# Linearization

October 30, 2025

## 0.1 Linearization

[1]:
```
import sympy as sp
from IPython.display import display

# Enable LaTeX pretty printing
sp.init_printing(use_latex=True)

# Define symbols for lateral and longitudinal dynamics with LaTeX names
y = sp.symbols('y')
y_dot = sp.symbols(r'\dot{y}')
psi = sp.symbols(r'\psi')
psi_dot = sp.symbols(r'\dot{\psi}')
x = sp.symbols('x')
x_dot = sp.symbols(r'\dot{x}')  # longitudinal velocity with LaTeX dot

delta, F = sp.symbols(r'\delta F')  # inputs

# Vehicle parameters
m, Iz, lf, lr, Ca, fm, g = sp.symbols('m Iz l_f l_r C_alpha f g')  # FIXED: use␣
↪C_alpha, not C_\alpha

# Replace vx with x_dot (longitudinal velocity is the second state)
vx = x_dot

# --- Lateral nonlinear dynamics with vx = x_dot ---
y_ddot = -psi_dot * vx + (2*Ca/m) * (sp.cos(delta)*(delta - (y_dot +␣
↪lf*psi_dot)/vx) - (y_dot - lr*psi_dot)/vx)
psi_ddot = (2*lf*Ca/Iz) * (delta - (y_dot + lf*psi_dot)/vx) - (2*lr*Ca/Iz) *␣
↪(-(y_dot - lr*psi_dot)/vx)

# Lateral state and input vectors
s1 = sp.Matrix([y, y_dot, psi, psi_dot])
u = sp.Matrix([delta, F])

# Lateral nonlinear state derivative
f1 = sp.Matrix([y_dot, y_ddot, psi_dot, psi_ddot])
```

```python
# Jacobians lateral
A1 = f1.jacobian(s1)
B1 = f1.jacobian(u)

# Small angle approx cos(delta) ~ 1
A1_lin = A1.subs(sp.cos(delta), 1)
B1_lin = B1.subs(sp.cos(delta), 1)

print("A1 (lateral system matrix):")
display(A1_lin)

print("\nB1 (lateral input matrix):")
display(B1_lin)

# --- Longitudinal nonlinear dynamics ---
# Assume lateral states y, y_dot, psi, psi_dot are constant during longitudinal
↪linearization
# Longitudinal states and inputs
s2 = sp.Matrix([x, x_dot])

# Define disturbance term from lateral coupling and rolling resistance
disturbance = psi_dot * y_dot - (fm * g) / m

# Longitudinal nonlinear state derivative
f2 = sp.Matrix([
    x_dot,
    disturbance + (1/m) * (F)  # no dependence on delta for x_ddot here
])

# Jacobians longitudinal
A2 = f2.jacobian(s2)
B2 = f2.jacobian(u)  # u = [delta, F], delta no effect on x_ddot

print("\nA2 (longitudinal system matrix):")
display(A2)

print("\nB2 (longitudinal input matrix):")
display(B2)

print("\nDisturbance term:")
display(disturbance)
```

```
A1 (lateral system matrix):
```

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{4C_{\alpha}}{\dot{x}m} & 0 & \frac{2C_{\alpha}\left(-\frac{l_f}{\dot{x}}+\frac{l_r}{\dot{x}}\right)}{m}-\dot{x} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{2C_{\alpha}l_f}{Iz\dot{x}}+\frac{2C_{\alpha}l_r}{Iz\dot{x}} & 0 & -\frac{2C_{\alpha}l_f^2}{Iz\dot{x}}-\frac{2C_{\alpha}l_r^2}{Iz\dot{x}} \end{bmatrix}$$

`B1 (lateral input matrix):`

$$\begin{bmatrix} 0 & 0 \\ \frac{2C_{\alpha}\left(-\left(\delta-\frac{\dot{\psi}l_f+\dot{y}}{\dot{x}}\right)\sin(\delta)+1\right)}{m} & 0 \\ 0 & 0 \\ \frac{2C_{\alpha}l_f}{Iz} & 0 \end{bmatrix}$$

`A2 (longitudinal system matrix):`

$$\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

`B2 (longitudinal input matrix):`

$$\begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{m} \end{bmatrix}$$

`Disturbance term:`

$$\dot{\psi}\dot{y}-\frac{fg}{m}$$