Introduction to Model Predictive Control

Lectures on CT and DT Models

Francesco Borrelli

University of California at Berkeley,
Mechanical Engineering Department

ME290J-Spring11

Check List

You should know how to:

1. Concept of continuous time and discrete time system model
2. Concept of State Space (SS) model
3. Discretize nonlinear systems using Euler approach
4. Discretize linear systems in SS form using Euler or ZOH approaches
5. Discretize a linear system in TF form
6. Simulate a linear system in State Space form with Simulink (and/or Matlab)
7. Simulate a closed loop system with a CT model and a DT state feedback controller in Simulink (and/or Matlab)
8. Write a Matlab s-function

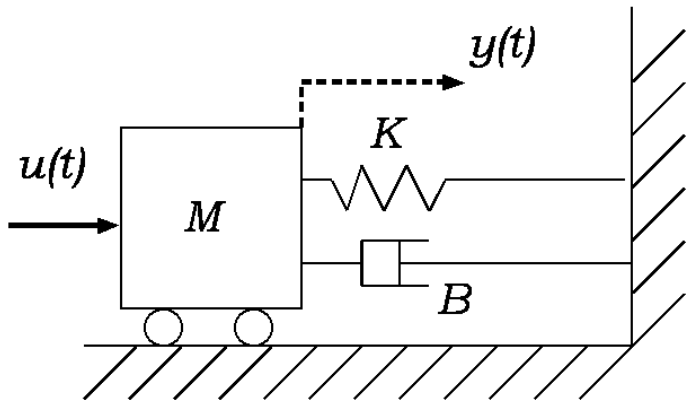
If so, solve Homework 1 on page 30 and Homework 2 on page 49.
Otherwise the following slides should help.

Continuous Time (CT) Systems

- Inputs and outputs of a CT system are defined for all time $t \in [0, +\infty)$, $t \in \mathbb{R}^+$
- Continuous linear dynamic systems are described by linear differential equations.

Example: The spring mass system

$$M \frac{d^2y(t)}{dt^2} + B \frac{dy(t)}{dt} + Ky(t) = u(t)$$



Discrete Time (DT) Systems

- Inputs and outputs of a DT system are defined at discrete time points. I.e. its inputs and outputs are sequences defined for $k = 0, 1, 2, \ldots,$ i.e. $k \in \mathbb{Z}^+$
- Discrete linear dynamic systems are described by linear difference equations.
  1. Bank savings account balance at the k-th month

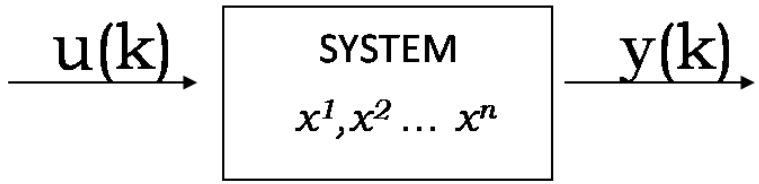
$$y(k+1) = (1+\alpha)y(k) + u(k)$$

  2. "Transformed" CT system

State Space Models of Discrete Time Systems

State space description of a system

Consider a discrete-time dynamic systems with an input $u(k)$ and an output $y(k)$.



- Given j and any $k > j$
- Given the input sequence $u(j), u(j+1), \ldots, u(k)$
- The state of the system at any instance j is the set of numbers $x^1(j), x^2(j), x^n(j)$ required to compute $y(k)$

Nonlinear DT State Space Description

$$\begin{aligned} x(k+1) &= f_d(x(k), u(k), k) \\ y(k) &= h_d(x(k), u(k), k) \end{aligned}$$

- Input vector $u(k) = [u^1(k),\ u^2(k), \cdots,\ u^m(k)]^T \in \mathcal{U} \subset \mathbb{R}^m$
- Output vector $y(k) = [y^1(k),\ y^2(k), \cdots,\ y^p(k)]^T \in \mathcal{Y} \subset \mathbb{R}^p$
- State vector $x(k) = [x^1(k),\ x^2(k), \cdots,\ x^n(k)]^T \mathcal{X} \subset \mathbb{R}^n$
- n is the order of the system
- State Equation: $x(k+1) = f_d(\ldots)$
- Output Equation: $y(t) = h_d(\ldots)$

DT LTI State Space Description

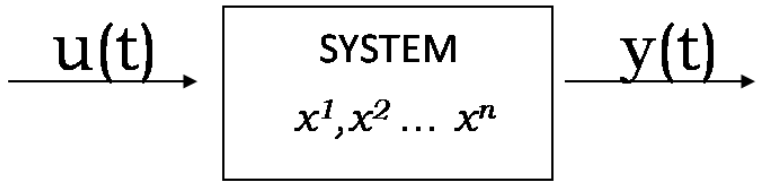
$$\begin{aligned} x(k+1) &= A_d\, x(k) + B_d\, u(k) \\ y(k) &= C_d\, x(k) + D_d\, u(k) \end{aligned}$$

- Input vector $u(k) = [u^1(k),\ u^2(k), \cdots,\ u^m(k)]^T \in \mathcal{U} \subset \mathbb{R}^m$
- Output vector $y(k) = [y^1(k),\ y^2(k), \cdots,\ y^p(k)]^T \in \mathcal{Y} \subset \mathbb{R}^p$
- State vector $x(k) = [x^1(k),\ x^2(k), \cdots,\ x^n(k)]^T \mathcal{X} \subset \mathbb{R}^n$
- n is the order of the system
- State Equation: $A_d \in \mathbb{R}^{n \times n}$, $B_d \in \mathbb{R}^{n \times m}$
- Output Equation: $C_d \in \mathbb{R}^{p \times n}$, $D_d \in \mathbb{R}^{p \times m}$

State Space Models of Continuous Time Systems

State Space Description of a CT System

Consider a continuous time dynamic systems with an input $u(t)$ and an output $y(t)$.



- Given t_0 and any $t > t_0$
- Given the input $u(\tau)$ for $t_0 \leq \tau \leq t$
- The state of the system at any instance t is the set of numbers $x^1(t_0), x^2(t_0), x^n(t_0)$ required to compute $y(t)$

Nonlinear CT State Space Description

$$\begin{aligned} \frac{d}{dt}x(t) &= f(x(t),u(t),t) \\ y(t) &= h(x(t),u(t),t) \end{aligned}$$

- Input vector $u(t) = [u^1(t),\ u^2(t), \cdots,\ u^m(t)]^T \in \mathcal{U} \subset \mathbb{R}^m$
- Output vector $y(t) = [y^1(t),\ y^2(t), \cdots,\ y^p(t)]^T \in \mathcal{Y} \subset \mathbb{R}^p$
- State vector $x(t) = [x^1(t),\ x^2(t), \cdots,\ x^n(t)]^T \mathcal{X} \subset \mathbb{R}^n$
- n is the order of the system
- State Equation: $\dot{x}(t) = f(\ldots)$
- Output Equation: $y(t) = h(\ldots)$

CT LTI State Space Description

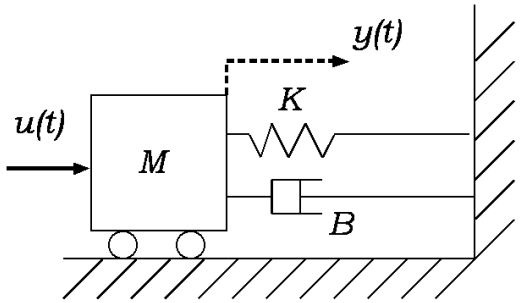
$$\begin{aligned} \frac{d}{dt}x(t) &= A\,x(t) + B\,u(t) \\ y(t) &= C\,x(t) + D\,u(t) \end{aligned}$$

- Input vector $u(t) = [u^1(t),\ u^2(t), \cdots,\ u^m(t)]^T \in \mathcal{U} \subset \mathbb{R}^m$
- Output vector $y(t) = [y^1(t),\ y^2(t), \cdots,\ y^p(t)]^T \in \mathcal{Y} \subset \mathbb{R}^p$
- State vector $x(t) = [x^1(t),\ x^2(t), \cdots,\ x^n(t)]^T \mathcal{X} \subset \mathbb{R}^n$
- n is the order of the system
- State Equation: $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$
- Output Equation: $C \in \mathbb{R}^{p \times n}$, $D \in \mathbb{R}^{p \times m}$

Example

The spring mass system

$$M\frac{d^2y(t)}{dt^2}+B\frac{dy(t)}{dt}+Ky(t)=u(t)$$



$$\frac{d}{dt}\underbrace{\begin{bmatrix} p(t) \\ v(t) \end{bmatrix}}_{x(t)} = \underbrace{\begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix}}_{A}\underbrace{\begin{bmatrix} p(t) \\ v(t) \end{bmatrix}}_{x(t)} + \underbrace{\begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}}_{B}u(t)$$

$$y(t) = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{C}\underbrace{\begin{bmatrix} p(t) \\ v(t) \end{bmatrix}}_{x(t)}$$

CT LTI Systems: Transfer Function Description

$$\begin{aligned} \frac{d}{dt}x(t) &= A\,x(t) + B\,u(t) \\ y(t) &= C\,x(t) + D\,u(t) \end{aligned}$$

- assume zero initial conditions, i.e. $x(0) = 0$
- Use the Laplace transforms

$$Y(s) = \mathcal{L}\{y(t)\} \quad U(s) = \mathcal{L}\{u(t)\}$$

- Then

LTI System in the Laplace Domain

$$Y(s) = G(s)\,U(s)$$

- $G(s)$ is called "Transfer Function"
- $G(s) = C(sI - A)^{-1}B + D$

From CT Systems to DT Systems

We Work With Discrete Time Models

We will use:

- Nonlinear Discrete Time

$$\begin{aligned} x(k+1) &= f_d(x(k), u(k), k) \\ y(k) &= h_d(x(k), u(k), k) \end{aligned}$$

- or LTI Discrete Time

$$\begin{aligned} x(k+1) &= A_d\, x(k) + B_d\, u(k) \\ y(k) &= C_d\, x(k) + D_d\, u(k) \end{aligned}$$

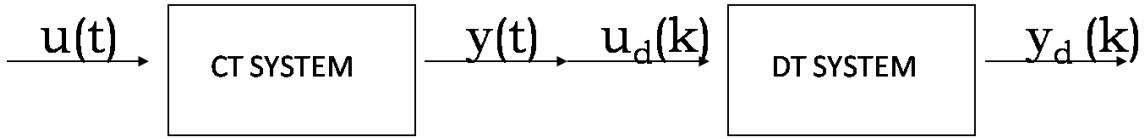
Discretization

We call discretization the procedure of obtaining an "equivalent" DT model from a CT one.

CT and DT in Simulink

1. Connecting DT and CT systems

2. Connecting CT and DT systems



3. Note that sampling time of many blocks can be changed directly

Euler Discretization of Nonlinear Models

1. Given CT model

$$\begin{aligned} \frac{d}{dt}x(t) &= f(x(t),u(t),t) \\ y(t) &= h(x(t),u(t),t) \end{aligned}$$

2. Approximate $\frac{d}{dt}x(t) \simeq \frac{x(t+\Delta T)-x(t)}{\Delta T}$
3. ΔT is the sampling time
4. With abuse of notation $x(k) \triangleq x(t)$, $x(k+1) \triangleq x(t+\Delta T)$
5. Then DT model is

$$\begin{aligned} x(k+1) &= x(k)+\Delta T f(x(k),u(k),k) = f_d(x(k),u(k),k) \\ y(k) &= h(x(k),u(k),k) = h_d(x(k),u(k),k) \end{aligned}$$

Under regularity assumptions, if ΔT is small and CT and DT have "same" initial conditions and inputs, then outputs of CT and DT systems "will be close"

Euler Discretization of Linear Models

1. Given CT model

$$\begin{aligned} \frac{d}{dt}x(t) &= A\,x(t) + B\,u(t) \\ y(t) &= C\,x(t) + D\,u(t) \end{aligned}$$

2. the DT model obtained with Euler discretization is

$$\begin{aligned} x(k+1) &= A_d\,x(k) + B_d\,u(k) \\ y(k) &= C_d\,x(k) + D_d\,u(k) \end{aligned}$$

with $A_d = I + \Delta T A,\ \ B_d = \Delta T B,\ \ C_d = C,\ \ D_d = D.$

- There are a variety of "better" discretization approaches (matlab: help c2d)
- will get rid of subscript d in the next lectures

ZOH ('Perfect') Discretization of Linear Models

1. Given CT model

$$\begin{aligned} \frac{d}{dt}x(t) &= A\,x(t) + B\,u(t) \\ y(t) &= C\,x(t) + D\,u(t) \end{aligned}$$

2. the DT model obtained with ZOH discretization is

$$\begin{aligned} x(k+1) &= A_d\,x(k) + B_d\,u(k) \\ y(k) &= C_d\,x(k) + D_d\,u(k) \end{aligned}$$

with $A_d = e^{A\Delta T}$, $B_d = \int_0^{\Delta T} e^{A}\eta B\ d\eta$, $C_d = C$, $D_d = D$
ΔT is the sampling time.

ZOH ('Perfect') Discretization of Linear Models

1. the DT model obtained with ZOH discretization is

$$\begin{aligned} x(k+1) &= A_d\, x(k) + B_d\, u(k) \\ y(k) &= C_d\, x(k) + D_d\, u(k) \end{aligned}$$

with $A_d = e^{A\Delta T}$, $B_d = \int_0^{\Delta T} e^{A\eta} B \; d\eta$, $C_d = C$, $D_d = D$

Main property:

- Simulate the CT model starting from x_0 and applying $\hat{u}(t) = ZOH(u(t), \Delta T)$:

$$\hat{u}(t) = u(k\Delta T) \quad \text{if} \quad k\Delta T \leq t < (k+1)\Delta T$$

- Simulate the DT model starting from x_0 and applying $u(0),\ u(\Delta T),\ u(2\Delta T), \ldots$
- Then

$$y(k) = y(k\Delta T)$$

where $y(k)$ is the output of the DT model at the k-th discrete instant and $y(k\Delta T)$ is the output of the CT model at time $k\Delta T$

From CT Systems in Transfer Function Form to CT Systems in State Space Form

Discretization of Linear Models: Transfer Functions

1. Consider the system described by the transfer Function

$$Y(s) = G(s)\,U(s)$$

2. Obtain state space equivalent model (matlab command tf2ss)

$$\begin{aligned} \tfrac{d}{dt}x(t) &= A\,x(t) + B\,u(t) \\ y(t) &= C\,x(t) + D\,u(t) \end{aligned}$$

3. Apply discretization to state space model in 2

Note: There are infinite sets of $\begin{bmatrix} A & B \\ C & D \end{bmatrix}$ that produce the same $G(s)$

From TF to SS

Example

$$G(s) = \frac{B(s)}{A(s)} = \frac{b_2 s^2 + b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0}$$

Result:

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 \end{bmatrix}}_{A} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}}_{B} u$$

$$y = \underbrace{\begin{bmatrix} b_o & b_1 & b_2 \end{bmatrix}}_{C} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

From TF to SS

Example

$$G(s) = \frac{B(s)}{A(s)} = \frac{b_2 s^2 + b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0}$$

1. Define $X(s) = \frac{1}{A(s)} U(s)$
2. So that $Y(s) = B(s)\, X(s)$
3. We obtain

$$\begin{aligned} A(s)X(s) &= U(s) \\ (s^3 + a_2 s^2 + a_1 s + a_0)X(s) &= U(s) \end{aligned}$$

4. Recall

$$\mathcal{L}\left\{ \frac{d^n}{dt^n} x(t) \right\} = s^n\, X(s)$$

5. Then

$$\frac{d^3\, x(t)}{dt^3} + a_2\, \frac{d^2\, x(t)}{dt^2} + a_1\, \frac{d\, x(t)}{dt} + a_o\, x(t) = u(t)$$

From TF to SS

Example

$$\frac{d^3\,x(t)}{dt^3} + a_2\,\frac{d^2\,x(t)}{dt^2} + a_1\,\frac{d\,x(t)}{dt} + a_o\,x(t) = u(t)$$

1. Define state Variables

$$x^1(t) = x(t),\ x^2(t) = \frac{d\,x(t)}{dt},\ x^3(t) = \frac{d^2\,x(t)}{dt^2}$$

2. Notice that $\frac{d^3\,x(t)}{dt^3} = \frac{d\,x^3(t)}{dt}$
3. Then

$$\begin{aligned}
\frac{d\,x^1(t)}{dt} &= x^2(t) \\
\frac{d\,x^2(t)}{dt} &= x^3(t) \\
\frac{d\,x^3(t)}{dt} &= -a_0\,x^1(t) - a_1\,x^2(t) - a_2\,x^3(t) + u(t)
\end{aligned}$$

From TF to SS

Example

$$X(s) = \frac{1}{A(s)}\,U(s), \quad Y(s) = B(s)\,X(s)$$

1. $Y(s) = (b_0 + b_1 s + b_2 s^2) X(s)$
2. $y(t) = b_o\,x(t) + b_1\,\frac{d\,x(t)}{dt} + b_2\,\frac{d^2\,x(t)}{dt^2}$
3. Therefore:
$$y(t) = b_o\,x^1(t) + b_1\,x^2(t) + b_2\,x^3(t)$$

From TF to SS

Example

$$G(s) = \frac{B(s)}{A(s)} = \frac{b_2 s^2 + b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0}$$

1. Is Equivalent to

$$\begin{aligned}
\frac{d\,x^1(t)}{dt} &= x^2(t) \\
\frac{d\,x^2(t)}{dt} &= x^3(t) \\
\frac{d\,x^3(t)}{dt} &= -a_0\,x^1(t) - a_1\,x^2(t) - a_2\,x^3(t) + u(t) \\
\\
y(t) &= b_o\,x^1(t) + b_1\,x^2(t) + b_2\,x^3(t)
\end{aligned}$$

From TF to SS

Example

$$G(s) = \frac{B(s)}{A(s)} = \frac{b_2 s^2 + b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0}$$

1. Is Equivalent to

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 \end{bmatrix}}_{A} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}}_{B} u$$

$$y = \underbrace{\begin{bmatrix} b_o & b_1 & b_2 \end{bmatrix}}_{C} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Homework 1

1. Download the "Ball and Plate" system description from bSpace. In the rest focus only on the x-axis and assume we measure all four states: $y_\beta(t) = x_\beta(t)$. Where sampling time is required use $\Delta T = 30ms$. (Note: we denoted the state vector x_β and the output $y_\beta(t)$ to distinguish them from the x and y positions of the ball).
2. Compute the transfer function of the system by using Matlab (hint: use the "ss2tf" command).
3. Compute a DT system by using Euler discretization, Call this system: DT system 1.
4. Compute a DT system the matlab command "c2d" with the method 'zoh'. Call this system: DT system 2.

Homework 1

1. Use Simulink to simulate the CT LTI State Space system starting from zero initial condition and with the following input profile: $u(t) = 0$ if $0 \leq t < 5$, $u(t) = 0.2V$ if $5 \leq t < 5.5$ and $u(t) = 0$ if $t \geq 5.5$.
2. Use Simulink to simulate the DT system 1 starting from zero initial conditions and with input obtained by sampling the continuous time input $u(t)$ of the previous point.
3. Use Simulink to simulate the DT system 2 starting from zero initial condition and with same input of the previous point.
4. Plot and compare the results of points 1,2,3

Homework 1

1. The control goal is to bring the ball from any admissible position to the origin of the state space (position and speed of the plate equal to zero and position and speed of the ball equal to zero).
   The state feedback discrete time controller $u(k) = Kx(k)$ with $K = [21.1 \quad 4.0 \quad -266.3 \quad -5.8]$ and sampling $\Delta T = 30ms$ should do the job.
2. Simulate for 3 seconds the closed loop system (DT controller and CT LTI system in state space form) when the system starts from the initial condition $x(0) = [20, 0, 0, 0]$ (plate and ball at rest and ball at distance 20cm from the origin)
   1. Does the controller bring the ball to the origin?
   2. Are state and input constraints satisfied?
   3. How would you modify the controller in order to satisfy the constraints?

S-Functions

M-File S-Functions

MATLAB function of the following form:
[sys,x_0,str,ts]=f(t,x,u,flag,p1,p2,...)

where

- f is the S-function's name
- t is the current time
- x is the state vector of the corresponding S-function block
- u is the block's inputs
- flag indicates a task to be performed
- p1, p2, ... are the block's parameters.

During simulation of a model, Simulink repeatedly invokes **f**, using flag to indicate the task to be performed for a particular invocation.

M-File S-Functions

subfunctions, called S-function callback methods, perform the tasks required of the S-function during simulation.

Simulation Stage	S-Function Routine	Flag
Initialization	`mdlInitializeSizes`	`flag = 0`
Calculation of next sample hit (variable sample time block only)	`mdlGetTimeOfNextVarHit`	`flag = 4`
Calculation of outputs	`mdlOutputs`	`flag = 3`
Update of discrete states	`mdlUpdate`	`flag = 2`
Calculation of derivatives	`mdlDerivatives`	`flag = 1`
End of simulation tasks	`mdlTerminate`	`flag = 9`

M-File S-Functions

Consider the system with m inputs u, q outputs (denoted by y)

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \vdots \\ \dot{x}_n(t) \\ \hline y_1(t) \\ y_2(t) \\ \vdots \\ y_q(t) \end{bmatrix} = \begin{bmatrix} f_1\left(x_1(t), x_2(t), \ldots, x_n(t), u_1(t), u_2(t), \ldots, u_m(t), t\right) \\ f_2\left(x_1(t), x_2(t), \ldots, x_n(t), u_1(t), u_2(t), \ldots, u_m(t), t\right) \\ \vdots \\ f_n\left(x_1(t), x_2(t), \ldots, x_n(t), u_1(t), u_2(t), \ldots, u_m(t), t\right) \\ \hline h_1\left(x_1(t), x_2(t), \ldots, x_n(t), u_1(t), u_2(t), \ldots, u_m(t), t\right) \\ h_2\left(x_1(t), x_2(t), \ldots, x_n(t), u_1(t), u_2(t), \ldots, u_m(t), t\right) \\ \vdots \\ h_q\left(x_1(t), x_2(t), \ldots, x_n(t), u_1(t), u_2(t), \ldots, u_m(t), t\right) \end{bmatrix}$$

When flag 1 (mldDerivatives)

function sys=mdlDerivatives(t,x,u)

$sys(1) = f_1(\ldots)$, ...,$sys(n) = f_n(\ldots)$

When flag 3 (mldOutputs)

function sys=mdlOutputs(t,x,u)

$sys(1) = h_1(\ldots)$, ...,$sys(n) = h_q(\ldots)$

M-File S-Functions

Hints:

- in bSpace: a template implementation of an M-file S-function, sfuntmpl.m, (also in matlabroot/toolbox/simulink/blocks).
- in bSpace: a simple example for continuous time systems
- in bSpace: a simple example for discrete time systems
- In Matlab help search for "S-functions"

Nonlinear Vehicle Model for Fuel Consumption

Simplified vehicle dynamical model

Recall requirements from first lecture

- Speed needs to be a state. It will be constrained over the horizon.
- Input is "Engine Torque"
- Output is "Fuel"

Main issue

- Traffic speed sensors at fixed distance

Need a distance-based model

Simplified vehicle model

Use $s(k)$ as the independent variable, representing the k-th position, and a fixed step size of $\Delta s = s(k+1) - s(k)$ for all k.

Distance Based Model

$$\begin{aligned}\tfrac{1}{2}mv^2(k+1) - \tfrac{1}{2}mv^2(k) = \quad & F_w(k)\Delta s - \\ & -\tfrac{1}{2}\rho C_a v^2(k)\Delta s - \mu mg\Delta s - mg\theta(k)\Delta s\end{aligned}$$

where

- $v(k)$: the velocity of the vehicle at position $s(k)$
- $F_w(k)$: the wheel force at position $s(k)$
- ρC_a: the coefficient of drag multiplied by the car frontal area
- m: mass of the vehicle
- g: gravitational constant
- μ: friction coefficient associated with the rolling resistance
- $\theta(k)$: road grade at position $s(k)$

Simplified vehicle model

The "real" input is the engine torque $T(k)$:

$$F_w(k) = \frac{T(k)\eta R(k)}{r} \tag{1}$$

i.e.

$$T(k) = \frac{F_w(k)r}{\eta R(k)} \tag{2}$$

where r is the rolling radius of the tire, R is the (gear) ratio of wheel to engine speed and η is the driveline efficiency.

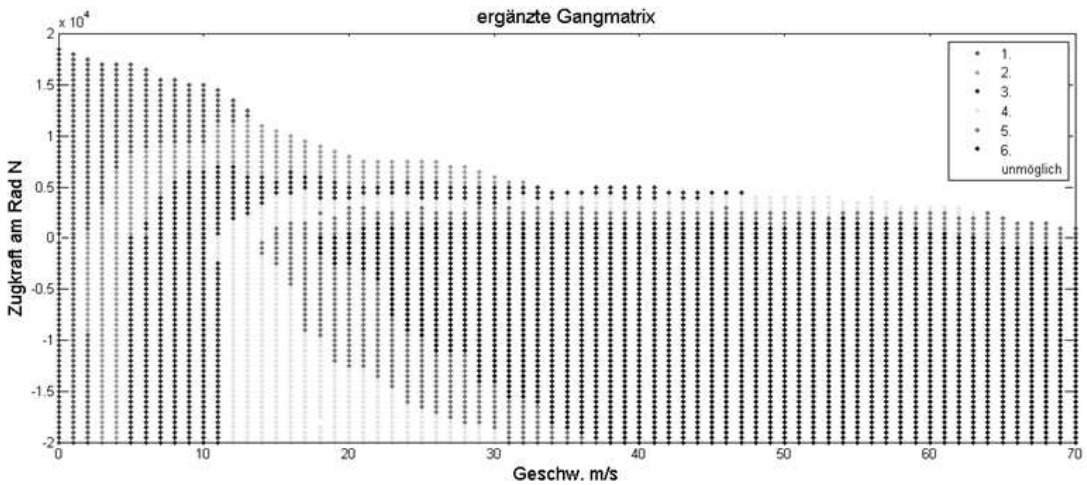
Simplified vehicle model

The gear ratio is given as a static nonlinear map of F_w and engine speed w_e:

$$R(k) = R(k, F_w(k), v(k))$$

so that

$$T(k) = \frac{F_w(k) r}{\eta R(k, F_w(k), v(k))}$$



Simplified vehicle model

Approach:

- The input for the model used in the control will be $F_w(k)$
- The engine torque will be an auxiliary variable function of state and input

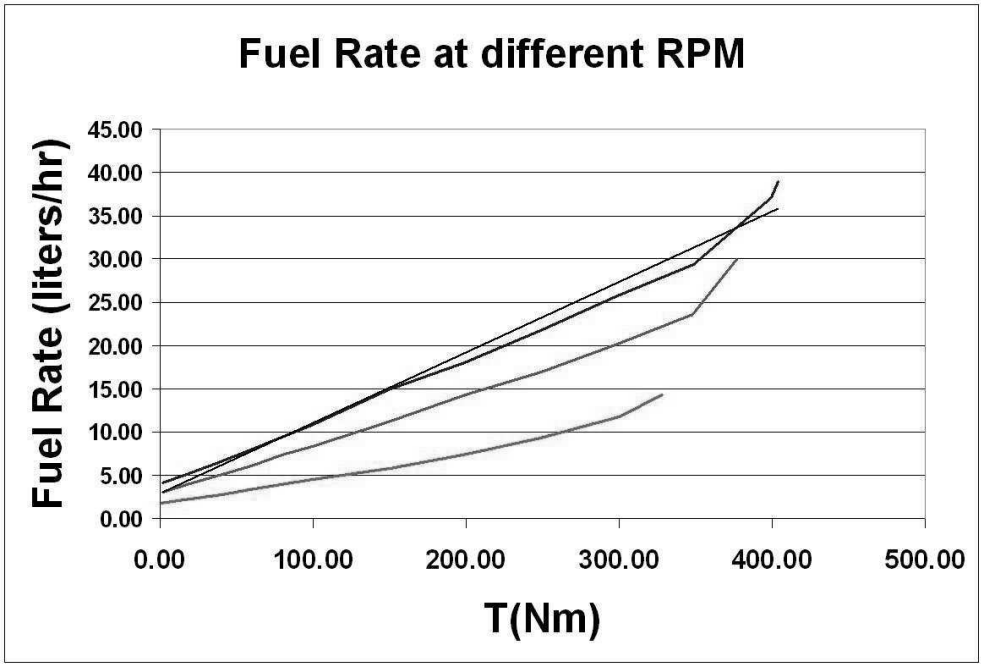
$$T(k) = \frac{F_w(k) r}{\eta R(k, F_w(k), v(k))}$$

$T(k)$ needed for

- Actual input to the vehicle
- Fuel model

Simplified fuel model

Denote the fuel rate as $\dot{f}$



Fist observation:

$$\dot{f} \simeq \alpha(\omega_e) + \beta(\omega_e)T(k)$$

where ω_e is the engine speed.

Simplified fuel model

Good empirical model:

$$\dot{f} \simeq c\omega_e^2 + d\omega_e T(k) \tag{3}$$

Note that

- at constant speed $w_e(k) = v(k)R/r$
- The fuel $f(k)$ used from position $s(k)$ till position $s(k+1)$ is $\dot{f}$ multiplied the time to complete the Δs segment: $(\Delta s/v(k))$.

In conclusion

$$\begin{aligned} f(k) &= c\left(\frac{R(k,F_w(k),v(k)}{r}\right)^2 \Delta s\ v(k) + d\left(\frac{R(k,F_w(k),v(k)}{r}\right)\Delta s\ T(k) \\ &= h(v(k), T(k), F_w(k)) \end{aligned} \tag{4}$$

Simplified fuel model

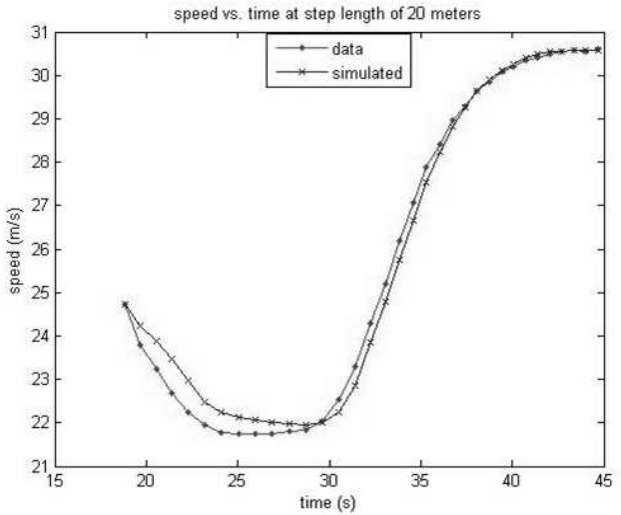
The resulting model is

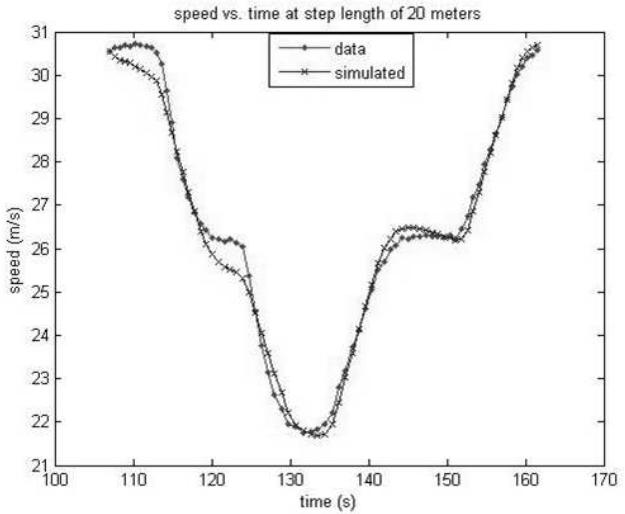
Distance Based Model

$$
\begin{aligned}
x(k+1) &= ax(k) + bu(k) - b_2\theta(k) - \gamma \\
T(k) &= g(x(k), u(k)) \\
y(k) &= h(\sqrt{x(k)}, T(k), u(k))
\end{aligned}
$$

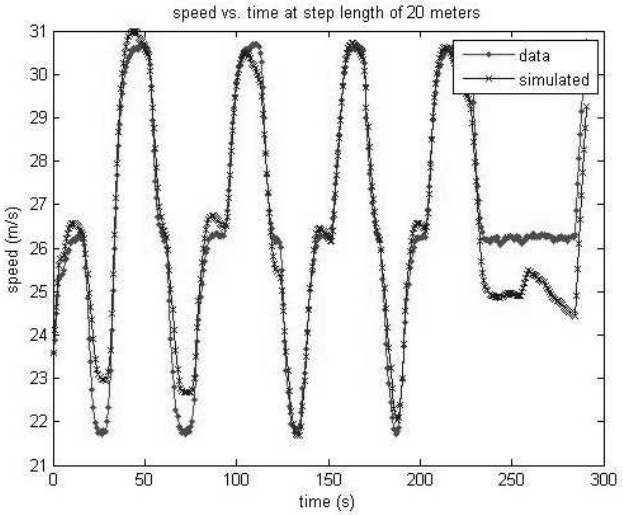
- distance based model
- $x(k) = v(k)^2$ is the speed squared, $u(k) = T_w(k)$ is the wheel torque, $y(k)$ is fuel used from $s(k)$ till $s(k+1)$
- nonlinear model, not in standard form.

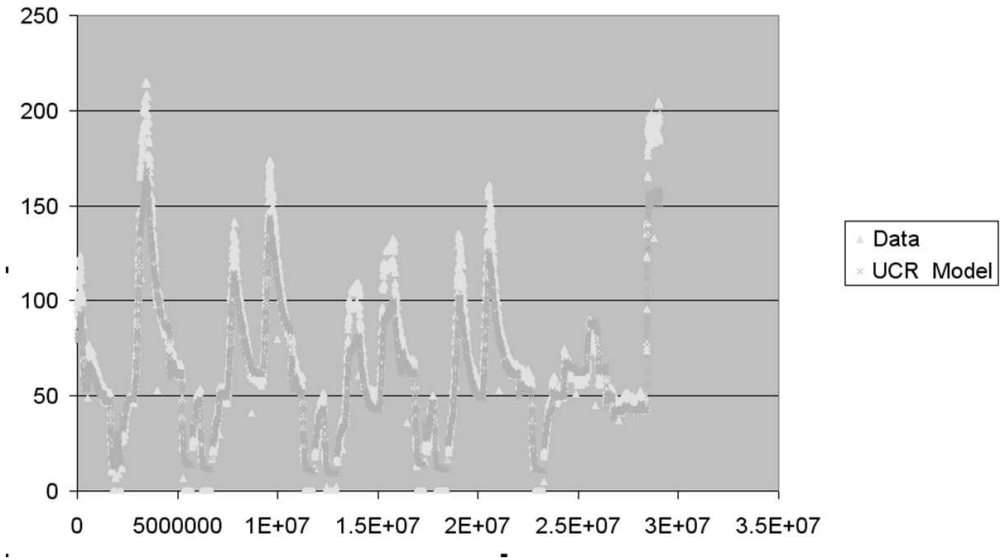
Is it a Good Model? - Identification





Is it a Good Model? - Validation





Homework 2

Consider the vehicle model (for fuel consumption), assume zero grade, do exercise in Simulink.

- Use all data provided on bSpace and write an S-Function for simulating the system. Use as input the wheel force $F_w(k)$ and as outputs the vehicle speed and the fuel consumption
- Design a very simple controller (P, PI,..) which regulates the wheel force $F_w(k)$ to follow a desired speed profile.
- Take the average speed profile provided on b-Space (101 highway) as a reference (position,speed) (be careful that the signal is sampled every 10 meters while the suggested Δs for the model is 40 meters).
- Show with a plot that the designed controller allows the vehicle to track the given reference.
- Provide the total fuel consumption for the trip.
- Provide the total time to complete the trip.

Homework 2

Clarifications

- The gear ratio R(k,Fw(k),v(k)) has to be computed as follows:
  - Compute the gear buy using the table Gear Matrix provided in Matlab
  - Then use the following ratios: R= 4.17*3.32 if gear number is 1, R= 2.34*3.32 if gear number is 2, R= 1.52*3.32 if gear number is 3, R= 1.14*3.32 if gear number is 4, R= 0.87*3.32 if gear number is 5, R= 0.69*3.32 if gear number is 6.
- When the fuel ratio is computed by using equation (3) AND using the "c" and "d" values provided in the matlab file on bSpace, THEN the fuel ratio is in liters per seconds.
- The fuel consumption needs to be computed by using the "c" and "d" values provided in the matlab file AND using the equation (4), with 1- the gear ratio described above, 2- distance in meters, time in seconds , torque in Nm The final value is the fuel consumption (in liters) during the "Delta s" traveled distance (in meters).

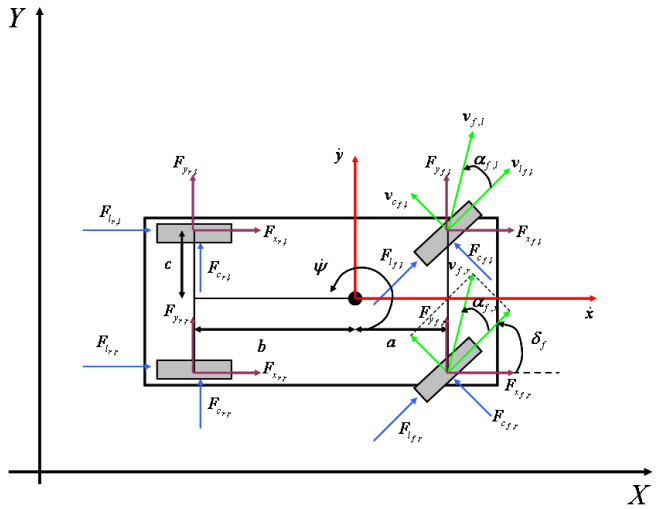
Homework 2

Consider the Ball and Plate model

- Consider the last part of Homework 1:
  - The control goal is to bring the ball from any admissible position to the origin of the state space (position and speed of the plate equal to zero and position and speed of the ball equal to zero). The state feedback discrete time controller $u(k) = Kx(k)$ with $K = [21.1 \quad 4.0 \quad -266.3 \quad -5.8]$ and sampling $\Delta T = 30ms$ should do the job.
  - Simulate for 3 seconds the closed loop system (DT controller and CT LTI system in state space form) when the system starts from the initial condition $x(0) = [20, 0, 0, 0]$ (plate and ball at rest and ball at distance 20cm from the origin)
- Redo the exercise by substituting the state-space simulation block with an S-Function.

Additional Material: 4-Wheels Model

Simplified vehicle dynamical model



$$m\ddot{y} = -m\dot{x}\dot{\psi} + F_{y_{f,l}} + F_{y_{f,r}} + F_{y_{r,l}} + F_{y_{r,r}}, \tag{5a}$$

$$m\ddot{x} = m\dot{y}\dot{\psi} + F_{x_{f,l}} + F_{x_{f,r}} + F_{x_{r,l}} + F_{x_{r,r}} \tag{5b}$$

$$\begin{aligned} I\ddot{\psi} = {} & a\left(F_{y_{f,l}} + F_{y_{f,r}}\right) - b\left(F_{y_{r,l}} + F_{y_{r,r}}\right) \\ & + c\left(-F_{x_{f,l}} + F_{x_{f,r}} - F_{x_{r,l}} + F_{x_{r,r}}\right), \end{aligned} \tag{5c}$$

$$\dot{Y} = \dot{x}\sin\psi + \dot{y}\cos\psi, \tag{5d}$$

$$\dot{X} = \dot{x}\cos\psi - \dot{y}\sin\psi. \tag{5e}$$

Simplified vehicle dynamical model

- $\dot{x}$ and $\dot{y}$: velocities along the longitudinal and lateral vehicle axes
- ψ: yaw angle, $\dot{\psi}$: yaw rate
- Y and X are the lateral and longitudinal positions of the vehicle Center of Gravity (CoG) in the inertial frame
- m is the mass of the car, I is the inertia of the vehicle along the vertical axis
- Constants a and b are the distances from the CoG of the front and rear axles, respectively, c is the distance of the left and right wheels from the longitudinal vehicle axis.

Simplified vehicle dynamical model

The lateral and longitudinal tire forces $F_{c_{\star,\bullet}}$ and $F_{l_{\star,\bullet}}$ generate $F_{y_{\star,\bullet}}$ and $F_{x_{\star,\bullet}}$, along the lateral and longitudinal vehicle axes:

$$F_{y_{\star,\bullet}} = F_{l_{\star,\bullet}} \sin\delta_\star + F_{c_{\star,\bullet}} \cos\delta_\star, \tag{6a}$$

$$F_{x_{\star,\bullet}} = F_{l_{\star,\bullet}} \cos\delta_\star - F_{c_{\star,\bullet}} \sin\delta_\star. \tag{6b}$$

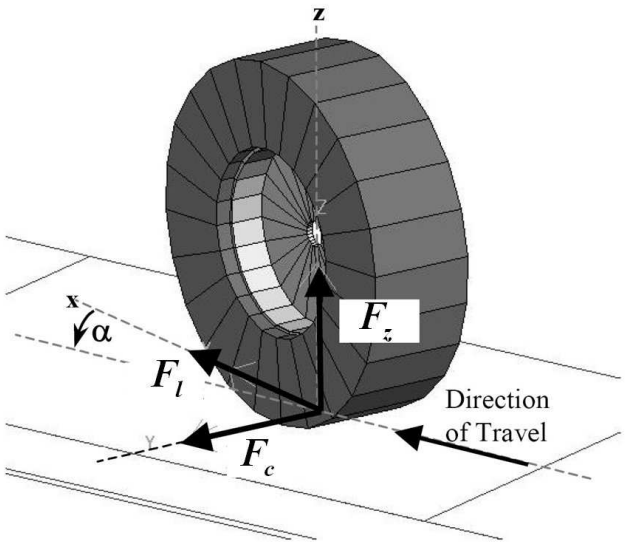


Figure: Illustration of tire model nomenclature

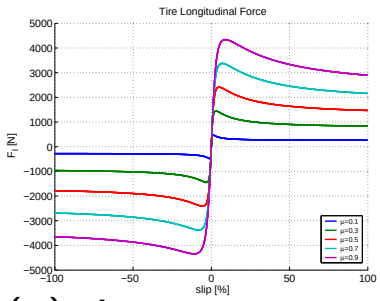
Tyre Model

The lateral and longitudinal tire forces $F_{c_{\star,\bullet}}$ and $F_{l_{\star,\bullet}}$ are directed as in previous Figure and computed as

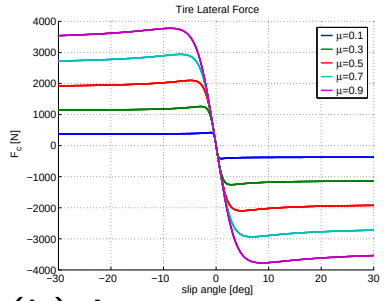
$$F_{c_{\star,\bullet}} = f_c(\alpha_{\star,\bullet}, s_{\star,\bullet}, \mu_{\star,\bullet}, F_{z_{\star,\bullet}}), \tag{7a}$$
$$F_{l_{\star,\bullet}} = f_l(\alpha_{\star,\bullet}, s_{\star,\bullet}, \mu_{\star,\bullet}, F_{z_{\star,\bullet}}), \tag{7b}$$

- $\alpha_{\star,\bullet}$: the tire slip angles,
- $s_{\star,\bullet}$: the slip ratios
- $\mu_{\star,\bullet}$ are the road friction coefficients
- $F_{z_{\star,\bullet}}$ are the tires normal forces.

Tyre Model

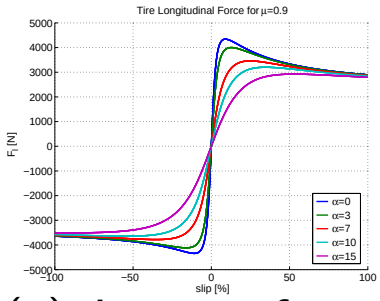


(a) Long. pure braking/driving.

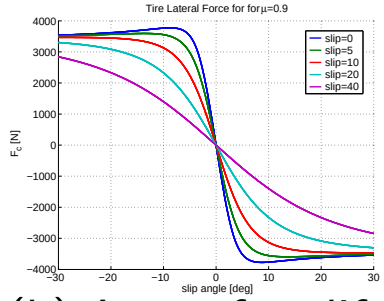


(b) Lat. pure cornering.

Figure: Longitudinal and lateral tire forces with different μ coefficient values.



(a) Long. function for different slip angle α.



(b) Lat. for different slip ratio s.

Figure: Longitudinal and lateral tire forces in combined braking/driving and

Tyre Model

The slip angle $\alpha_{\star,\bullet}$ in (7) represents the angle between the wheel velocity vector $v_{\star,\bullet}$ and the direction of the wheel itself, and can be compactly expressed as:

$$\alpha_{\star,\bullet} = \arctan \frac{v_{c_{\star,\bullet}}}{v_{l_{\star,\bullet}}}, \tag{8}$$

where $v_{c_{\star,\bullet}}$ and $v_{l_{\star,\bullet}}$ are the lateral and longitudinal wheels velocities, respectively.

$$v_{c_{\star,\bullet}} = v_{y_{\star,\bullet}} \cos\delta_\star - v_{x_{\star,\bullet}} \sin\delta_\star, \tag{9a}$$

$$v_{l_{\star,\bullet}} = v_{y_{\star,\bullet}} \sin\delta_\star + v_{x_{\star,\bullet}} \cos\delta_\star, \tag{9b}$$

Tyre Model

The velocities $v_{x_{\star,\bullet}}$ and $v_{y_{\star,\bullet}}$ for the four wheels are computed as follows:

$$v_{y_{f,l}} = \dot{y} + a\dot{\psi} \quad v_{x_{f,l}} = \dot{x} - c\dot{\psi}, \tag{10a}$$

$$v_{y_{f,r}} = \dot{y} + a\dot{\psi} \quad v_{x_{f,r}} = \dot{x} + c\dot{\psi}, \tag{10b}$$

$$v_{y_{r,l}} = \dot{y} - b\dot{\psi} \quad v_{x_{r,l}} = \dot{x} - c\dot{\psi}, \tag{10c}$$

$$v_{y_{r,r}} = \dot{y} - b\dot{\psi} \quad v_{x_{r,r}} = \dot{x} + c\dot{\psi}. \tag{10d}$$

The slip ratio $s_{\star,\bullet}$ in (7) is the defined as

$$s_{\star,\bullet} = \begin{cases} \dfrac{r_w\omega_{\star,\bullet}}{v_{l_{\star,\bullet}}} - 1 \text{ if } v_{l_{\star,\bullet}} > r_w\omega_{\star,\bullet},\ v_{l_{\star,\bullet}} \neq 0 \text{ for braking} \\ 1 - \dfrac{v_{l_{\star,\bullet}}}{r_w\omega_{\star,\bullet}} \text{ if } v_{l_{\star,\bullet}} < r_w\omega_{\star,\bullet},\ \omega_{\star,\bullet} \neq 0 \text{ for driving,} \end{cases} \tag{11}$$

where r_w and $\omega_{\star,\bullet}$ are the radius and the angular speed of the wheels, respectively, and $v_{l_{\star,\bullet}}$ are the wheel longitudinal velocities computed in (9).

Tyre Model

The wheel angular speeds $\omega_{\star,\bullet}$ in (11) are obtained by integrating the following set of differential equations:

$$J_{w_{\star,\bullet}}\dot{\omega}_{\star,\bullet} = -F_{l_{\star,\bullet}}r_w - T_{b_{\star,\bullet}} - b \cdot \omega_{\star,\bullet}, \tag{12}$$

where $J_{w_{\star,\bullet}}$ include the wheel and driveline inertias, b is the damping coefficient, $T_{b_{\star,\bullet}}$ are the braking torques at the braking pads.$F_{z_{\star,\bullet}}$ in (7) are the normal forces Assumption: the normal forces $F_{z_{\star,\bullet}}$ are constant and distributed between the front and rear axles as

$$F_{z_{f,\bullet}} = \frac{bmg}{2(a+b)}, \quad F_{z_{r,\bullet}} = \frac{amg}{2(a+b)}.$$

Final Model

4-Wheel Model

$$\dot{\xi}(t) = f_{\mu(t)}(\xi(t), u(t)),$$

where the state and input vectors are

- $\xi = [\dot{y},\ \dot{x},\ \psi,\ \dot{\psi},\ Y,\ X,\ \omega_{f,l},\ \omega_{f,r},\ \omega_{r,l},\ \omega_{r,r}]$
- $u = [\delta_f,\ T_{b_{f,l}},\ T_{b_{f,r}},\ T_{b_{r,l}},\ T_{b_{r,r}}]$,

and $\mu(t) = [\mu_{f,l}(t),\ \mu_{f,r}(t),\ \mu_{r,l}(t),\ \mu_{r,r}(t)]$.