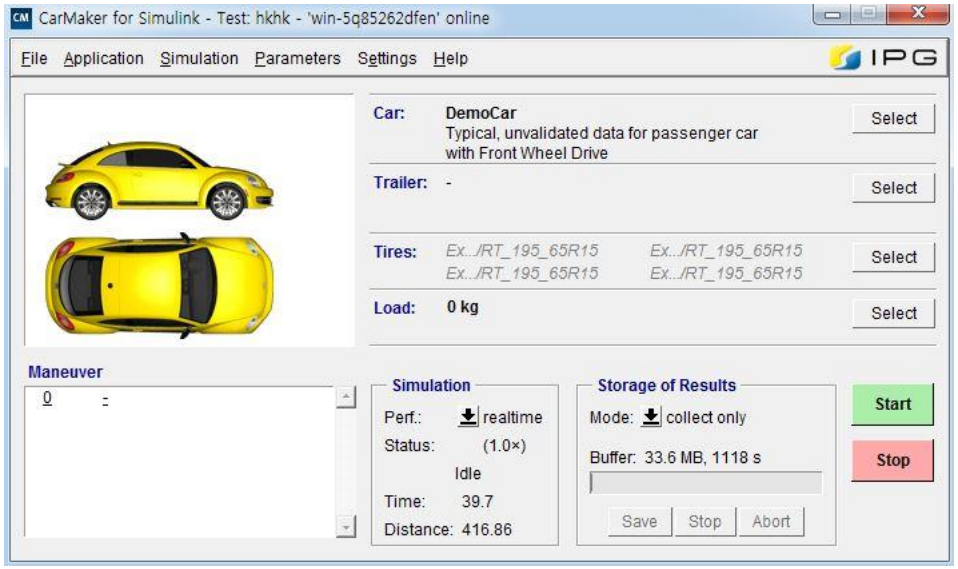
Vehicle Control

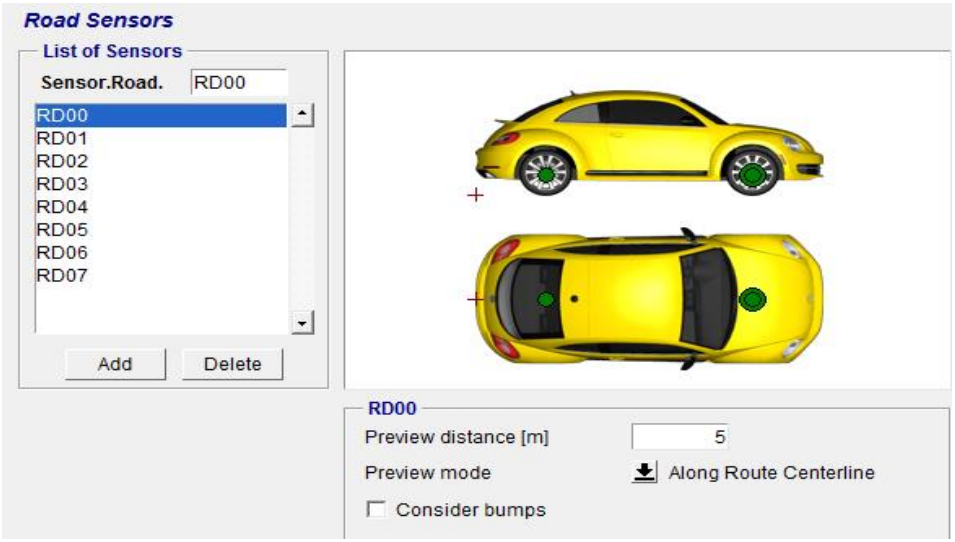
Vehicle Dynamics를 이용한 차량 위치 추정, Pure pursuit을 이용한 Steering Control

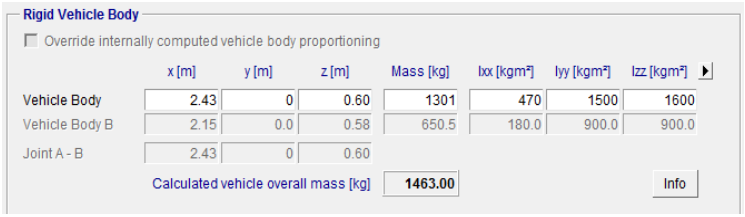
김은찬 이민성 권나현

CONTENTS

1. CAR MAKER


CarMaker for Simulink - Test: hkhk - 'win-5q85262dfen' online
File Application Simulation Parameters Settings Help
IPG
Car: DemoCar
Typical, unvalidated data for passenger car
with Front Wheel Drive
Select
Trailer: -
Select
Tires: Ex.../RT_195_65R15 Ex.../RT_195_65R15
Ex.../RT_195_65R15 Ex.../RT_195_65R15
Select
Load: 0 kg
Select
Maneuver
0 -
Simulation
Perf.: realtime
Status: (1.0×)
Idle
Time: 39.7
Distance: 416.86
Storage of Results
Mode: collect only
Buffer: 33.6 MB, 1118 s
Save Stop Abort
Start
Stop

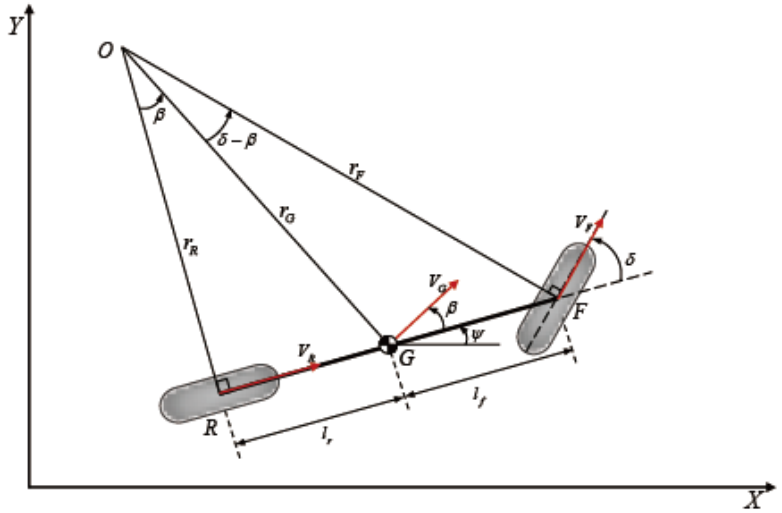

Road Sensors
List of Sensors
Sensor.Road. RD00
RD00
RD01
RD02
RD03
RD04
RD05
RD06
RD07
Add
Delete
RD00
Preview distance [m] 5
Preview mode Along Route Centerline
Consider bumps

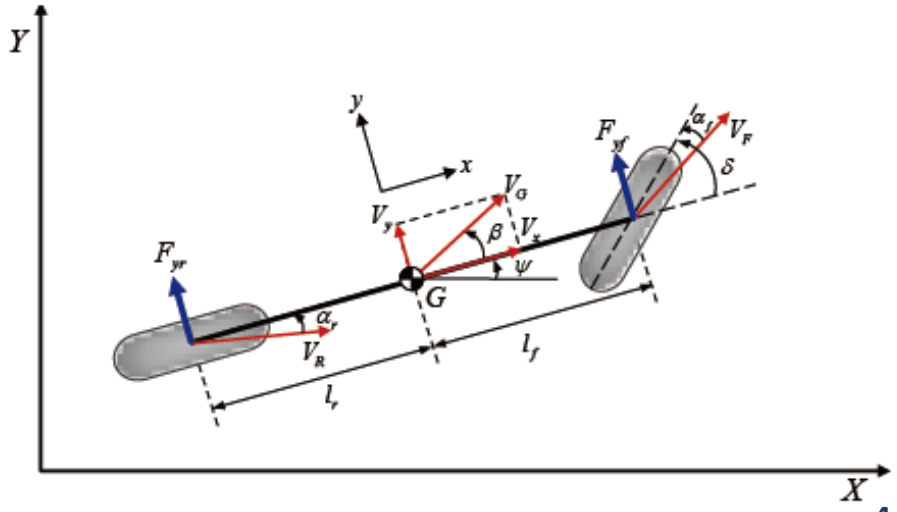

Rigid Vehicle Body
Override internally computed vehicle body proportioning
x [m] y [m] z [m] Mass [kg] Ixx [kgm²] Iyy [kgm²] Izz [kgm²]
Vehicle Body 2.43 0 0.60 1301 470 1500 1600
Vehicle Body B 2.15 0.0 0.58 650.5 180.0 900.0 900.0
Joint A - B 2.43 0 0.60
Calculated vehicle overall mass [kg] 1463.00
Info

2. Vehicle Localization

Vehicle Model

- **Kinematic Model**
  - **Slip X**
- **Dynamic Model**
  - **Slip O**





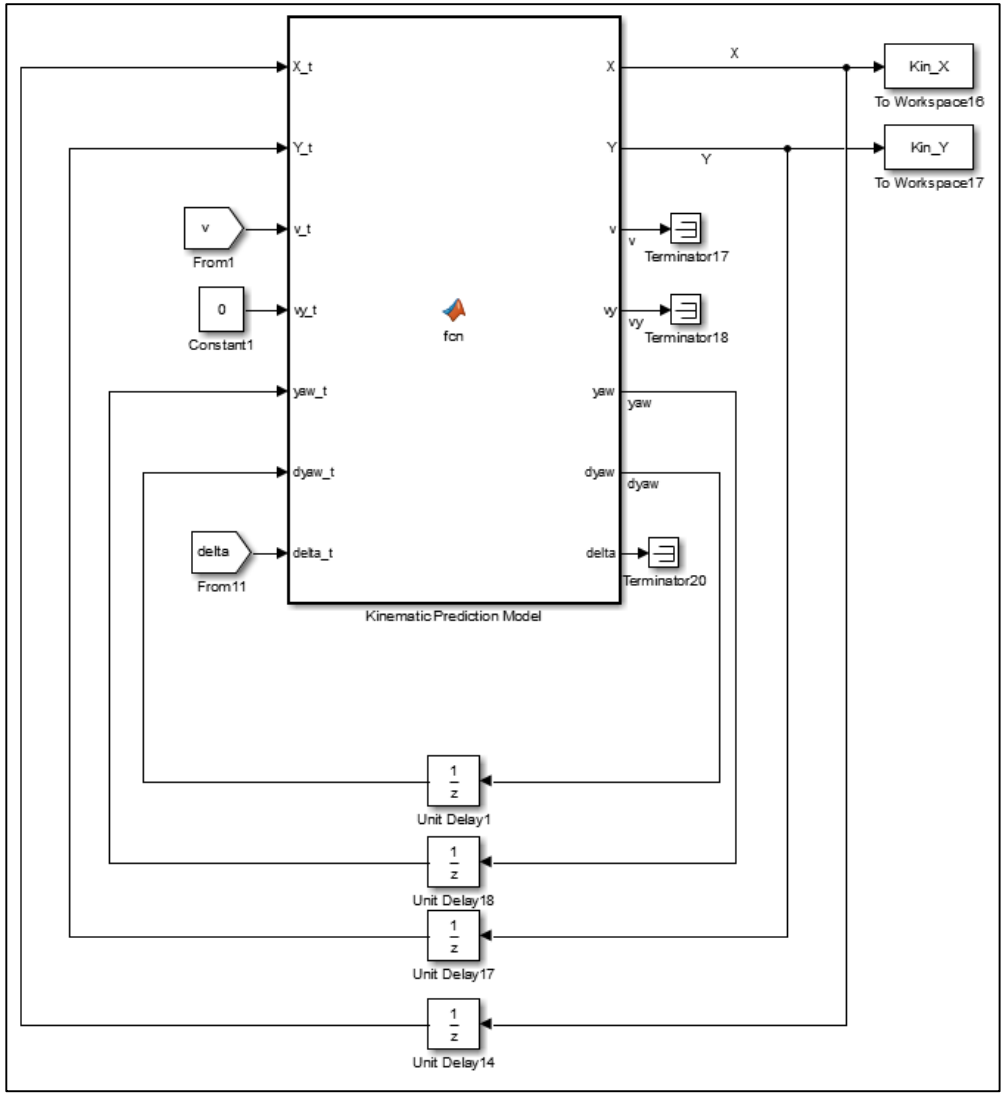
2-1) Kinematic Model

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} V_G \cos(\psi + \beta) \\ V_G \sin(\psi + \beta) \\ V_G \cos(\beta) \tan(\delta) / (l_f + l_r) \end{bmatrix}$$

where

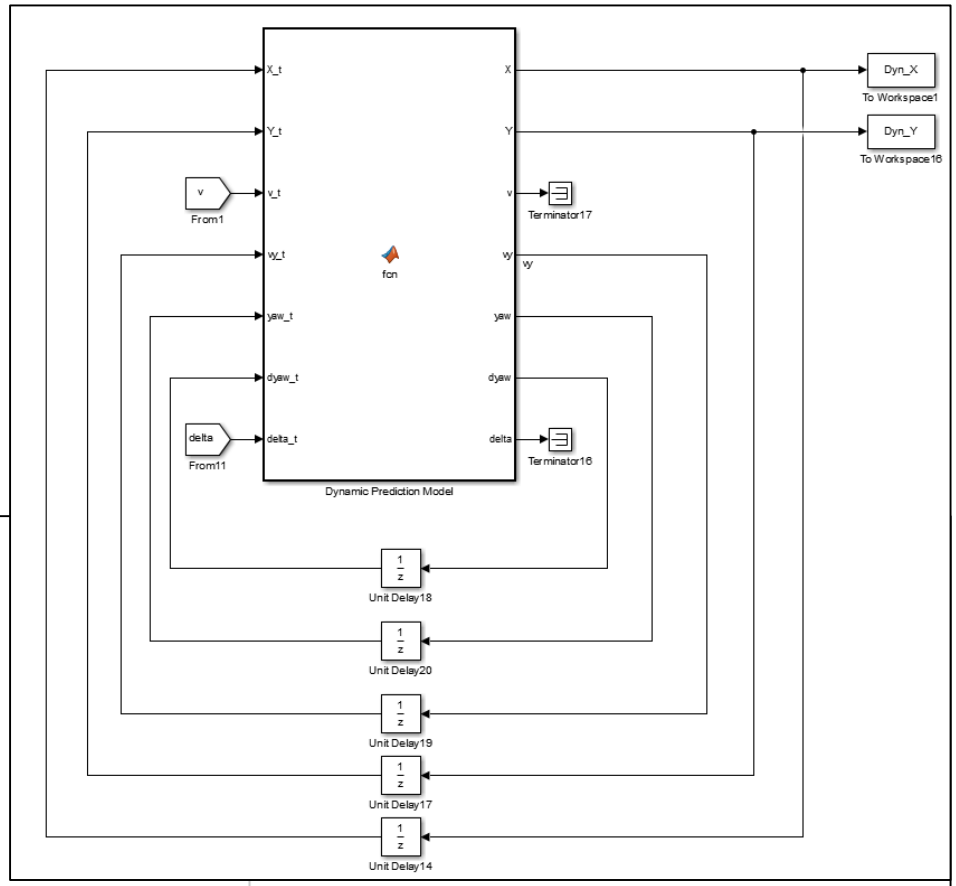
$$\beta = \tan^{-1}\left(\frac{l_r \tan(\delta)}{l_f + l_r} \right)$$

```
X = X_t + v_t * cos(beta + yaw_t) * dT;
Y = Y_t + v_t * sin(beta + yaw_t) * dT;
yaw = yaw_t + v_t * tan(delta_t)/(lr+lf) * dT;
dyaw = dyaw_t;
v = v_t;
vy = vy_t;
delta = delta_t;
```



2-2) Dynamic Model

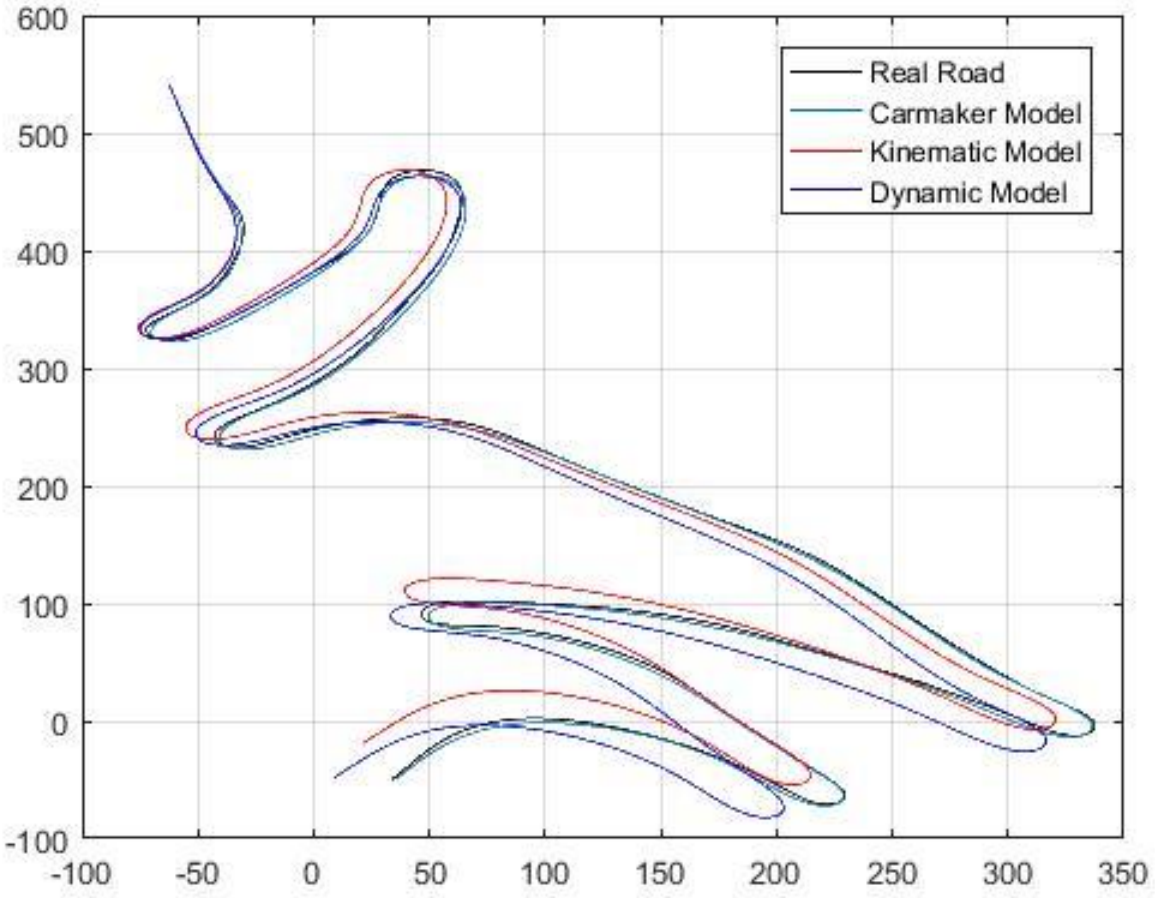
$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \ddot{\psi} \\ \dot{V}_y \end{bmatrix} = \begin{bmatrix} V_x\cos(\psi) - V_y\sin(\psi) \\ V_y\cos(\psi) + V_x\sin(\psi) \\ -\left(\frac{2C_f l_f^2 + 2C_r l_r^2}{I_z V_x}\right)\dot{\psi} - \left(\frac{2C_f l_f - 2C_r l_r}{I_z V_x}\right)V_y + \frac{2C_f l_f}{I_z} \\ -\left(V_x + \frac{2C_f l_f - 2C_r l_r}{V_x m}\right)\dot{\psi} - \left(\frac{2C_f + 2C_r}{V_x m}\right)V_y + \frac{2C_f\,\delta}{m} \end{bmatrix}$$




```
if v_t >= 0 && v_t <= 1
    v_tt = 1;
elseif v_t < 0 && v_t >= -1
    v_tt = -1;
else
    v_tt = v_t;
end

X = X_t + (v_t * cos(yaw_t) - vy_t * sin(yaw_t)) * dT;
Y = Y_t + (vy_t * cos(yaw_t) + v_t * sin(yaw_t)) * dT;
v = v_t;
vy = vy_t + (-(v_t + 2*(Cf*lf - Cr*lr)/(v_tt*m))*dyaw_t - 2*(Cf + Cr)/(v_tt*m)*vy_t + 2*Cf*delta_t/m) * dT;
yaw = yaw_t + dyaw_t * dT;
dyaw = dyaw_t + (-(2*(Cf*lf^2 + Cr*lr^2)/(Iz*v_tt))*dyaw_t - 2*((Cf*lf - Cr*lr)/(Iz*v_tt))*vy_t + 2*Cf*lf*delta_t/Iz) * dT;
delta = delta_t;
```

2-3) Analysis

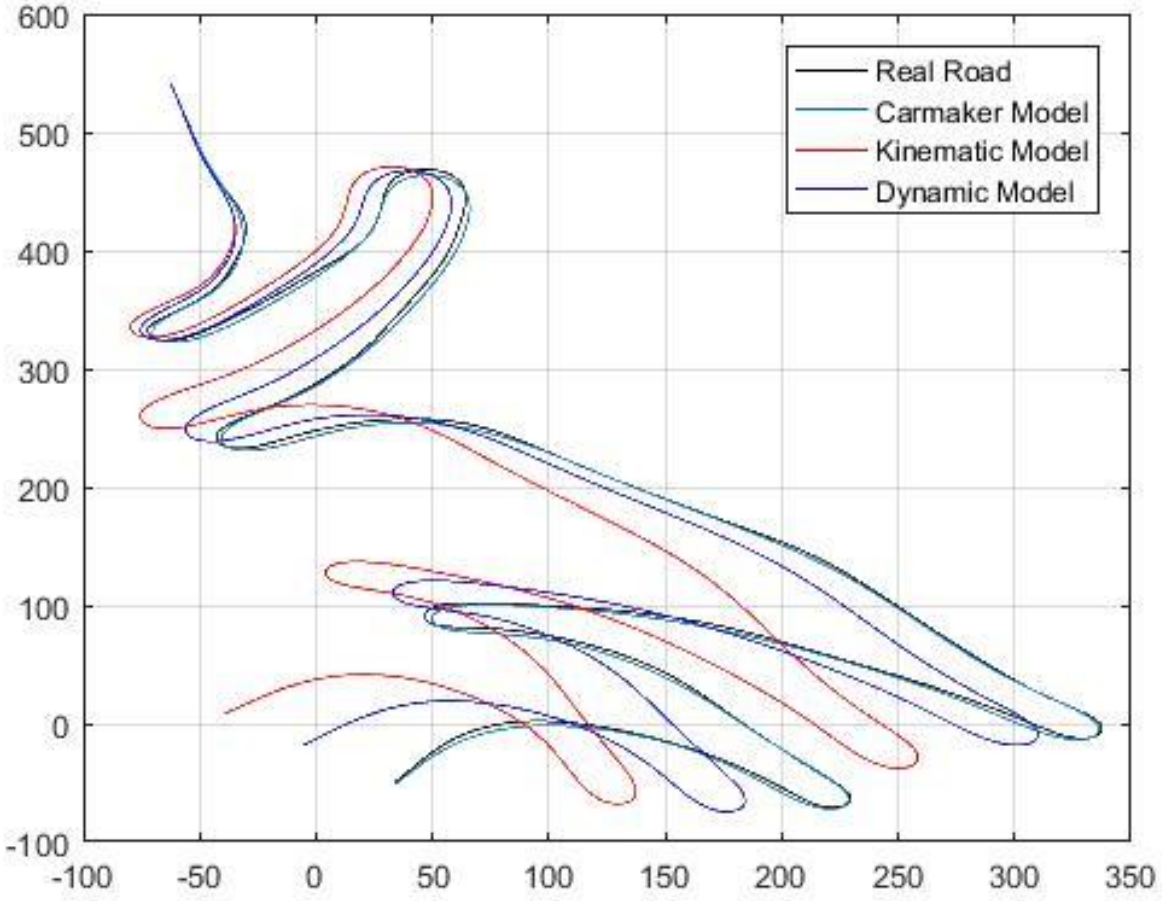
Low speed (20km/h)



Kinematic Model is reasonable for low-speed vehicle motion.

2-3) Analysis

High speed (80km/h)



Dynamic Model is reasonable for high-speed vehicle motion.

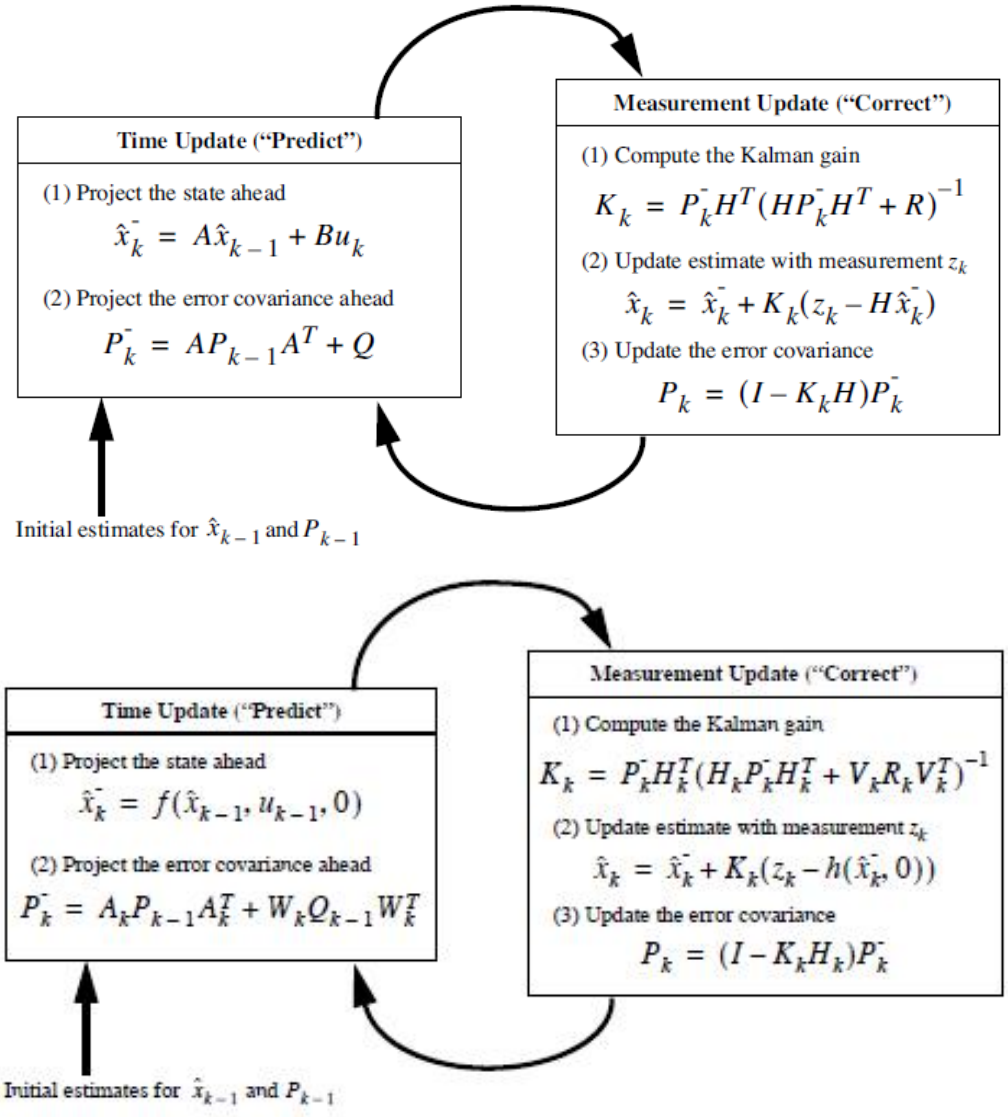
3. Extended Kalman Filter

Linear System

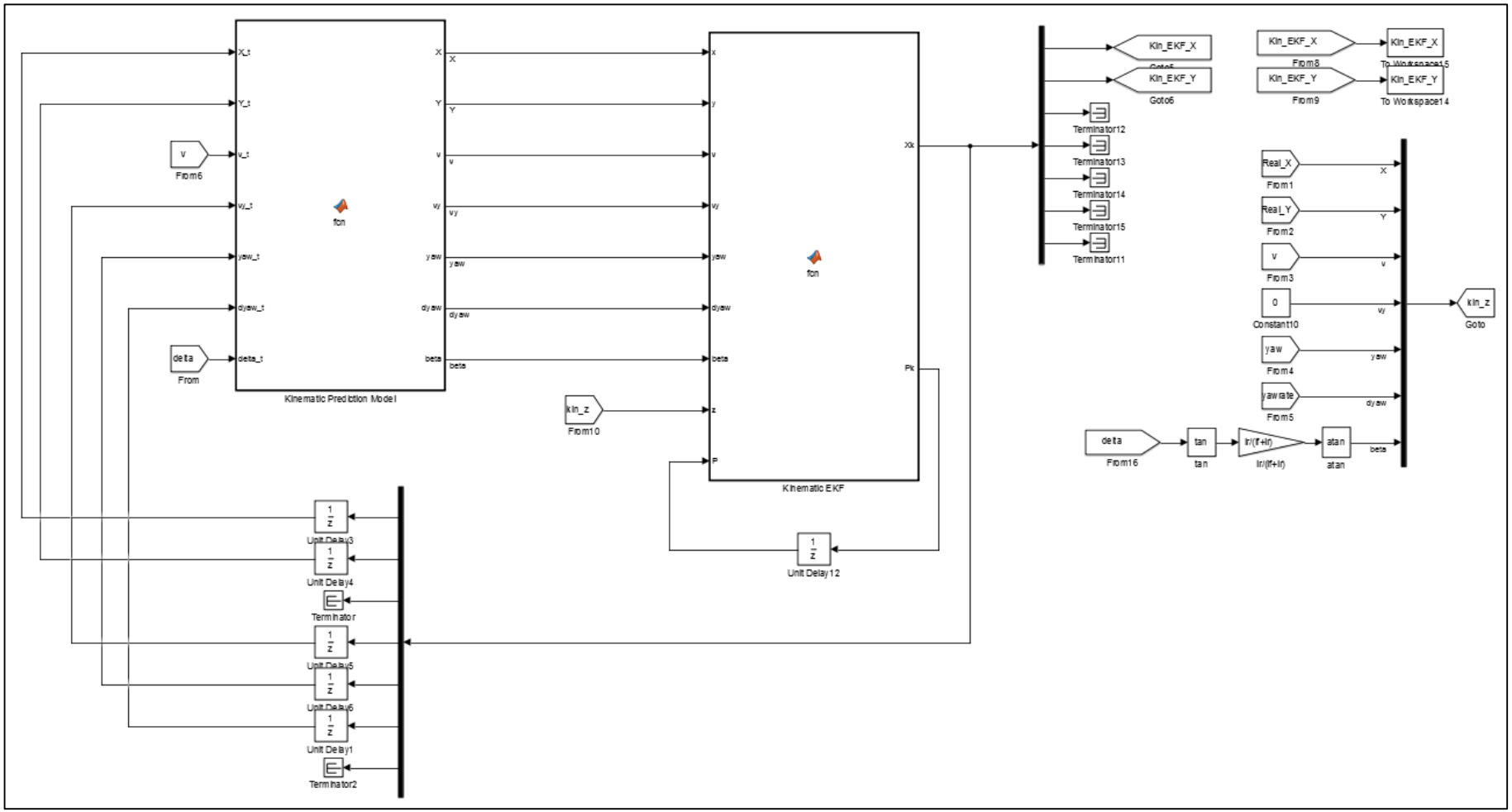
Kalman Filter

Extended Kalman Filter

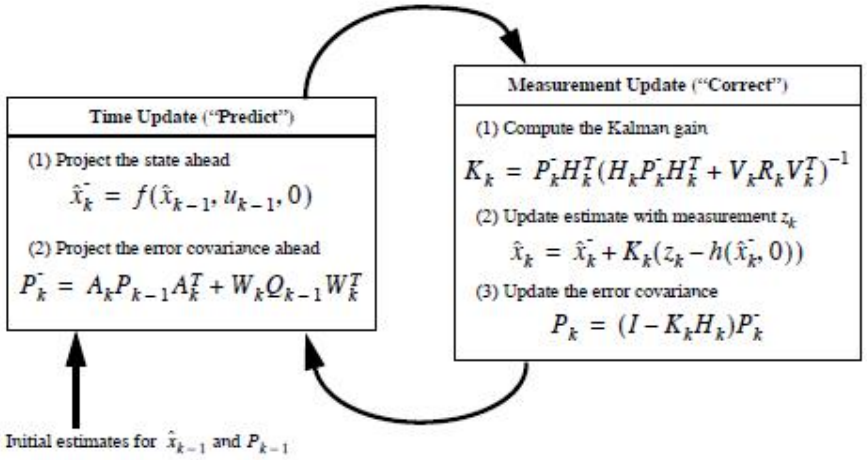
Nonlinear System



3-1) Kinematic Model with EKF



3-1) Kinematic Model with EKF



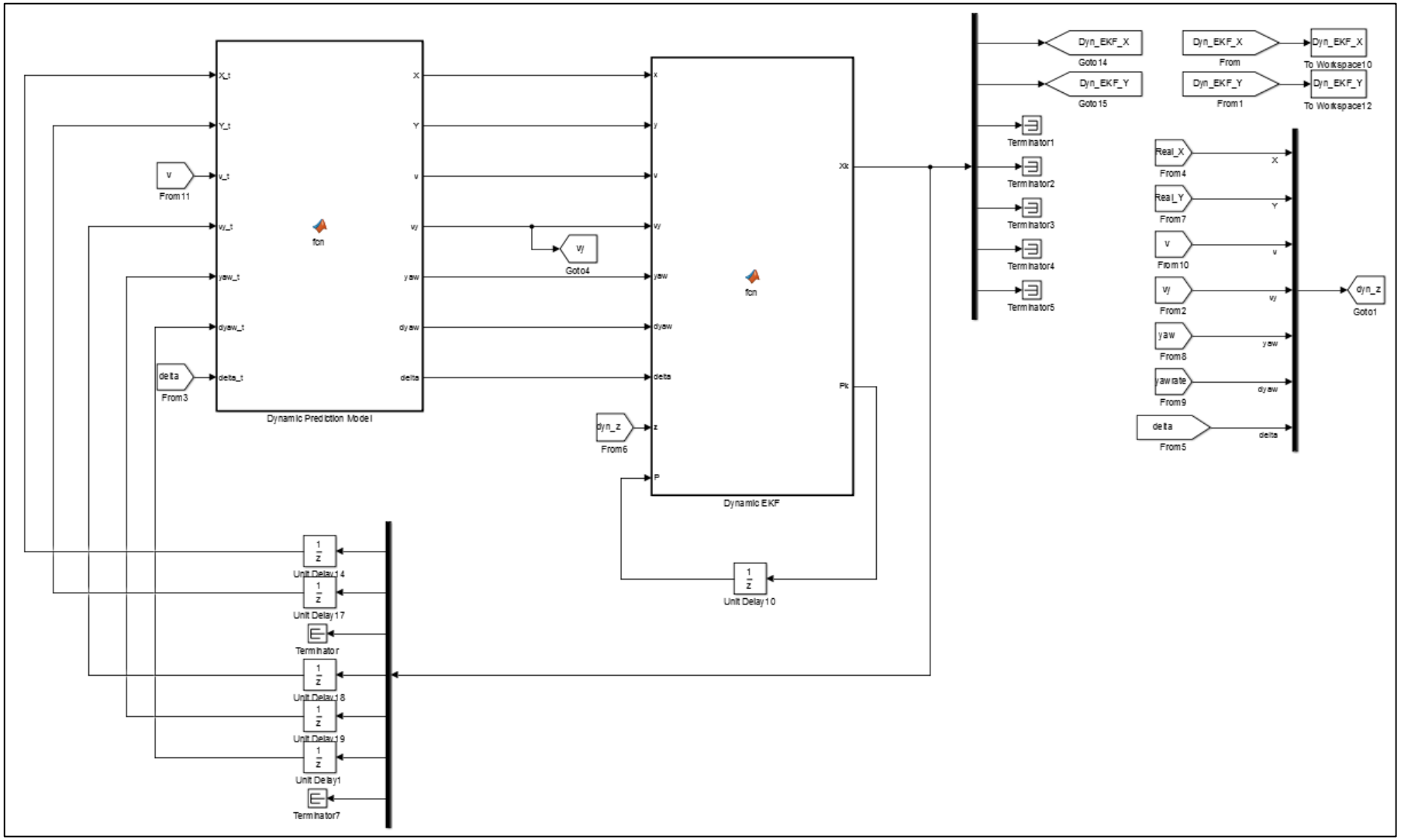
$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} V_G \cos(\psi + \beta) \\ V_G \sin(\psi + \beta) \\ V_G \cos(\beta) \tan(\delta) / (l_f + l_r) \end{bmatrix}$$

where

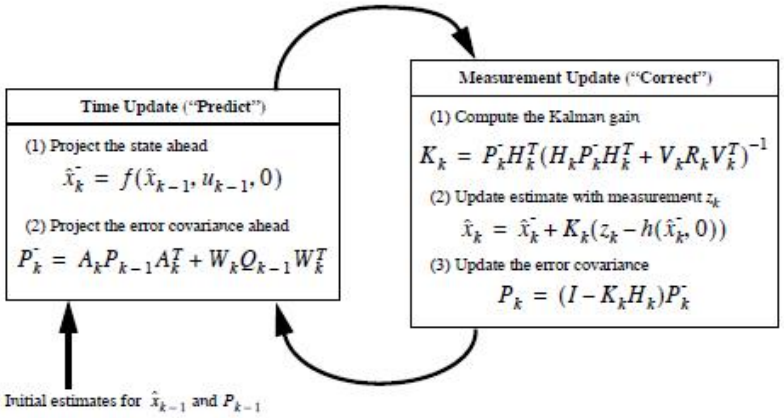
$$\beta = \tan^{-1}\left(\frac{l_r \tan(\delta)}{l_f + l_r} \right)$$

```
X = zeros(7,1);

X(1) = x;
X(2) = y;
X(3) = v;
X(4) = vy;
X(5) = yaw;
X(6) = dyaw;
X(7) = beta;

F = [0, 0, cos(yaw+beta), 0, -v*sin(yaw+beta), 0, -v*sin(yaw+beta); ...
    0, 0, sin(yaw+beta), 0, v*cos(yaw+beta), 0, v*cos(yaw+beta); ...
    0, 0, 0, 0, 0, 0, 0; ...
    0, 0, 0, 0, 0, 0, 0; ...
    0, 0, 1/lr*sin(beta), 0, 0, 0, v/lr*cos(beta); ...
    0, 0, 0, 0, 0, 0, 0; ...
    0, 0, 0, 0, 0, 0, 0];

Pi = eye(7) + F*dT;

Q = diag([0, 0, 0.1^2, 0, (0.05*pi/180*dT)^2, (0.05*pi/180)^2, (0.1*pi/180)^2]);

P = Pi * P * Pi' + Q;
R = diag([2^2, 2^2, 0.1^2, 0.1^2, 0, 0, (0.1*pi/180)^2]);
H = eye(7);
S = H*P*H' + R;
K = P*H'/S;
Xk = X + K*(z - H*X);

Pk = (eye(7) - K*H)*P;
```

3-2) Dynamic Model with EKF



3-2) Dynamic Model with EKF

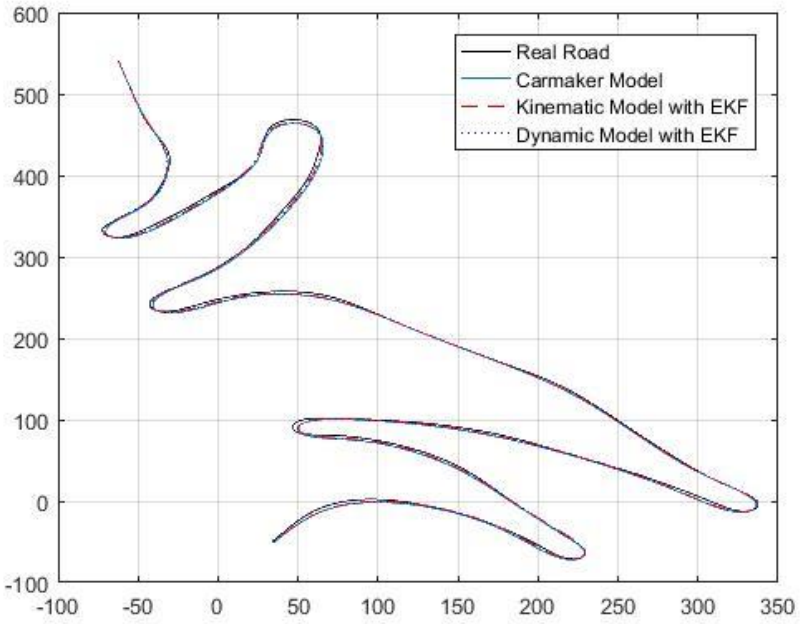


$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \ddot{\psi} \\ \dot{V}_y \end{bmatrix} = \begin{bmatrix} V_x\cos(\psi) - V_y\sin(\psi) \\ V_y\cos(\psi) + V_x\sin(\psi) \\ -\left(\frac{2C_f l_f^2 + 2C_r l_r^2}{I_z V_x}\right)\dot{\psi} - \left(\frac{2C_f l_f - 2C_r l_r}{I_z V_x}\right)V_y + \frac{2C_f l_f}{I_z} \\ -\left(V_x + \frac{2C_f l_f - 2C_r l_r}{V_x m}\right)\dot{\psi} - \left(\frac{2C_f + 2C_r}{V_x m}\right)V_y + \frac{2C_f \delta}{m} \end{bmatrix}$$

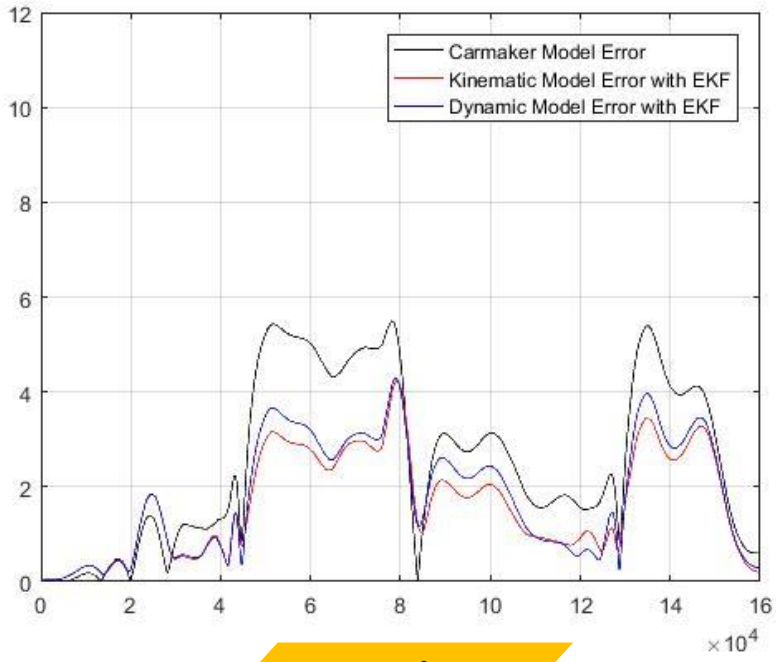
```
X = zeros(7,1);

X(1) = x;
X(2) = y;
X(3) = v;
X(4) = vy;
X(5) = yaw;
X(6) = dyaw;
X(7) = delta;

if v >= 0 && v <= 1
    vt = 1;
elseif v < 0 && v >= -1
    vt = -1;
else
    vt = v;
end

F = [0, 0, cos(yaw), -sin(yaw), -v*sin(yaw)-vy*cos(yaw), 0, 0; ...
    0, 0, sin(yaw), cos(yaw), v*cos(yaw)-vy*sin(yaw), 0, 0; ...
    0, 0, 0, 0, 0, 0, 0; ...
    0, 0, (-1+2*(Cf*lf-Cr*lr)/(m*vt^2))*dyaw+2*(Cf+Cr)/(m*vt^2)*vy,...
    -2*(Cf+Cr)/(m*vt), 0, -(v+2*(Cf*lf-Cr*lr)/(m*vt)), 2*Cf/m; 0, 0, 0, 0, 0, 1, 0; ...
    0, 0, 2*(Cf*lf^2+Cr*lr^2)/(Iz*vt^2)*dyaw+2*(Cf*lf-Cr*lr)/(Iz*vt^2)*vy, -2*(Cf*lf-Cr*lr)/(Iz*vt), 0, ...
    -2*(Cf*lf^2+Cr*lr^2)/(Iz*vt), 2*Cf*lf/Iz; 0, 0, 0, 0, 0, 0, 0];

Pi = eye(7) + F*dT;

Q = diag([0, 0, 0.1^2, 0, (0.05*pi/180*dT)^2, (0.05*pi/180)^2, (0.1*pi/180)^2]);

P = Pi * P * Pi' + Q;
R = diag([2^2, 2^2, 0.1^2, 0.1^2, 0, 0, 0]);
H = eye(7);
S = H*P*H' + R;
K = P*H'/S;
Xk = X + K*(z - H*X);

Pk = (eye(7) - K*H)*P;
```

3-3) Analysis

Low speed (20km/h)



Road Plot

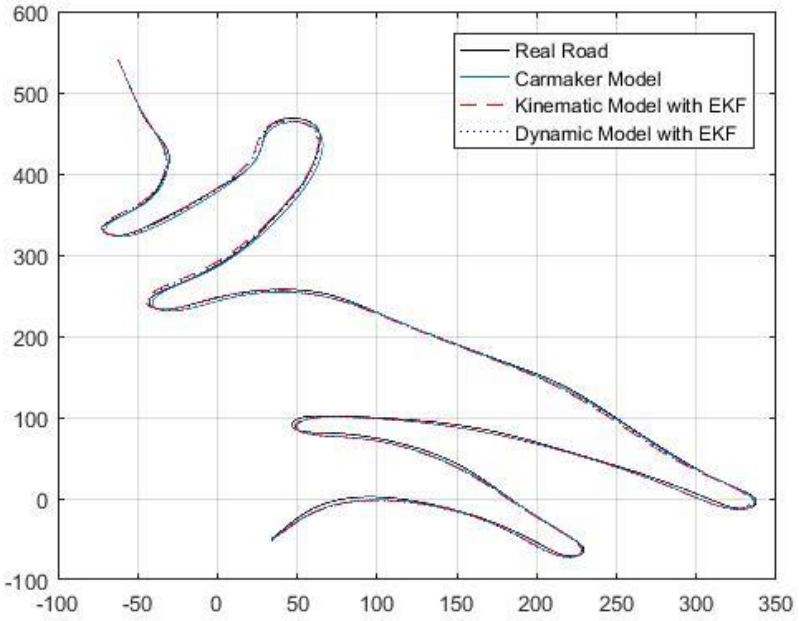


Error Plot

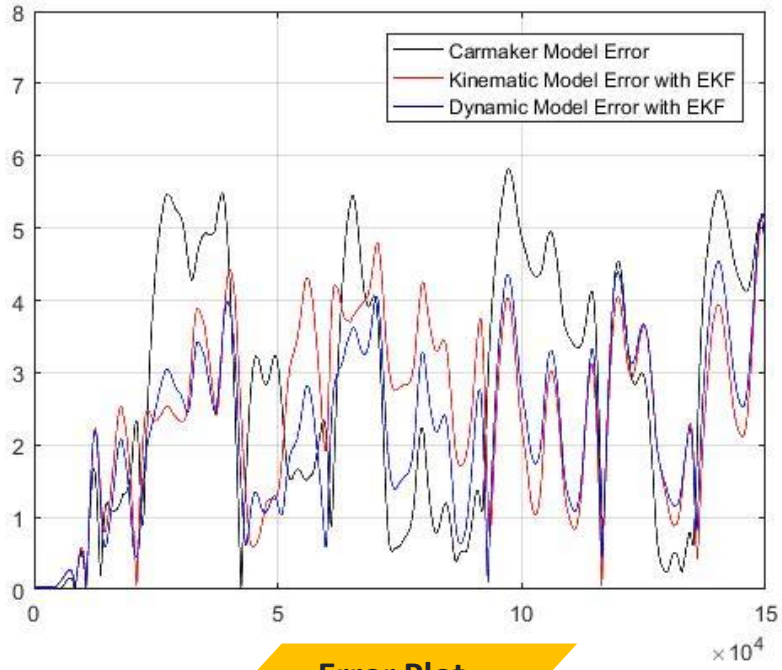
The error is noticeably reduced.
But Kinematic Model is reasonable for low-speed vehicle motion.

3-3) Analysis

High speed (80km/h)



Road Plot

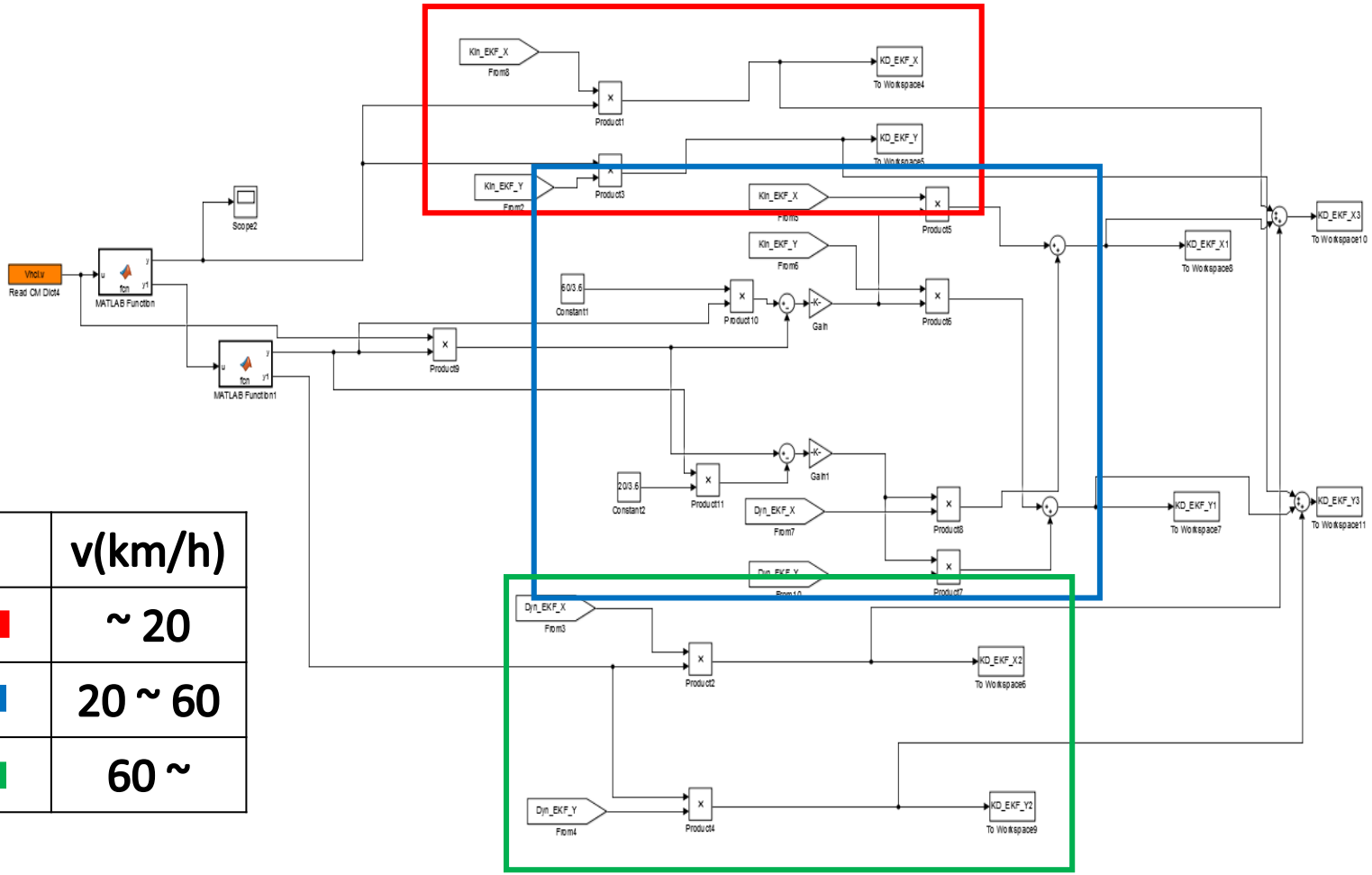


Error Plot

The error is noticeably reduced.
But Dynamic Model is reasonable for high-speed vehicle motion.

3-3) Integration Model

Kinematic + Dynamic (Velocity)

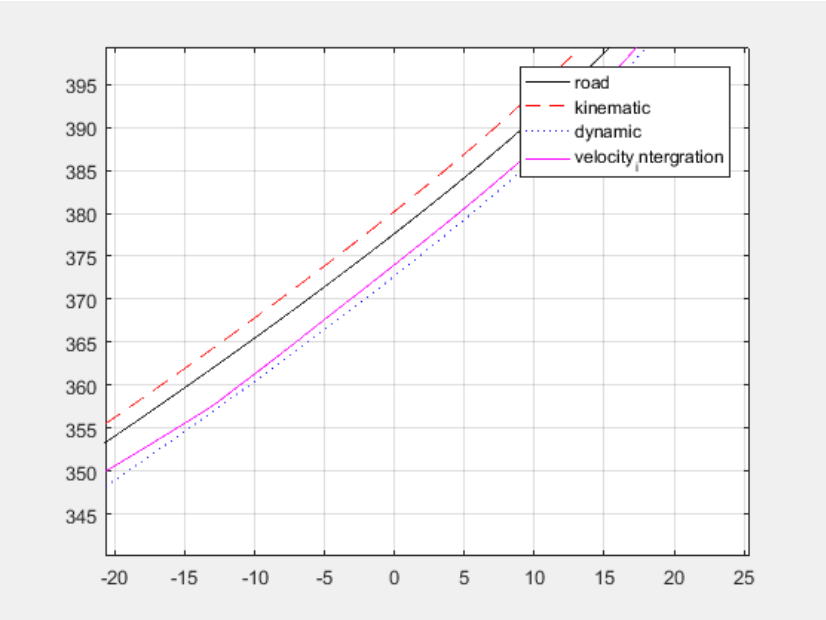


	v(km/h)
	~ 20
	20 ~ 60
	60 ~

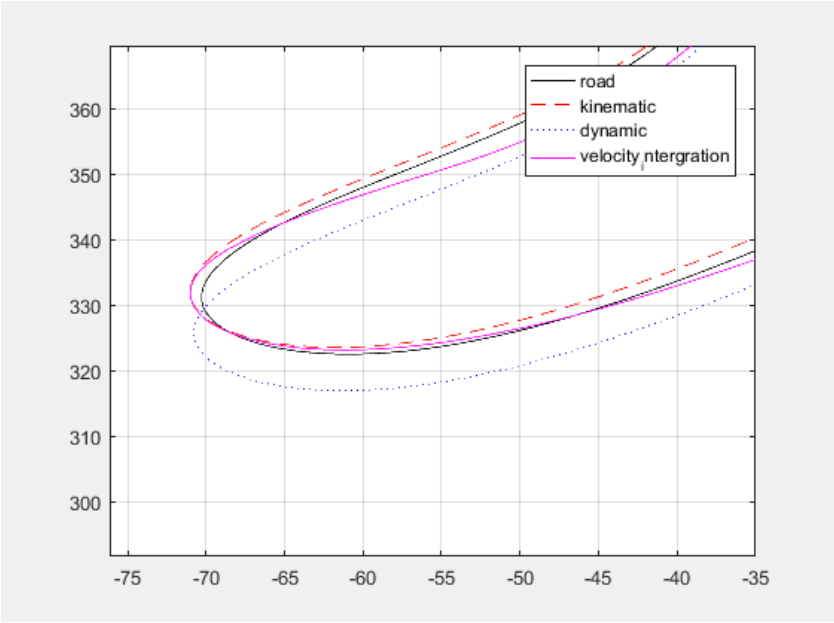
3-3) Integration Model

Kinematic + Dynamic (Velocity)

Analysis in variable speed



Straight Road

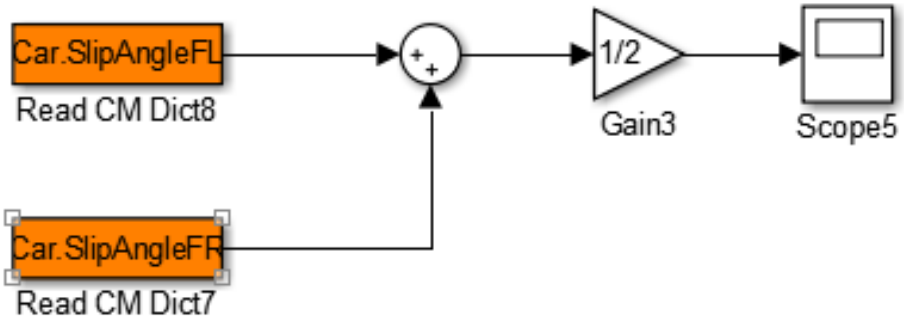


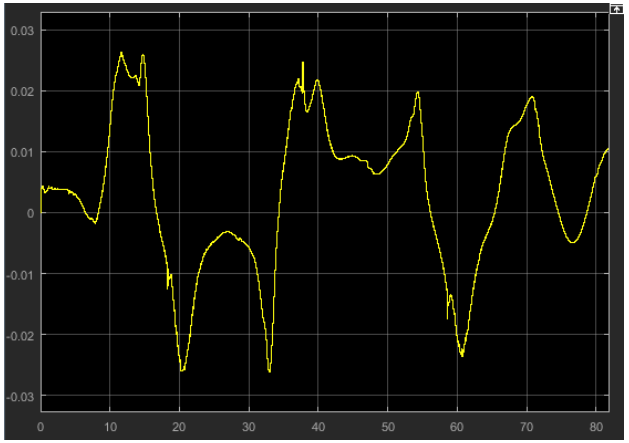
Curve

Integration Model according to velocity is reasonable for curve.

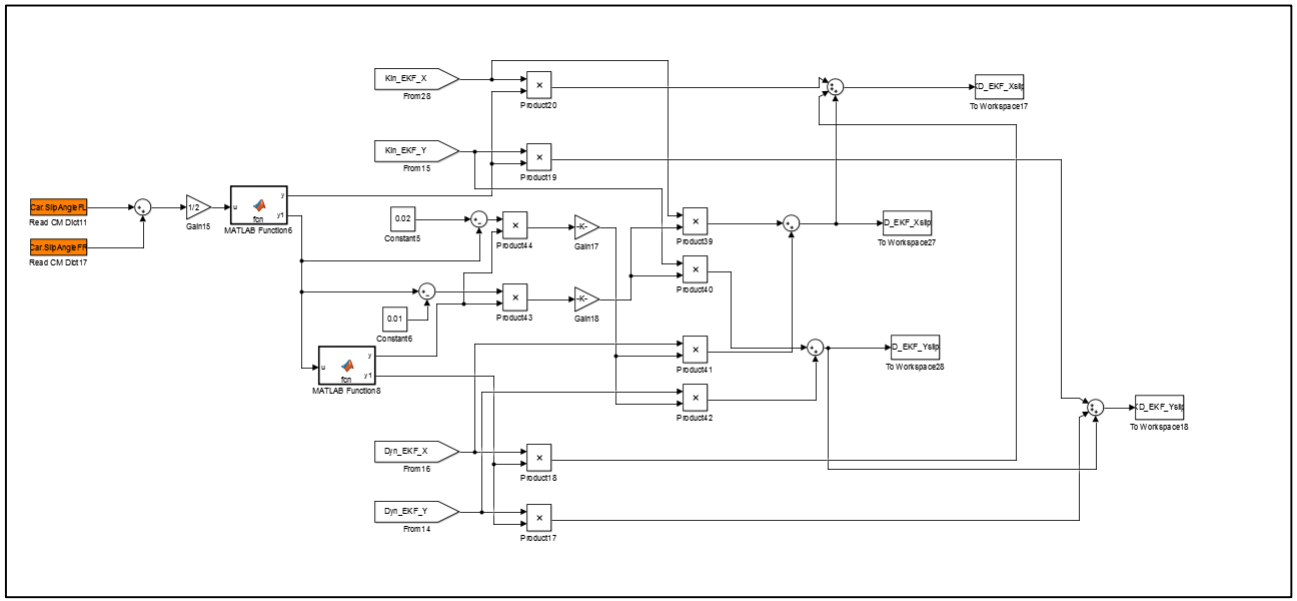
3-3) Integration Model

Kinematic + Dynamic (Slip)





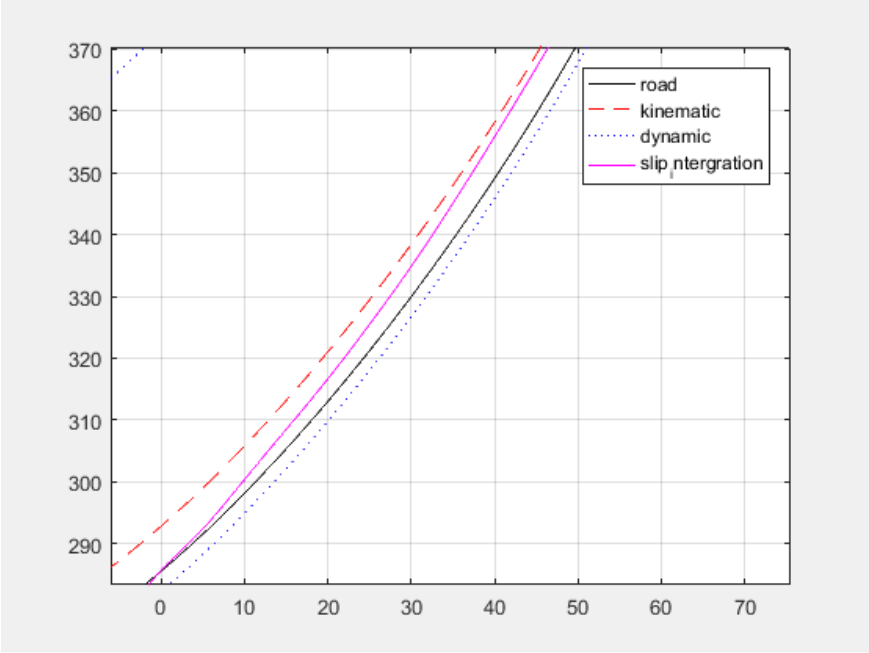
-0.03 ~ 0.03



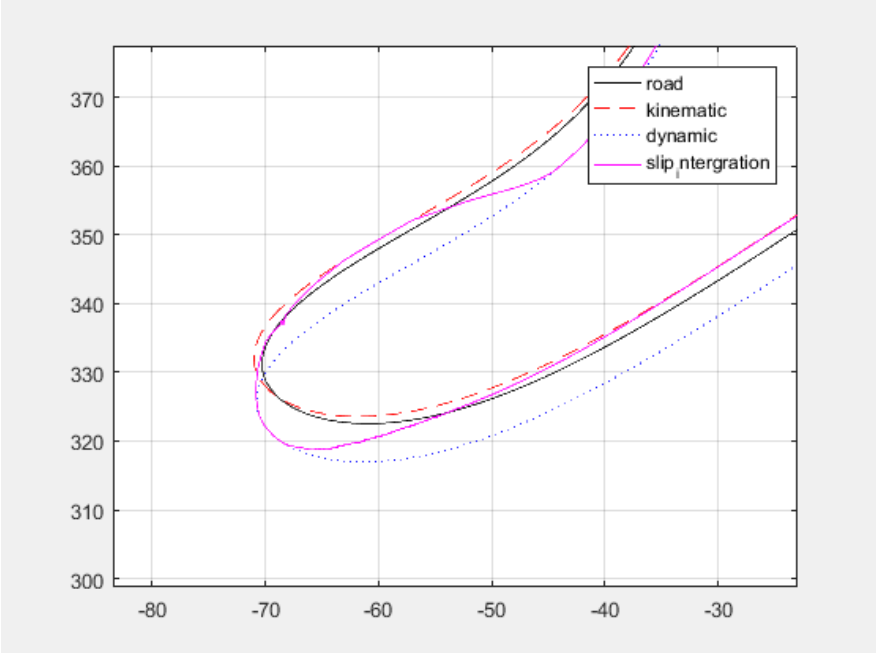
3-3) Integration Model

Kinematic + Dynamic (Slip)

Analysis in variable speed



Straight Road

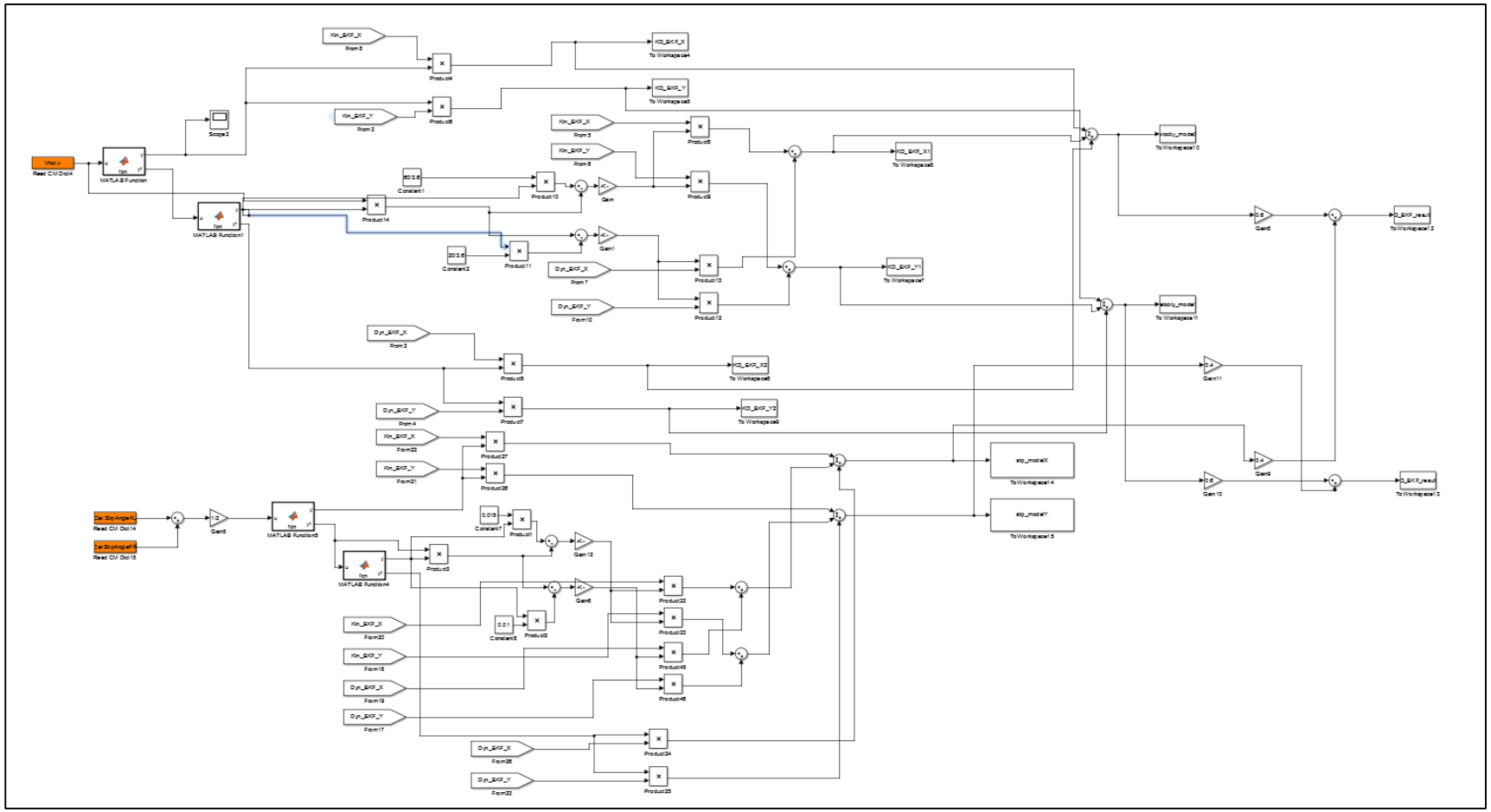


Curve

Integration Model according to slip is reasonable for straight road.

3-3) Integration Model

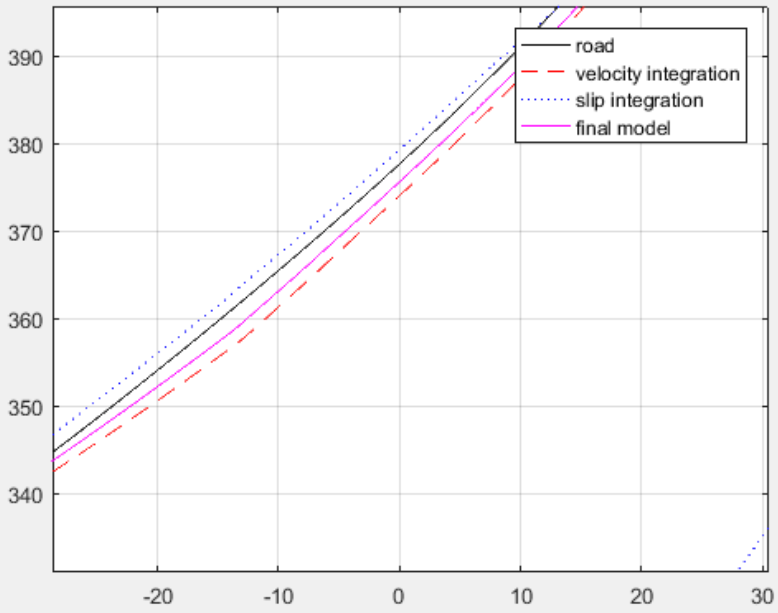
Kinematic + Dynamic (Velocity + Slip)



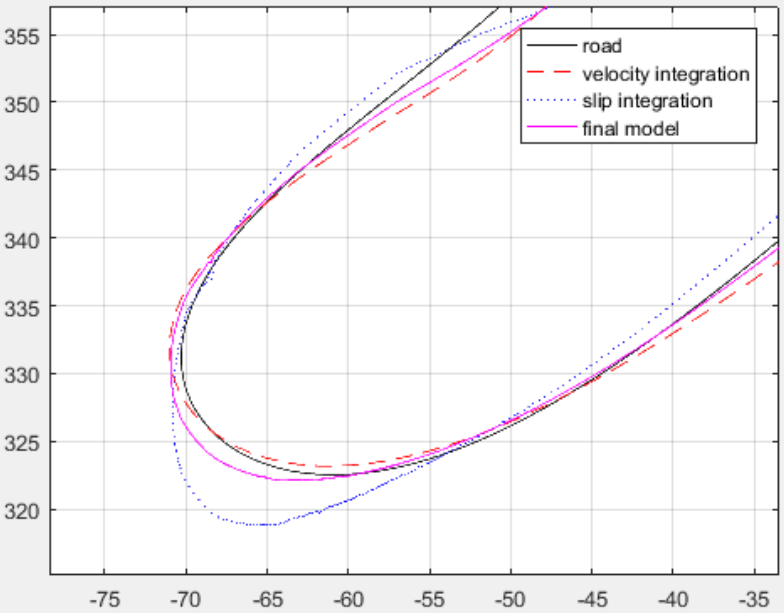
3-3) Integration Model

Kinematic + Dynamic (Velocity 7 + Slip 3)

Analysis in variable speed



Straight Road

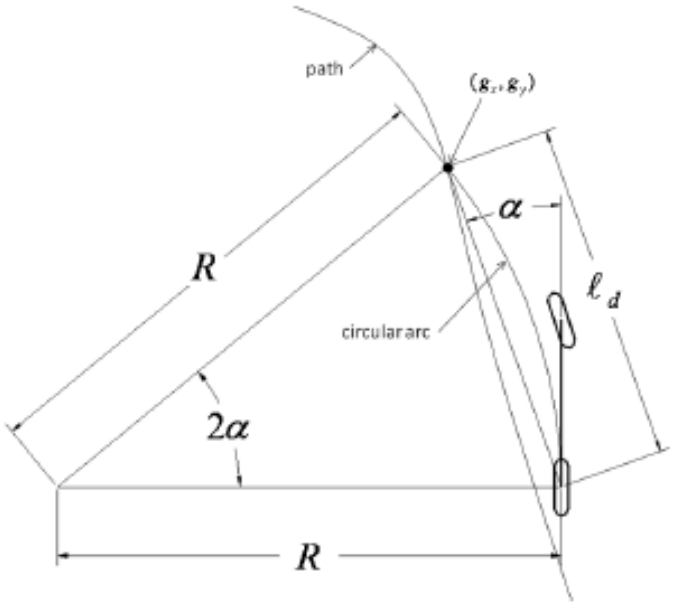


Curve

As a result of the tuning, the integration model with the ratio of 7: 3 was the best along the road.

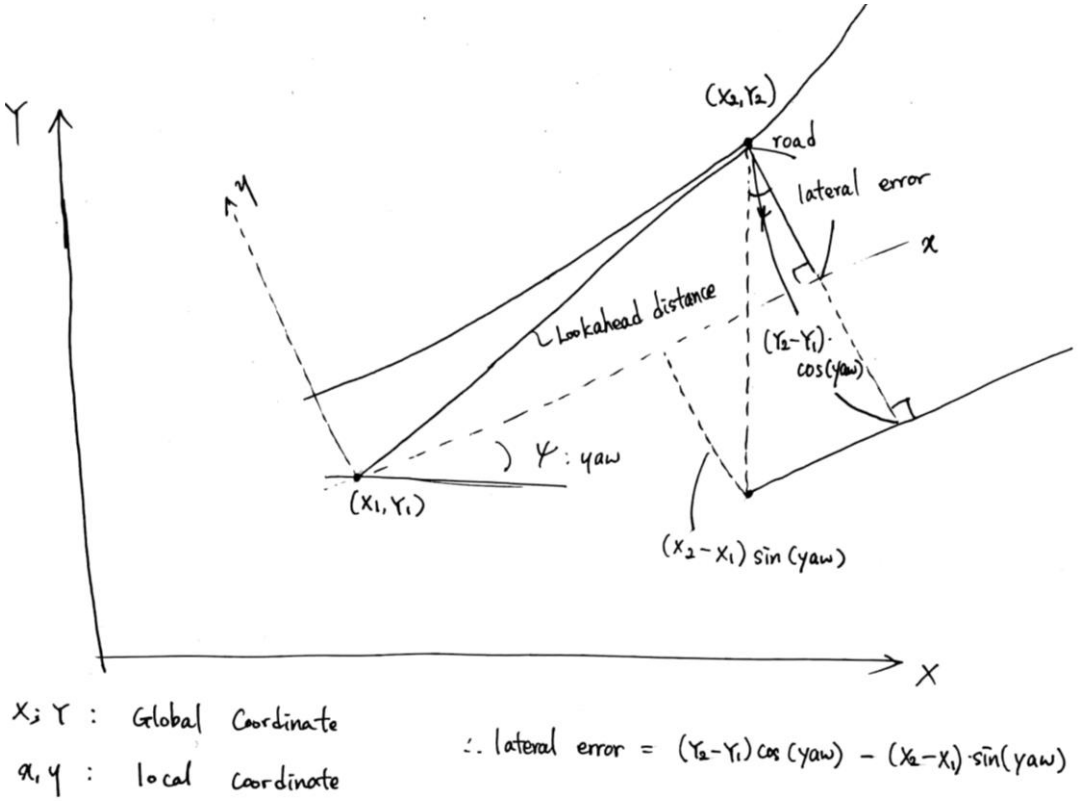
4. Steering Control

Pure pursuit



$$\delta(t) = \tan^{-1}\left(\frac{2L\sin(\alpha(t))}{\ell_d}\right)$$

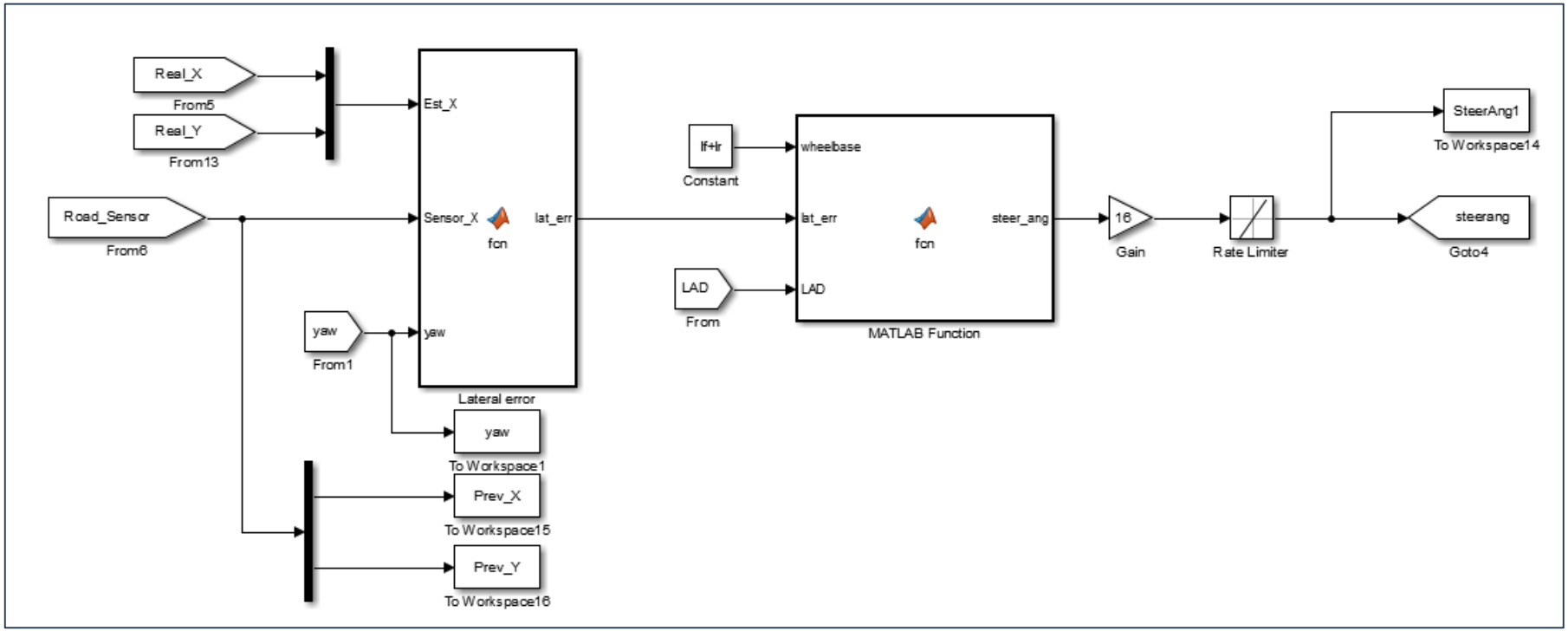
$$\sin(\alpha) = \frac{e_{\ell_d}}{\ell_d}$$



$$e_{l_d} = (Y_2 - Y_1)\cos(yaw) - (X_2\text{-}X_1)\cos(yaw)$$

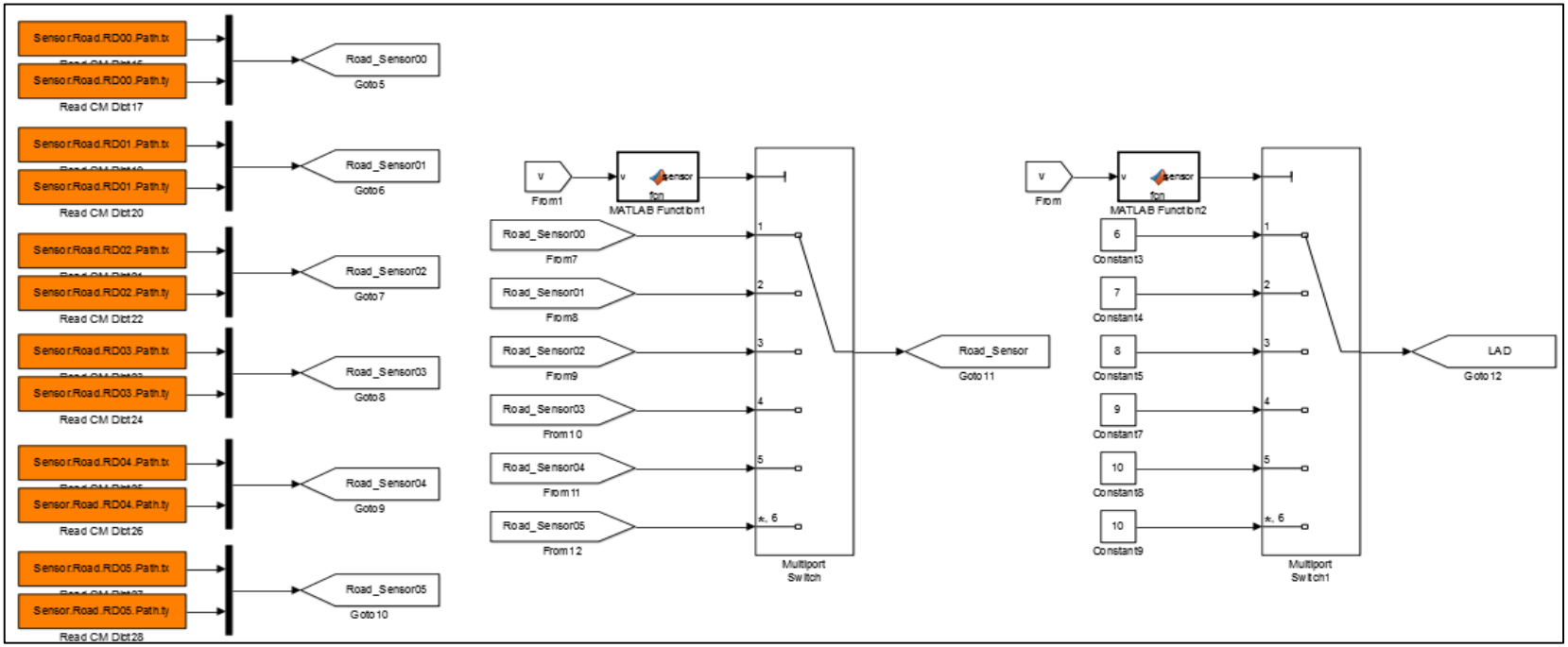
4. Steering Control

Pure pursuit



4. Steering Control

Look-ahead distance control




```
if v <= 30/3.6
    sensor = 1;
elseif v <= 40/3.6
    sensor = 2;
elseif v <= 60/3.6
    sensor = 3;
elseif v <= 80/3.6
    sensor = 4;
elseif v <= 100/3.6
    sensor = 5;
else
    sensor = 6;
end
```

4-1) Video

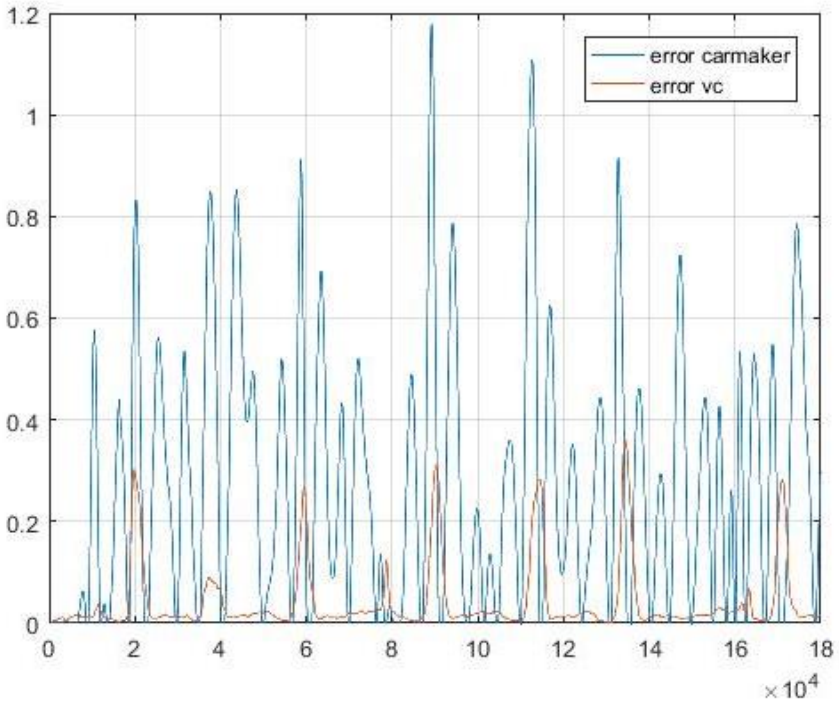
Carmaker model

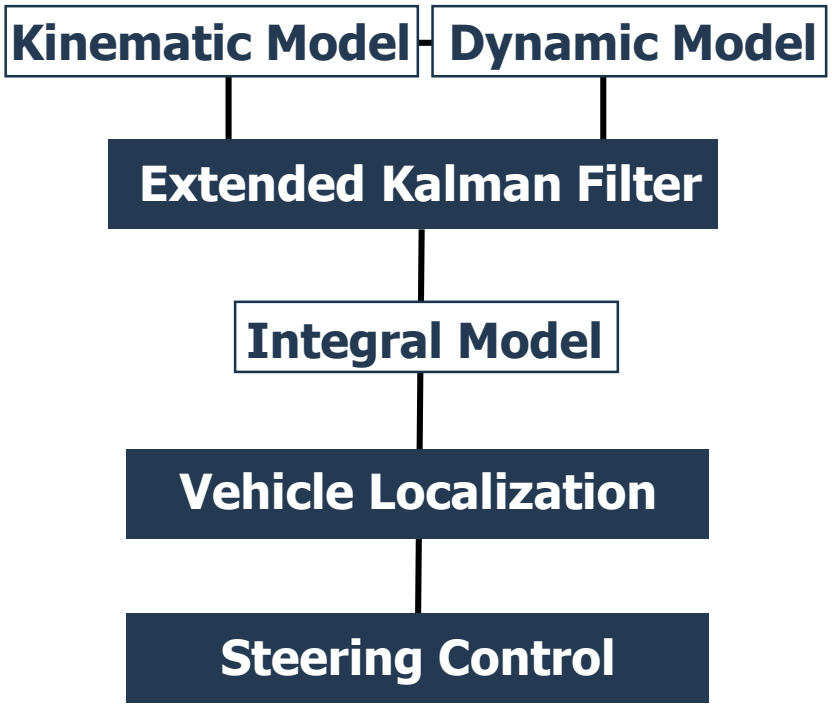
Integration Model

START

4-2) Analysis & Conclusion

Error Analysis





Better steering control than the original model

Reference

Jarrod M. Snider. Automatic Steering Methods for Autonomous Automobile Path Tracking. Carnegie Mellon University, 2009.

Kichun Jo. Integration of Multiple Vehicle Models with an IMM Filter for Vehicle Localization. IEEE Intelligent Vehicles Symposium, 2010.

https://en.wikipedia.org/wiki/Kalman_filter

THANK YOU