Chuan972 

Collaborative Control of Wire-Controlled Chassis Subsystems Based on Game Theory

Overall framework - B-Class, Hatchback

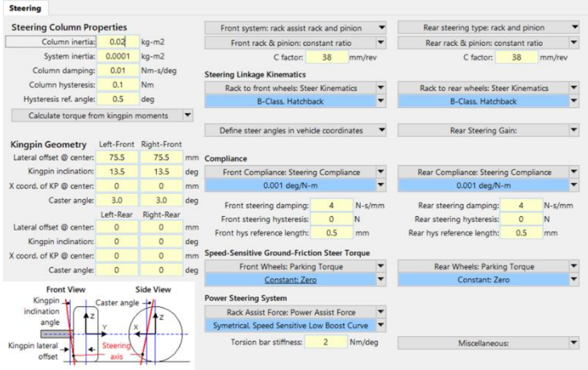
Matlab/Simulink

- Vehicle controller
- Hub motor model

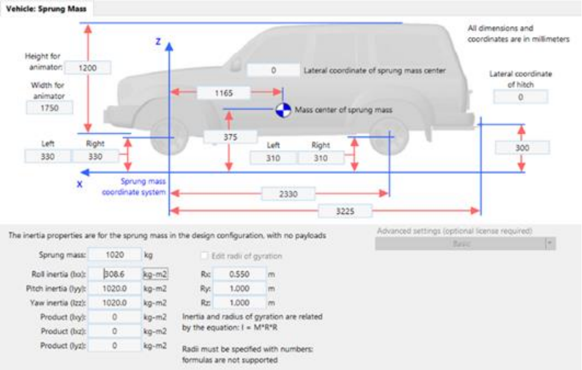
CarSim

- Steering system
- Braking system
- Vehicle body model
- Tire model
- Road surface information

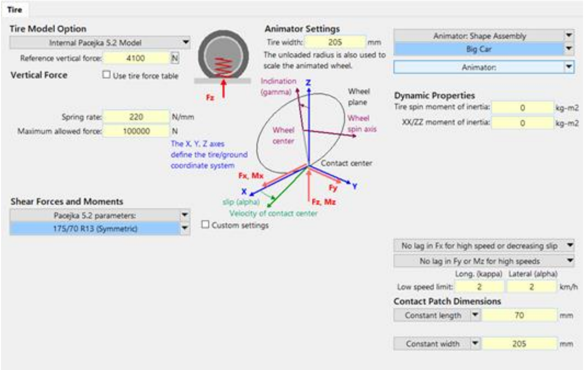
Using built-in components of Carsim



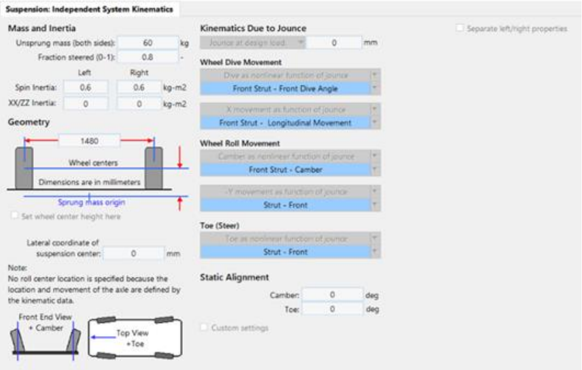
Steering system



Vehicle body model

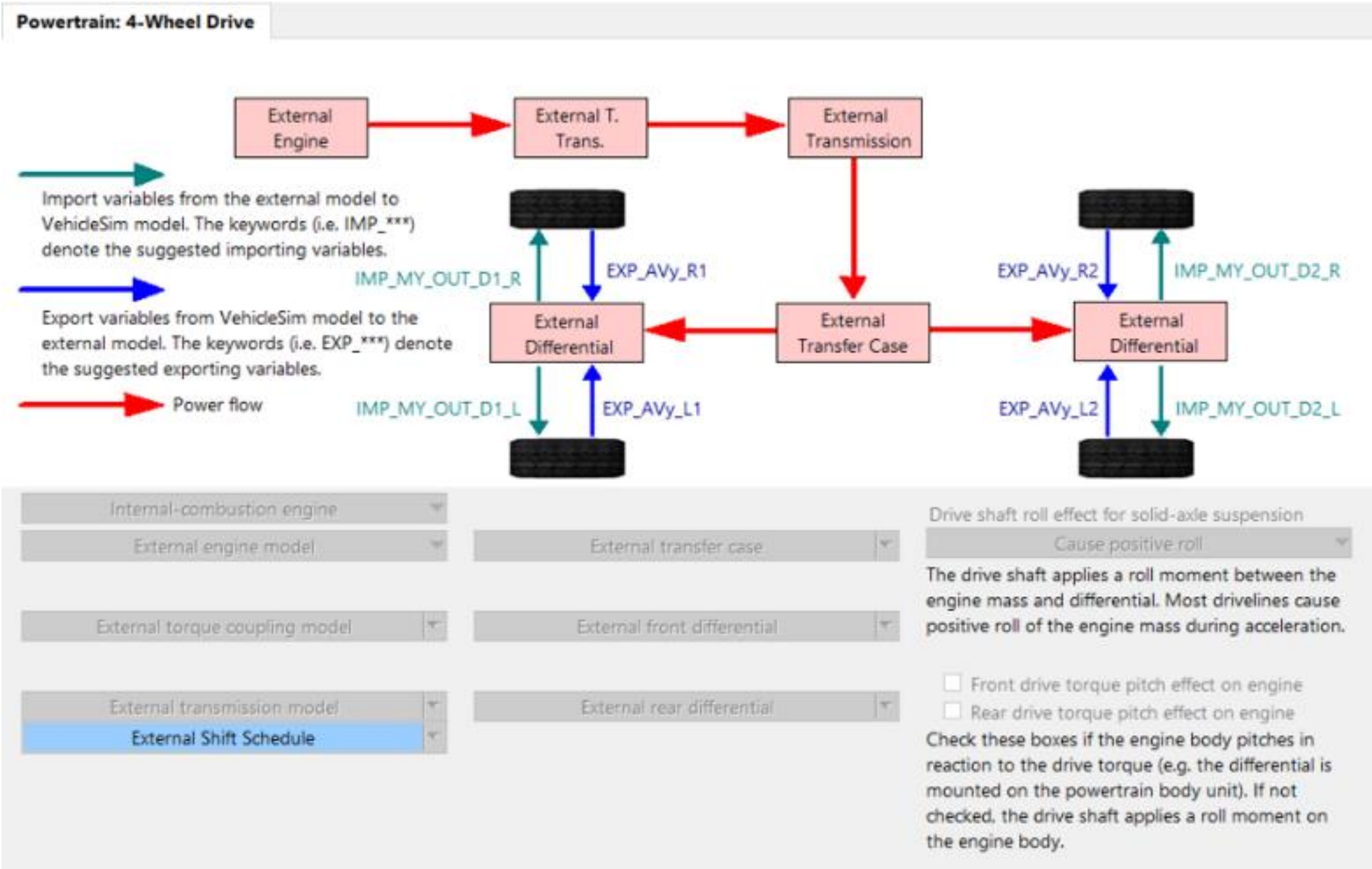


Tire model



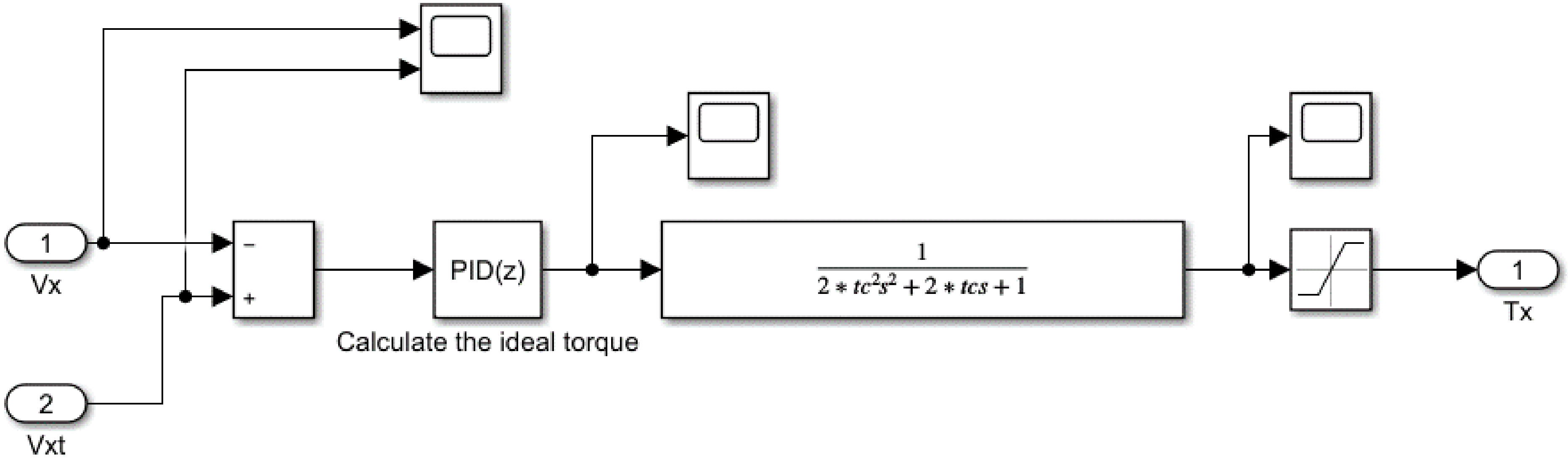
Suspension model

Main modification part - Powertrain system



Electric vehicle (front wheel steering angle and four-wheel torque)

Main Modification Section - Powertrain System



$$K(s)=\frac{T_d}{T_{ld}}=\frac{1}{1+2\xi s+2\xi^2 s^2}$$

$$\xi=0.001$$

Primary inputs and outputs

Interface	Interface Name	Interface Definitions
Input Interface	IMP_MYUSM_L1	Left front wheel input torque
	IMP_MYUSM_L2	Left rear wheel input torque
	IMP_MYUSM_R1	Right front wheel input torque
	IMP_MYUSM_R2	Right rear wheel input torque
	IMP_STEER_L1	Left front wheel input steering angle
	IMP_STEER_R1	Right front wheel input steering angle
Output Interface	Beta-Vehicle slip angle	Vehicle center of mass lateral deviation angle
	Vx-Longitudinal speed	Vehicle longitudinal velocity
	Vy-Lateral speed	Vehicle lateral velocity
	AVz-Yaw rate	Vehicle yaw rate
	AVy_L1-Wheel L1 spin	Left front wheel speed
	AVy_R1-Wheel R1 spin	Right front wheel speed
	Fz_L1	Left front wheel vertical tire force
	Fz_L2	Left rear wheel vertical tire force
	Fz_R1	Right front wheel vertical tire force
	Fz_R2	Right rear wheel vertical tire force

Non-cooperative game theory

- Linear control state equation:

$$\dot{x}(t) = Ax(t) + B_1 u_1(t) + B_2 u_2(t)$$

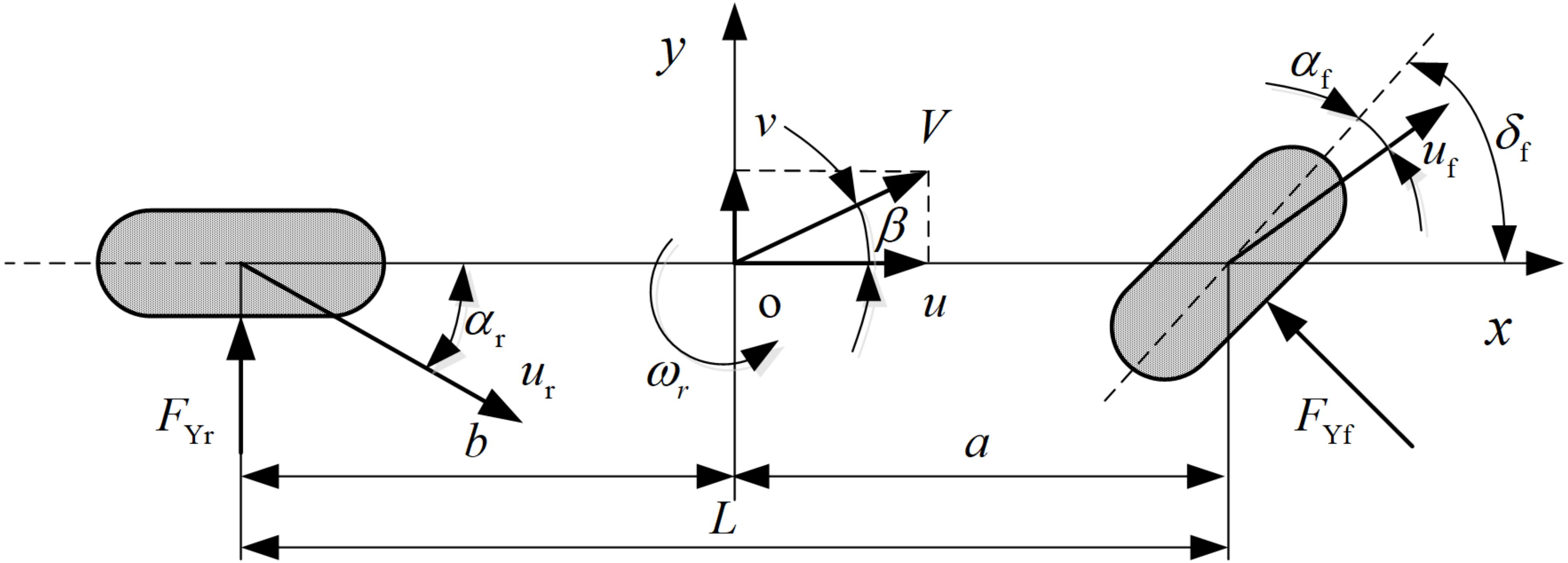
- Cost function:

$$\begin{cases} J_1 = \int_0^T \{x^T(t) Q_1 x(t) + u_1^T(t) R_{11} u_1(t) + u_2^T(t) R_{12} u_2(t)\} dt \\ J_2 = \int_0^T \{x^T(t) Q_2 x(t) + u_1^T(t) R_{21} u_1(t) + u_2^T(t) R_{22} u_2(t)\} dt \end{cases}$$

- Nash equilibrium control strategy:

$$\begin{cases} J_1(u_{F1}, u_{F2}) \leq J_2(u_1, u_{F2}) \\ J_2(u_{F1}, u_{F2}) \leq J_2(u_{F1}, u_2) \end{cases}$$

Upper-level controller design - Two-degree-of-freedom model



Upper-level controller design - Two-degree-of-freedom model

- Differential equations for the two-degree-of-freedom model:

$$
\begin{cases}
(k_f + k_r)\beta + \dfrac{1}{V_x}(ak_f - bk_r)\omega - k_f\delta_f = m(\dot{V}_y + V_x\omega) \\
(ak_f - bk_r)\beta + \dfrac{1}{V_x}(a^2k_f - b^2k_r)\omega - ak_f\delta_f = I_z\dot{\omega}
\end{cases}
$$

- State-space equations for the two-degree-of-freedom model:

$$
\begin{bmatrix} \dot{\beta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} -\dfrac{k_f + k_r}{mV_x} & -\dfrac{ak_f - bk_r}{m{V_x}^2} - 1 \\ -\dfrac{ak_f - bk_r}{I_z} & -\dfrac{a^2k_f + b^2k_r}{I_zV_x} \end{bmatrix} \begin{bmatrix} \beta \\ \omega \end{bmatrix} + \begin{bmatrix} \dfrac{k_f}{mV_x} \\ \dfrac{ak_f}{I_z} \end{bmatrix} \delta_f
$$

Upper-level controller design - Two-degree-of-freedom model

- For the sake of convenience in research, redefine the state-space equations:

$$\begin{bmatrix}\Delta\dot{\beta}\\ \Delta\dot{\omega}\end{bmatrix}=\begin{bmatrix}-\dfrac{k_f+k_r}{mV_x} & -\dfrac{ak_f-bk_r}{mV_x^{\,2}}-1\\ -\dfrac{ak_f-bk_r}{I_z} & -\dfrac{a^2k_f+b^2k_r}{I_zV_x}\end{bmatrix}\boxed{\begin{bmatrix}\Delta\beta\\ \Delta\omega\end{bmatrix}}+\begin{bmatrix}\dfrac{k_f}{mV_x}\\ \dfrac{ak_f}{I_z}\end{bmatrix}\Delta\delta_f+\begin{bmatrix}0\\ \dfrac{1}{I_z}\end{bmatrix}\Delta M$$

$$\begin{cases}\Delta\beta=\beta-\boxed{\beta_{des}}\\ \Delta\omega=\omega-\boxed{\omega_{des}}\end{cases}$$

$$\begin{cases}\beta_{des}=\dfrac{b-\dfrac{amV_x^{\,2}}{k_r(a+b)}}{(a+b)(1+K_{us}V_x^{\,2})}\delta_f\\ \omega_{des}=\dfrac{V_x}{(a+b)(1+K_{us}V_x^{\,2})}\delta_f\\ K_{us}=\dfrac{m(bk_2-ak_1)}{(a+b)^2k_fk_r}\end{cases}$$

Upper-level controller design - Two-degree-of-freedom model

- For ease of computation, simplify the state-space equations:

$$\begin{bmatrix} \Delta\dot{\beta} \\ \Delta\dot{\omega} \end{bmatrix} = \begin{bmatrix} -\dfrac{k_f + k_r}{mV_x} & -\dfrac{ak_f - bk_r}{mV_x{}^2} - 1 \\ -\dfrac{ak_f - bk_r}{I_z} & -\dfrac{a^2k_f + b^2k_r}{I_zV_x} \end{bmatrix} \begin{bmatrix} \Delta\beta \\ \Delta\omega \end{bmatrix} + \begin{bmatrix} \dfrac{k_f}{mV_x} \\ \dfrac{ak_f}{I_z} \end{bmatrix} \Delta\delta_f + \begin{bmatrix} 0 \\ \dfrac{1}{I_z} \end{bmatrix} \Delta M$$

$$\dot{x} = Ax_n + B_1u_1 + B_2u_2$$

Upper-level controller design - Cost function

- Define the game model based on the cost function:

$$\begin{cases} J_1(u_1,u_2)=\int_0^{\infty}\frac{1}{2}\{x_n^T(t)Q_1x_n(t)+u_1^T(t)R_{11}u_1(t)+u_2^T(t)R_{12}u_2(t)\}dt \\ J_2(u_1,u_2)=\int_0^{\infty}\frac{1}{2}\{x_n^T(t)Q_2x_n(t)+u_1^T(t)R_{21}u_1(t)+u_2^T(t)R_{22}u_2(t)\}dt \end{cases}$$

- The cost function satisfies the following conditions:

$$\begin{cases} J_1(u_1^*,u_2^*)\le J_1(u_1,u_2^*) \\ J_2(u_1^*,u_2^*)\le J_2(u_1^*,u_2) \end{cases}$$

Upper-level controller design - Hamiltonian equation

- Establish the Hamiltonian equation

$$\begin{cases} H_1 = \dfrac{1}{2}(x_n^T Q_1 x_n + u_1^T R_{11} u_1 + u_2^T R_{12} u_2) + \lambda_1^T (A x_n + B_1 u_1 + B_2 u_2) \\ H_2 = \dfrac{1}{2}(x_n^T Q_2 x_n + u_1^T R_{21} u_1 + u_2^T R_{22} u_2) + \lambda_2^T (A x_n + B_1 u_1 + B_2 u_2) \end{cases}$$

- Construct the extremum condition equation

$$\begin{cases} \dfrac{\partial H_1}{\partial x_n} = R_{11} u_1 + B_1^T \lambda_1 = 0 \\ \dfrac{\partial H_2}{\partial x_n} = R_{22} u_2 + B_2^T \lambda_2 = 0 \end{cases} \xrightarrow{\text{solve}} \begin{cases} u_1 = -R_{11}^{-1} B_1^T \lambda_1 \\ u_2 = -R_{22}^{-1} B_2^T \lambda_2 \end{cases}$$

Upper-level controller design - Hamiltonian equation

- Substitute the simplified state-space equations

$$\begin{cases} u_1 = -R_{11}^{-1} B_1^T \lambda_1 \\ u_2 = -R_{22}^{-1} B_2^T \lambda_2 \end{cases} \longrightarrow \dot{x} = Ax_n + B_1 u_1 + B_2 u_2 \longrightarrow \dot{x}_n = Ax_n - B_1 R_{11}^{-1} B_1^T \boxed{\lambda_1} - B_2 R_{22}^{-1} B_2^T \boxed{\lambda_2}$$

- According to the optimization control conditions

$$\begin{cases} \dot{\lambda}_1 = -\dfrac{\partial H_1}{\partial x_n} = -A^T \lambda_1 - Q_1 x_n \\ \dot{\lambda}_2 = -\dfrac{\partial H_2}{\partial x_n} = -A^T \lambda_2 - Q_2 x_n \end{cases}$$

- Write it in terms of the co-state matrix

$$\begin{bmatrix} \dot{x}_n \\ \dot{\lambda}_1 \\ \dot{\lambda}_2 \end{bmatrix} = \begin{bmatrix} A & -B_1 R_{11}^{-1} B_1^T & -B_2 R_{22}^{-1} B_2^T \\ -Q1 & -A^T & 0 \\ -Q2 & 0 & -A^T \end{bmatrix} \cdot \begin{bmatrix} x_n \\ \lambda_1 \\ \lambda_2 \end{bmatrix}$$

Upper-level controller design - Hamiltonian equation

- There is a relationship between state variables

$$\begin{bmatrix} \dot{x}_n \\ \dot{\lambda}_1 \\ \dot{\lambda}_2 \end{bmatrix} = \begin{bmatrix} A & -B_1 R_{11}{}^{-1} B_1{}^T & -B_2 R_{22}{}^{-1} B_2{}^T \\ -Q1 & -A^T & 0 \\ -Q2 & 0 & -A^T \end{bmatrix} \cdot \boxed{\begin{bmatrix} x_n \\ \lambda_1 \\ \lambda_2 \end{bmatrix}} \longrightarrow \begin{cases} \lambda_1 = G_1 x_n \\ \lambda_2 = G_2 x_n \end{cases}$$

- Substitute into the extremum condition equation for a solution

$$\begin{cases} u_1 = -R_{11}{}^{-1} B_1{}^T \lambda_1 \\ u_2 = -R_{22}{}^{-1} B_2{}^T \lambda_2 \end{cases} \longrightarrow \begin{cases} u_1{}^* = -R_{11}{}^{-1} B_1{}^T G_1 x_n \\ u_2{}^* = -R_{22}{}^{-1} B_2{}^T G_2 x_n \end{cases}$$

Upper-level controller design - Riccati equation

- G_1 and G_2 are solutions to the following coupled Riccati equations

$$\begin{cases} (A-T_2G_2)^T G_1 + G_1(A-T_2G_2) - G_1T_1G_1 + Q_1 + G_2T_{12}G_2 = 0 \\ (A-T_1G_1)^T G_2 + G_2(A-T_1G_1) - G_2T_2G_2 + Q_2 + G_1T_{21}G_1 = 0 \end{cases}$$

- Expressions for T_{ij} and T_i are as follows

$$\begin{cases} T_i = B_i R_{ii}^{-1} B_i^T & i = 1,2 \\ T_{ij} = B_i R_{ii}^{-1} R_{ji} R_{ii}^{-1} B_i^T & i, j = 1,2 \,\&\, i \neq j \end{cases}$$

The game process is specifically reflected in the process of solving the coupled Riccati equations.

Upper-level controller design - Riccati equation

- Discretize and reconstruct the equations as follows:

$$\begin{cases}(A-T_2G_2)^T G_1 + G_1(A-T_2G_2) - G_1T_1G_1 + Q_1 + G_2T_{12}G_2 = 0 \\ (A-T_1G_1)^T G_2 + G_2(A-T_1G_1) - G_2T_2G_2 + Q_2 + G_1T_{21}G_1 = 0\end{cases}$$

$$\downarrow$$

$$\begin{cases}(A-T_2{G_2}^c)^T {G_1}^{c+1} + {G_1}^{c+1}(A-T_2{G_2}^c) - {G_1}^{c+1}T_1{G_1}^{c+1} + Q_1 + {G_2}^cT_{12}{G_2}^c = 0 \\ (A-T_1{G_1}^c)^T {G_2}^{c+1} + {G_2}^{c+1}(A-T_1{G_1}^c) - {G_2}^{c+1}T_2{G_2}^{c+1} + Q_2 + {G_1}^cT_{21}{G_1}^c = 0\end{cases}$$

- Obtain the initial solutions for the decoupled Riccati equations through the following equations:

$$\begin{cases}A^T {G_1}^0 + {G_1}^0 A + Q_1 + {G_1}^0 T_{11} {G_1}^0 = 0 \\ A^T {G_2}^0 + {G_2}^0 A + Q_2 + {G_2}^0 T_{11} {G_2}^0 = 0\end{cases}$$

Upper-level controller design - Riccati equation

- Treating G^0 as a constant

$$\begin{cases} \boxed{(A - T_2 G_2^{\,c})^T} G_1^{\,c+1} + G_1^{\,c+1} \boxed{(A - T_2 G_2^{\,c})} - G_1^{\,c+1} T_1 G_1^{\,c+1} \boxed{+ Q_1 + G_2^{\,c} T_{12} G_2^{\,c}} = 0 \\ \boxed{(A - T_1 G_1^{\,c})^T} G_2^{\,c+1} + G_2^{\,c+1} \boxed{(A - T_1 G_1^{\,c})} - G_2^{\,c+1} T_2 G_2^{\,c+1} \boxed{+ Q_2 + G_1^{\,c} T_{21} G_1^{\,c}} = 0 \end{cases}$$

$$\begin{cases} A_1^T = (A - T_2 G_2^{\,c})^T \\ A_2^T = (A - T_1 G_1^{\,c})^T \end{cases} \qquad \begin{cases} Q_1' = Q_1 + G_2^{\,c} T_{12} G_2^{\,c} \\ Q_2' = Q_2 + G_1^{\,c} T_{21} G_1^{\,c} \end{cases}$$

- Transform the equation into the same form as the strong solution

$$\begin{cases} A_1^T G_1^{\,c+1} + G_1^{\,c+1} A_1 + Q_1' + G_1^{\,c+1} T_{11} G_1^{\,c+1} = 0 \\ A_2^T G_2^{\,c+1} + G_2^{\,c+1} A_2 + Q_2' + G_2^{\,c+1} T_{22} G_2^{\,c+1} = 0 \end{cases}$$

Upper-level controller design - Riccati equation

- Iteration termination condition:

$$\begin{cases} \left\| G_1^{\,c+1} - G_1^{\,c} \right\| < 10^{-4} \\ \left\| G_2^{\,c+1} - G_2^{\,c} \right\| < 10^{-4} \end{cases}$$

- Computed results:

$$\begin{cases} u_1^{\,*} = -R_{11}^{\,-1} B_1^{\,T} G_1^{\,c+1} x_n \\ u_2^{\,*} = -R_{22}^{\,-1} B_2^{\,T} G_2^{\,c+1} x_n \end{cases}$$

- Corresponding results:

$$\begin{cases} \Delta\delta_f = u_1^{\,*} \\ \Delta M = u_2^{\,*} \end{cases}$$

Lower-level controller design - Quadratic programming

$$
\begin{cases}
\min \varepsilon = \sum_j \dfrac{{T_{xj}}^2}{(\mu F_{zj})^2 R^2} \\
s.t. \quad T_x = \sum_j T_{xj} \\
\qquad \Delta M = \dfrac{t_r}{2}\left(\dfrac{T_{xfr}}{R} - \dfrac{T_{xfl}}{R}\right) + \dfrac{t_r}{2}\left(\dfrac{T_{xrl}}{R} - \dfrac{T_{xrr}}{R}\right) \\
\qquad 0 \le T_{xj} \le \min(T_{m\_\max}, \mu F_{zj} R)
\end{cases}
$$

Lower-level controller design - Quadratic programming

$$
\begin{cases}
\boxed{\min \varepsilon = \sum_j \dfrac{T_{xj}{}^2}{(\mu F_{zj})^2 R^2}} \\
s.t. \quad T_x = \sum_j T_{xj} \\
\qquad \Delta M = \dfrac{t_r}{2}\left(\dfrac{T_{xfr}}{R} - \dfrac{T_{xfl}}{R}\right) + \dfrac{t_r}{2}\left(\dfrac{T_{xrl}}{R} - \dfrac{T_{xrr}}{R}\right) \\
\qquad 0 \le T_{xj} \le \min(T_{m\_\max}, \mu F_{zj} R)
\end{cases}
$$

With the objective of minimizing tire utilization

Lower-level controller design - Quadratic programming

$$\begin{cases} \min \varepsilon = \sum_j \dfrac{T_{xj}^{\ 2}}{(\mu F_{zj})^2 R^2} \\ s.t. \quad \boxed{T_x = \sum_j T_{xj}} \\ \qquad \Delta M = \dfrac{t_r}{2}\left(\dfrac{T_{xfr}}{R} - \dfrac{T_{xfl}}{R}\right) + \dfrac{t_r}{2}\left(\dfrac{T_{xrl}}{R} - \dfrac{T_{xrr}}{R}\right) \\ \qquad 0 \le T_{xj} \le \min(T_{m\_\max}, \mu F_{zj} R) \end{cases}$$

Vehicle propulsion total torque =
Sum of torques from all four drive wheels

Lower-level controller design - Quadratic programming

$$
\begin{cases}
\min \varepsilon = \sum_j \dfrac{{T_{xj}}^2}{(\mu F_{zj})^2 R^2} \\
s.t. \quad T_x = \sum_j T_{xj} \\
\qquad \boxed{\Delta M = \dfrac{t_r}{2}\left(\dfrac{T_{xfr}}{R} - \dfrac{T_{xfl}}{R}\right) + \dfrac{t_r}{2}\left(\dfrac{T_{xrl}}{R} - \dfrac{T_{xrr}}{R}\right)} \\
\qquad 0 \le T_{xj} \le \min(T_{m\_\max}, \mu F_{zj} R)
\end{cases}
$$

Additional total torque should also satisfy allocation conditions

Lower-level controller design - Quadratic programming

$$
\begin{cases}
\min \varepsilon = \sum_j \dfrac{{T_{xj}}^2}{(\mu F_{zj})^2 R^2} \\
s.t. \quad T_x = \sum_j T_{xj} \\
\qquad \Delta M = \dfrac{t_r}{2}\left(\dfrac{T_{xfr}}{R} - \dfrac{T_{xfl}}{R}\right) + \dfrac{t_r}{2}\left(\dfrac{T_{xrl}}{R} - \dfrac{T_{xrr}}{R}\right) \\
\qquad \boxed{0 \le T_{xj} \le \min(T_{m\_\max}, \mu F_{zj} R)}
\end{cases}
$$

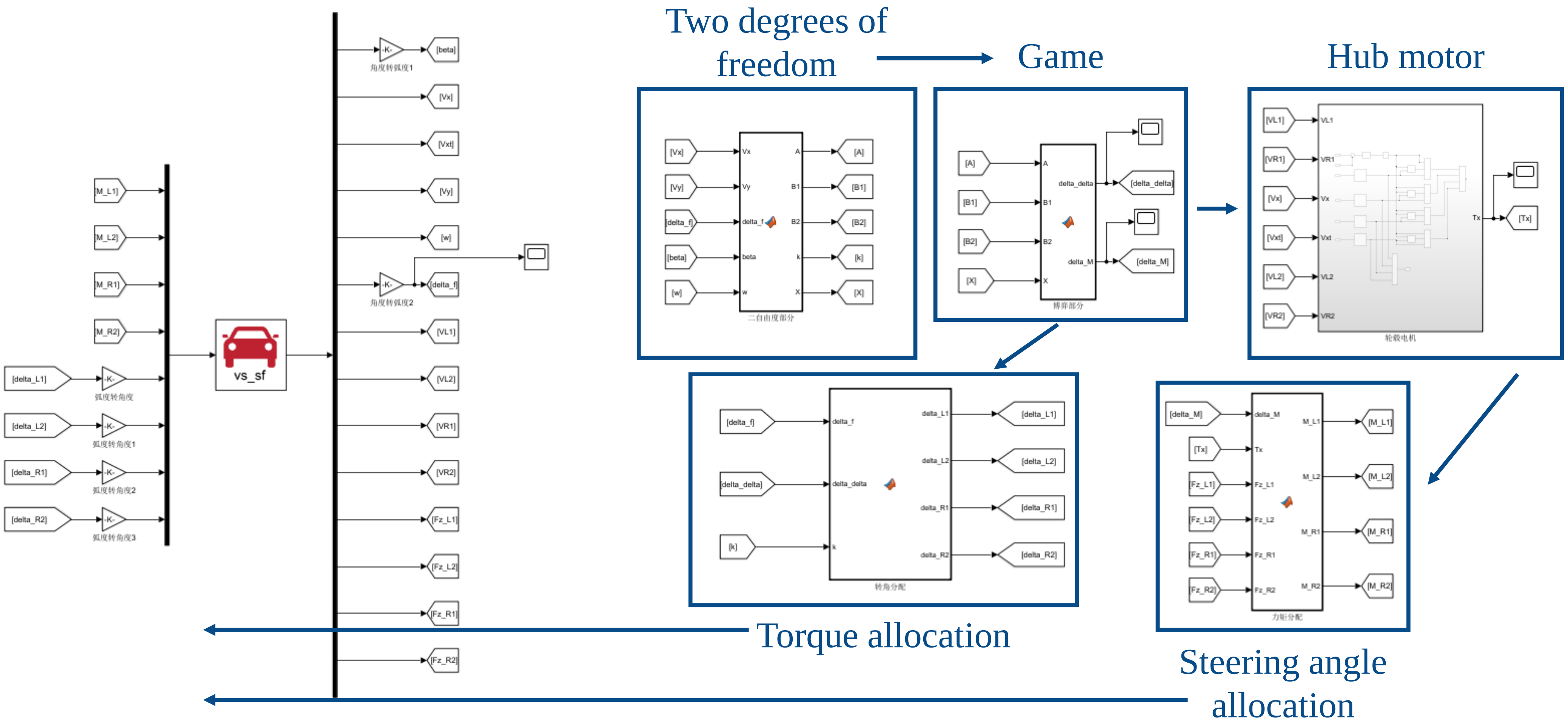
Subject to peak torque limits

Lower-level controller design - Quadratic programming

$$
\begin{cases}
\min \varepsilon = \sum_j \dfrac{{T_{xj}}^2}{(\mu F_{zj})^2 R^2} \\
s.t. \quad T_x = \sum_j T_{xj} \\
\qquad \Delta M = \dfrac{t_r}{2}\left(\dfrac{T_{xfr}}{R} - \dfrac{T_{xfl}}{R}\right) + \dfrac{t_r}{2}\left(\dfrac{T_{xrl}}{R} - \dfrac{T_{xrr}}{R}\right) \\
\qquad 0 \le T_{xj} \le \min(T_{m\_\max}, \mu F_{zj} R)
\end{cases}
$$

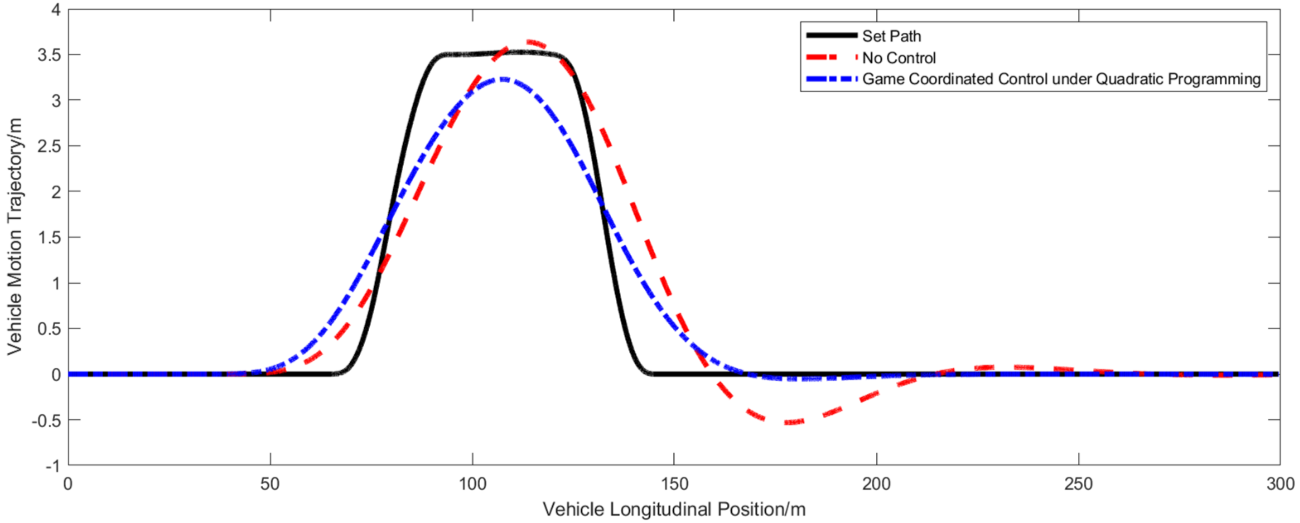
- A typical quadratic programming problem.
- Solve it using MATLAB toolbox.

Building a Simulink model

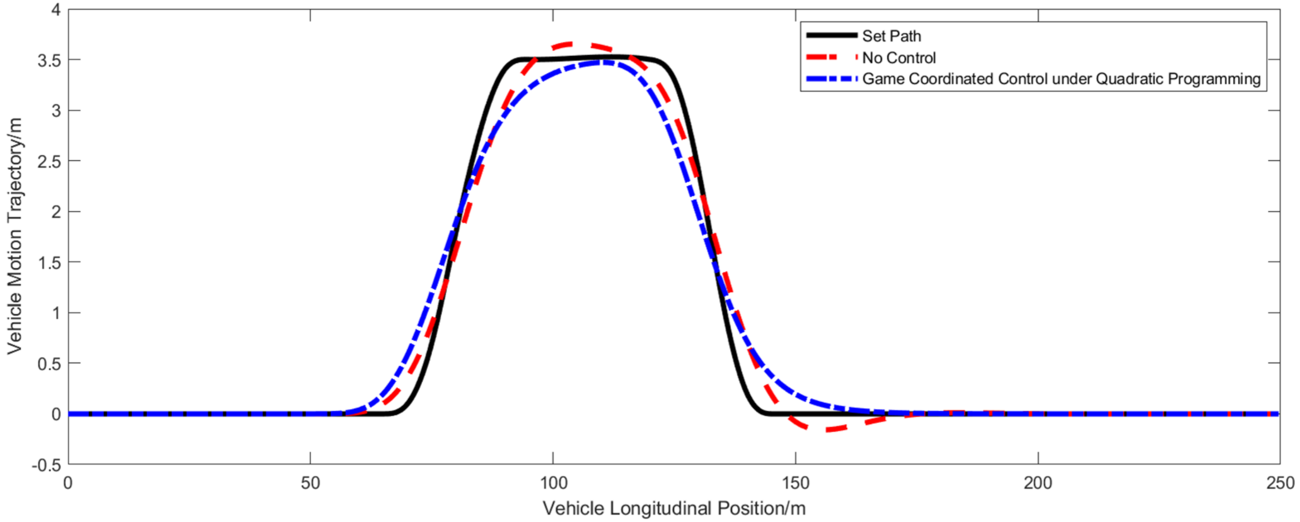


Vehicle trajectory tracking performance

- Vehicle speed: 120 km/h
- Road surface adhesion coefficient: 0.85

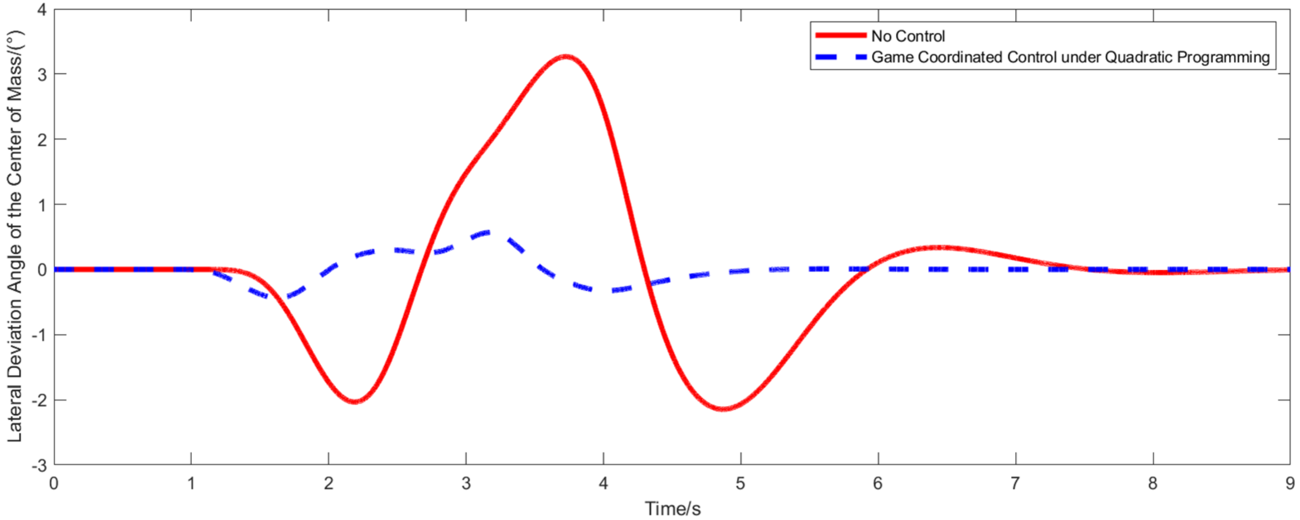


- Vehicle speed: 60 km/h
- Road surface adhesion coefficient: 0.3

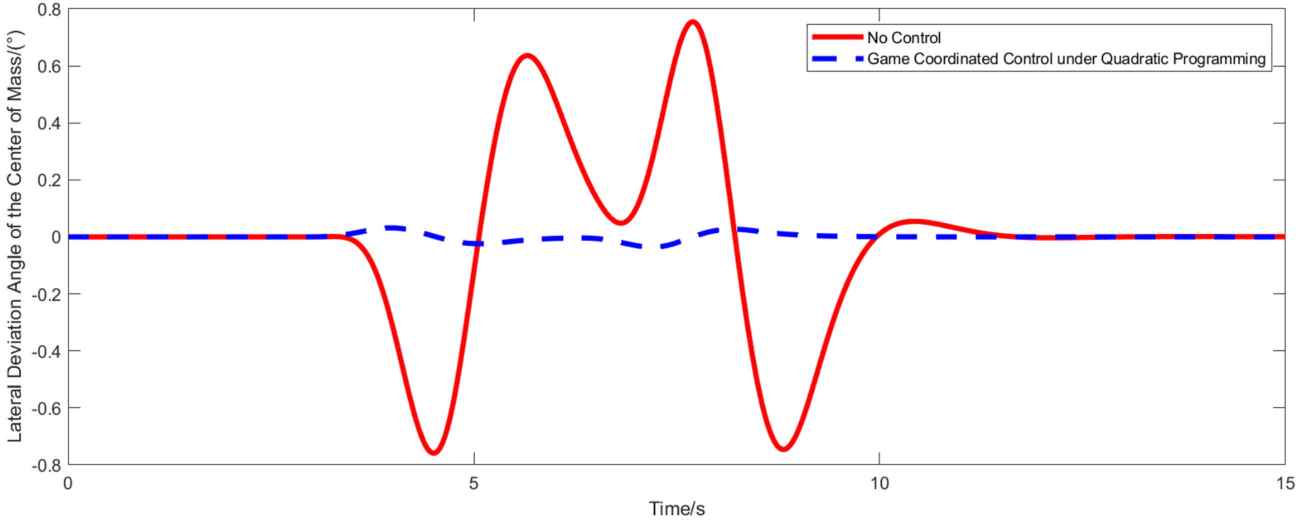


Simulation results for lateral deviation angle of the center of mass

- Vehicle speed: 120 km/h
- Road surface adhesion coefficient: 0.85

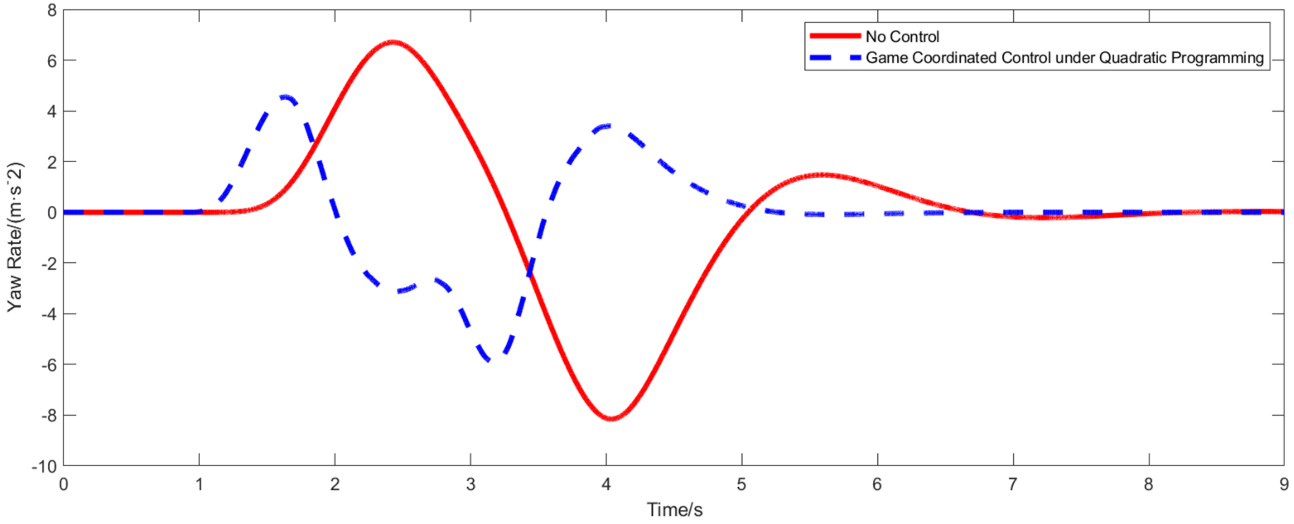


- Vehicle speed: 60 km/h
- Road surface adhesion coefficient: 0.3

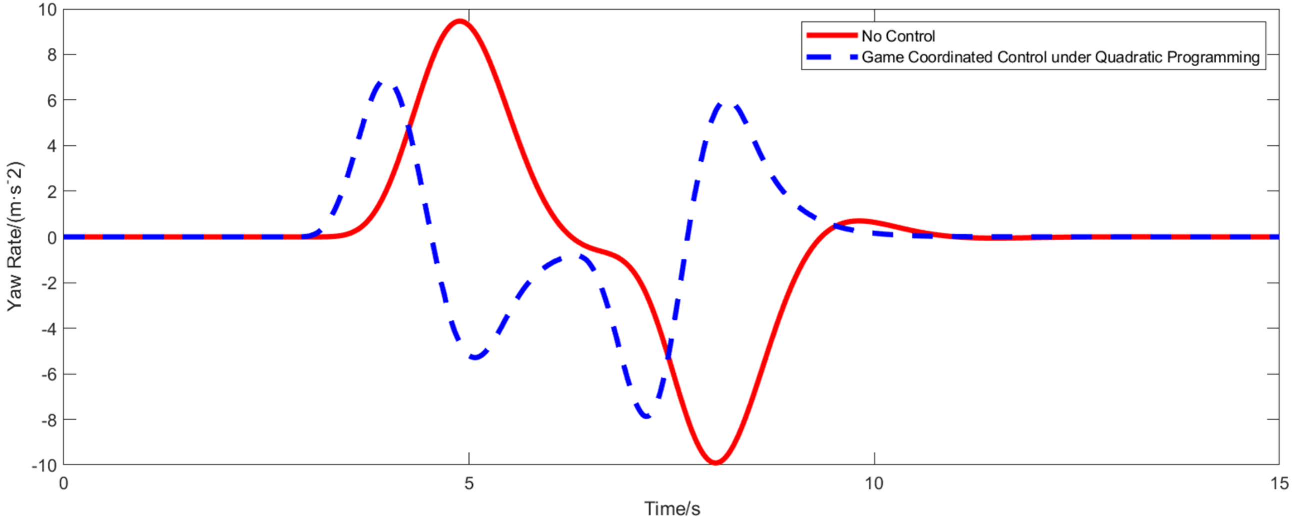


Simulation results for yaw rate

- Vehicle speed: 120 km/h
- Road surface adhesion coefficient: 0.85

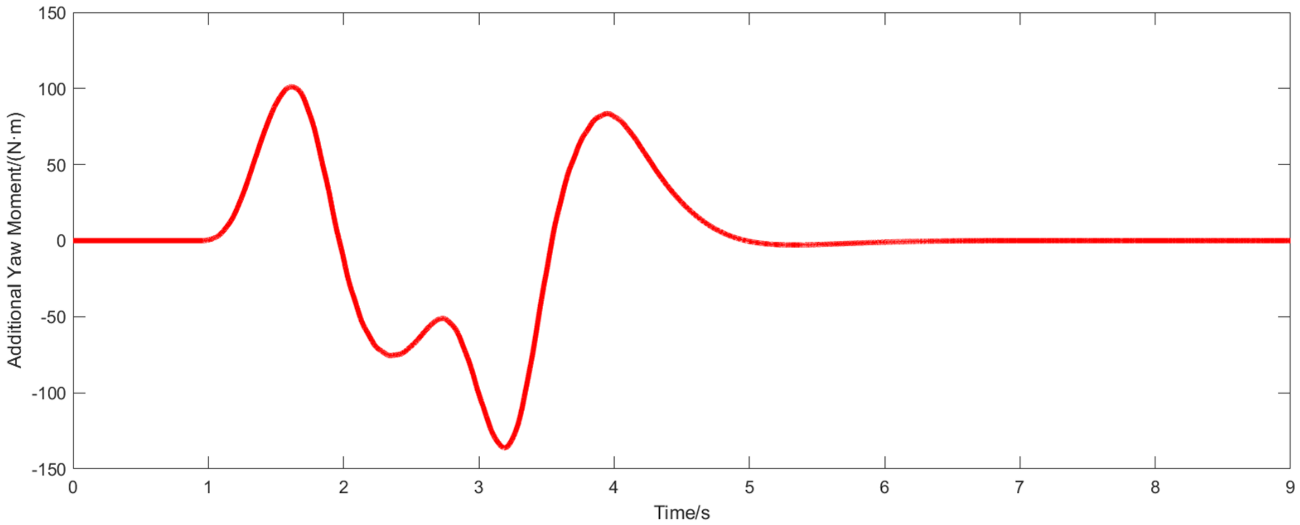


- Vehicle speed: 60 km/h
- Road surface adhesion coefficient: 0.3

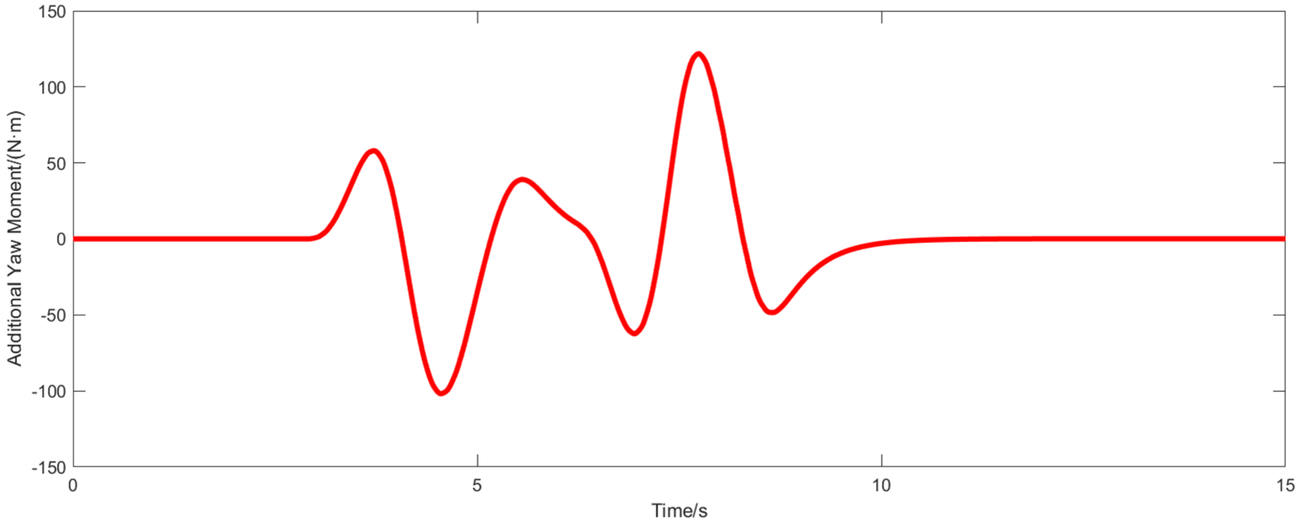


Simulation results for additional yaw moment

- Vehicle speed: 120 km/h
- Road surface adhesion coefficient: 0.85

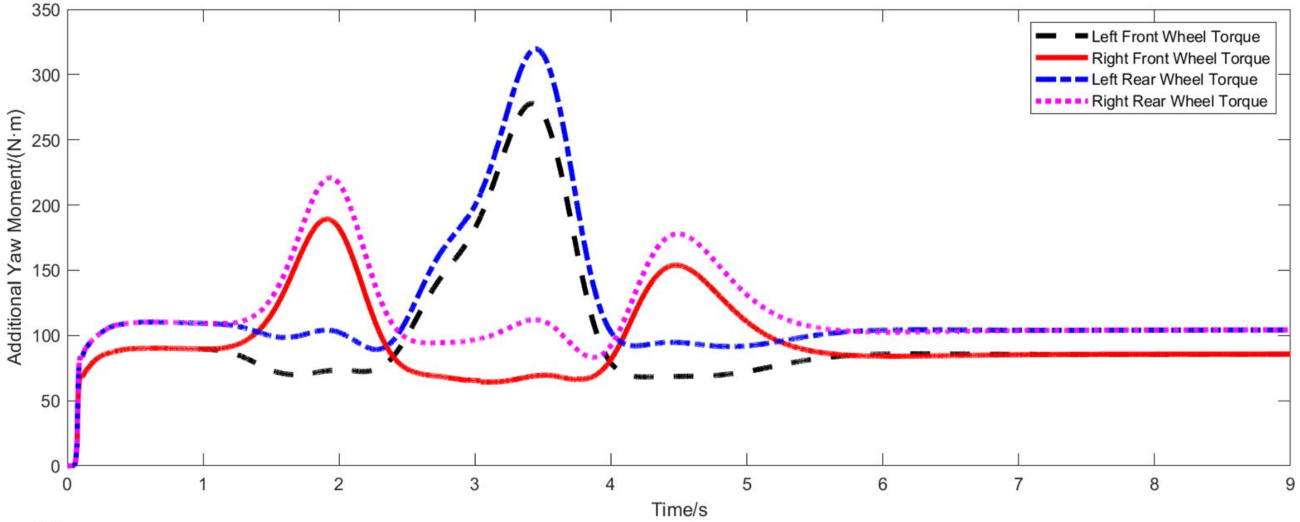


- Vehicle speed: 60 km/h
- Road surface adhesion coefficient: 0.3

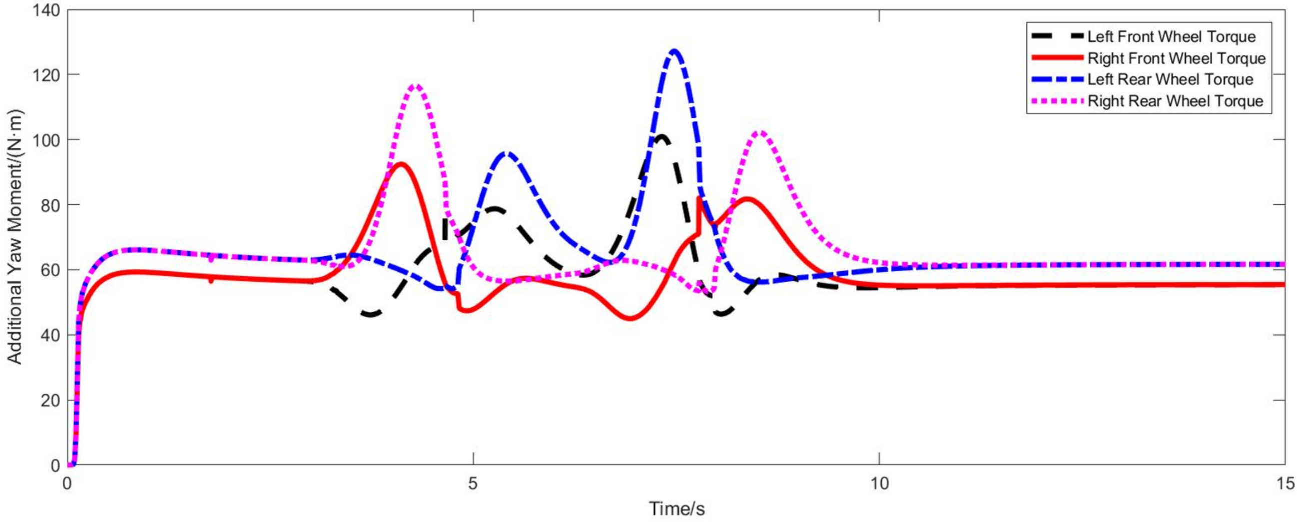


Simulation results for four-wheel torque

- Vehicle speed: 120 km/h
- Road surface adhesion coefficient: 0.85

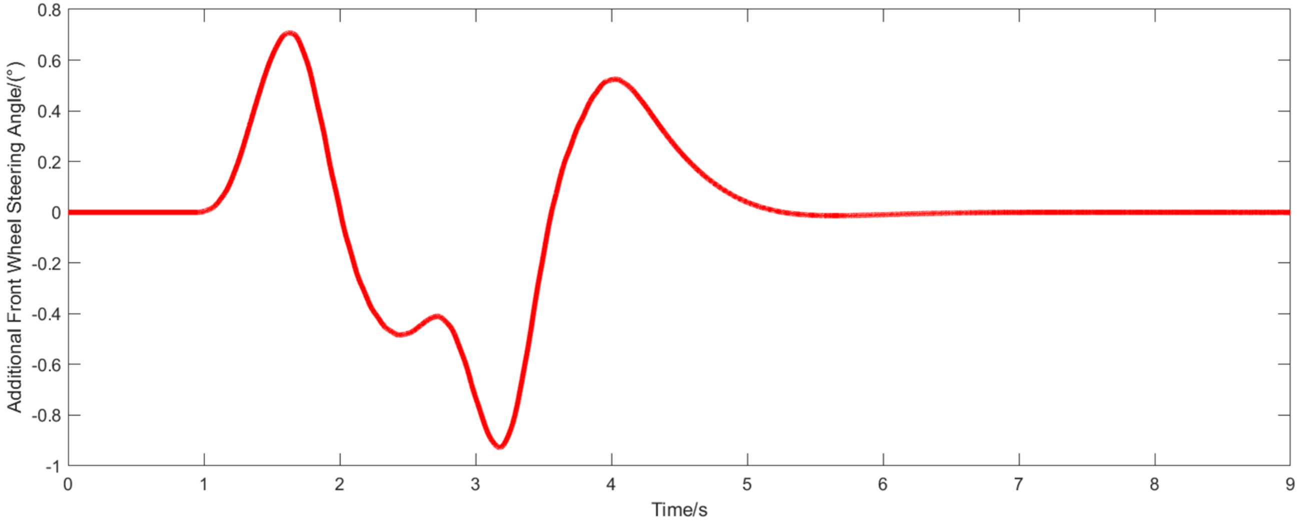


- Vehicle speed: 60 km/h
- Road surface adhesion coefficient: 0.3

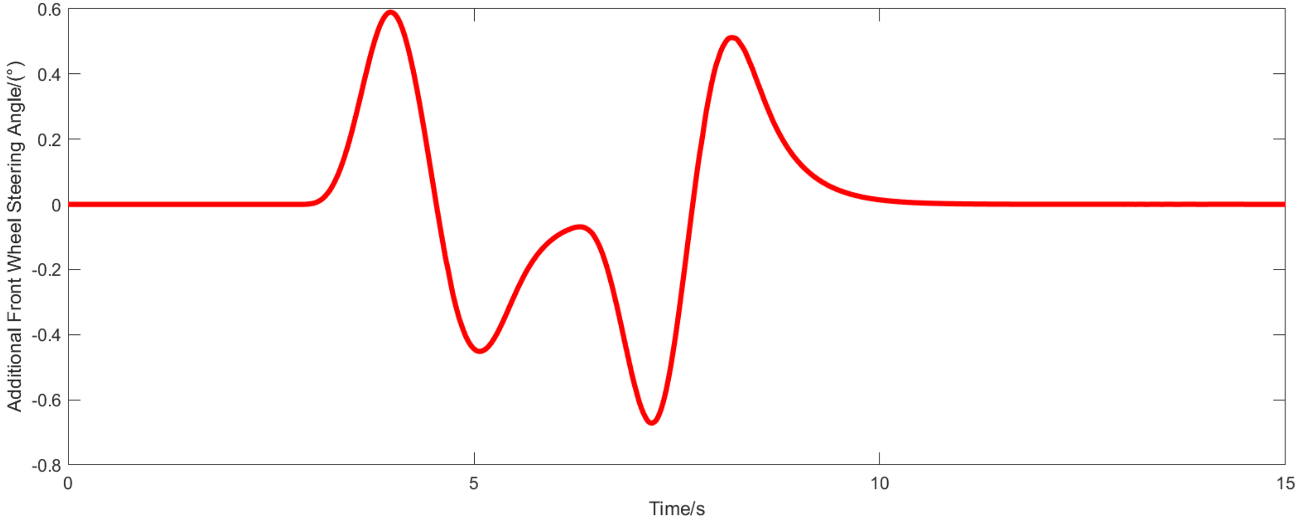


Simulation results for additional front wheel steering angle

- Vehicle speed: 120 km/h
- Road surface adhesion coefficient: 0.85

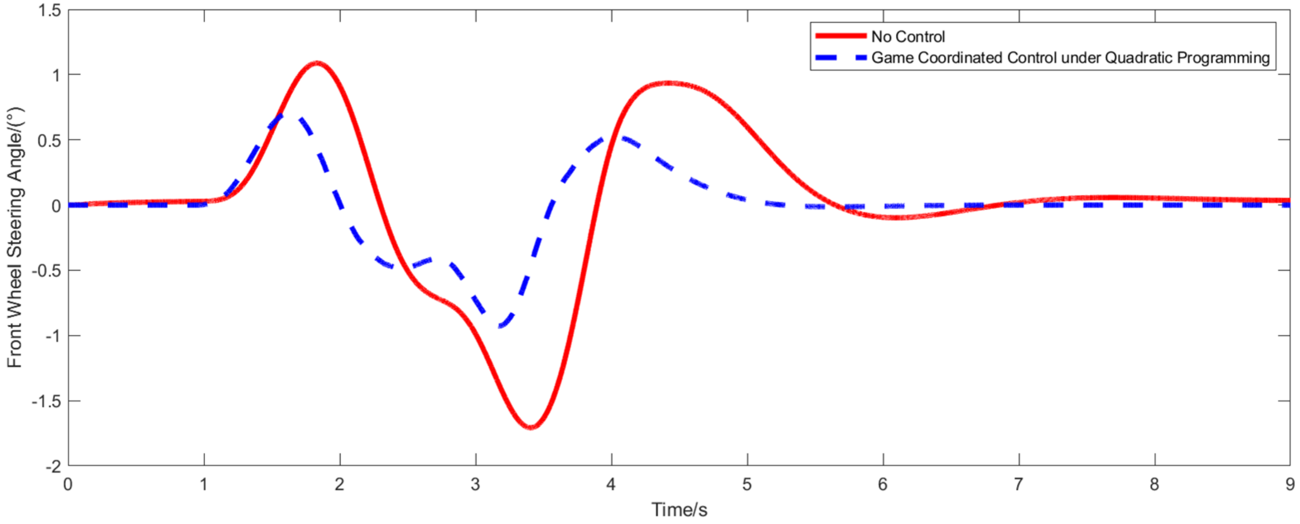


- Vehicle speed: 60 km/h
- Road surface adhesion coefficient: 0.3



Simulation results for front wheel steering angle

- Vehicle speed: 120 km/h
- Road surface adhesion coefficient: 0.85



- Vehicle speed: 60 km/h
- Road surface adhesion coefficient: 0.3