

u\_lim=0.2;

c=ones(2\*N,1); % for inequality constraints

for k=1:N

c(k)=u(k)-u\_lim;

c(k+N)=-u(k)-u\_lim;

end

ceq=zeros(4,1); % for equality constraints

%%%%%%%%%%TODO\_1\_end%%%%%%%%%

end

Locate TODO\_2, which is the objective function. Implement the function. An example code is shown below:

function f=myobj1(u)

%%%%%%%%%TODO\_2%%%%%%%%%%%%%%%%

global Ty M Izz lf lr dt xx vx N X

k1=0.2;k2=0.002;k3=0.2;

f=k1\*u(1)^2;

for i=1:N-1

f=f+k1\*u(i+1)^2+k3\*(u(i+1)-u(i))^2;

end

shape=10; dx1=50; dx2=4;

dy1=Ty;

Xs1=2.3\*vx;

vy0=xx(3,1); wr0=xx(4,1); fy\_0=xx(2,1); s\_y0=xx(1,1); X\_predict=X;

for i=1:N

X\_DOT=vx\*cos(fy\_0)-vy0\*sin(fy\_0);

X\_predict=X\_predict+X\_DOT\*dt;

z1=shape/dx1\*(X\_predict-Xs1)-shape/dx2;

Y\_ref=dy1/2\*(1+tanh(z1));

alpha1=-((vy0+lf\*wr0)/vx-u(i))\*180/pi;

alpha2=-(vy0-lr\*wr0)/vx\*180/pi;

Fy1=alpha1\*1250;

Fy2=alpha2\*755;

vy=vy0+(2\*Fy2/M+2\*Fy1/M-vx\*wr0)\*dt;

wr=wr0+(lf\*2\*Fy1/Izz-lr\*2\*Fy2/Izz)\*dt;

fy=fy\_0+wr0\*dt;

sy=s\_y0+vy0\*cos(fy\_0)\*dt+vx\*sin(fy\_0)\*dt;

vy0=vy; wr0=wr; fy\_0=fy; s\_y0=sy;

f=f+k2\*(sy-Y\_ref)^2;

end

%%%%%%%%%%TODO\_2\_end%%%%%%%%%%%%%

end

Locate TODO\_3, which computes the optimal control inputs using the “fmincon” function. An example is shown below:

%%%%%%%%%%%%TODO\_3%%%%%%%%%%%%%%%

u=ones(N,1)\*0; % control variable--delta\_f

options=optimset('Algorithm','interior-point','TolFun',1e-4,'LargeScale','on','MaxFunEvals',1e10, 'MaxIter',1e10);

[uu,fval,exitflag]=fmincon(@myobj1,u,[],[],[],[],[],[],@mycon1,options);

y(1)=uu(1); % use the first variable in the control horizon

%%%%%%%%%%%TODO\_3 end%%%%%%%%%%

**Finish the control in Matlab/Simulink using this example. Use and compare the performance with different values of control parameters like “***k1,k2,k3***”, fix values and reference trajectory. Explain your observation.**

***1.6 Sample Model Usage Instruction***

To run the simulation, first open the main model “MPC\_Controller.mdl”, then run parameter file “VehicleDynamicData.m”, and at last run the“MPC\_Controller.mdl” model.