Autonomous Vehicle Planning and Control

Wu Ning

Session 4

Vehicle Lateral Optimal Control

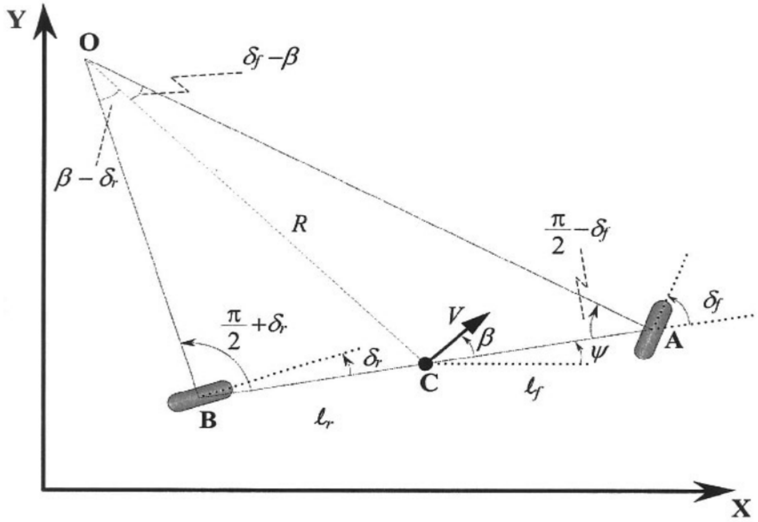
Outline

1. Lateral dynamic model
   a. Dynamic bicycle model
   b. Linearized dynamic bicycle model
   c. Model parameters identification 模型参数识别
2. Linear quadratic regulator (LQR) review
   a. What is LQR?
   b. 1D Scalar example
   c. General solution and Raccati Equation
   d. LQR tuning case study
   调优
3. Vehicle lateral optimal control
   a. Path coordinate model
   b. Trajectory tracking with LQR
   c. Trajectory tracking with Preview control

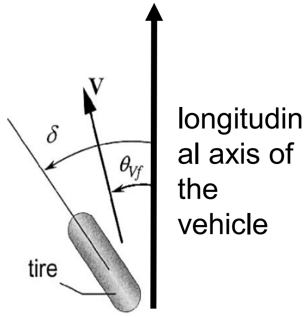
Lateral Dynamic Bicycle Model

Why do we need vehicle dynamic model?

Higher vehicle speeds, the assumption that the velocity at each wheel is in the direction of the wheel can no longer be made, i.e. **non-zero slip angle.**

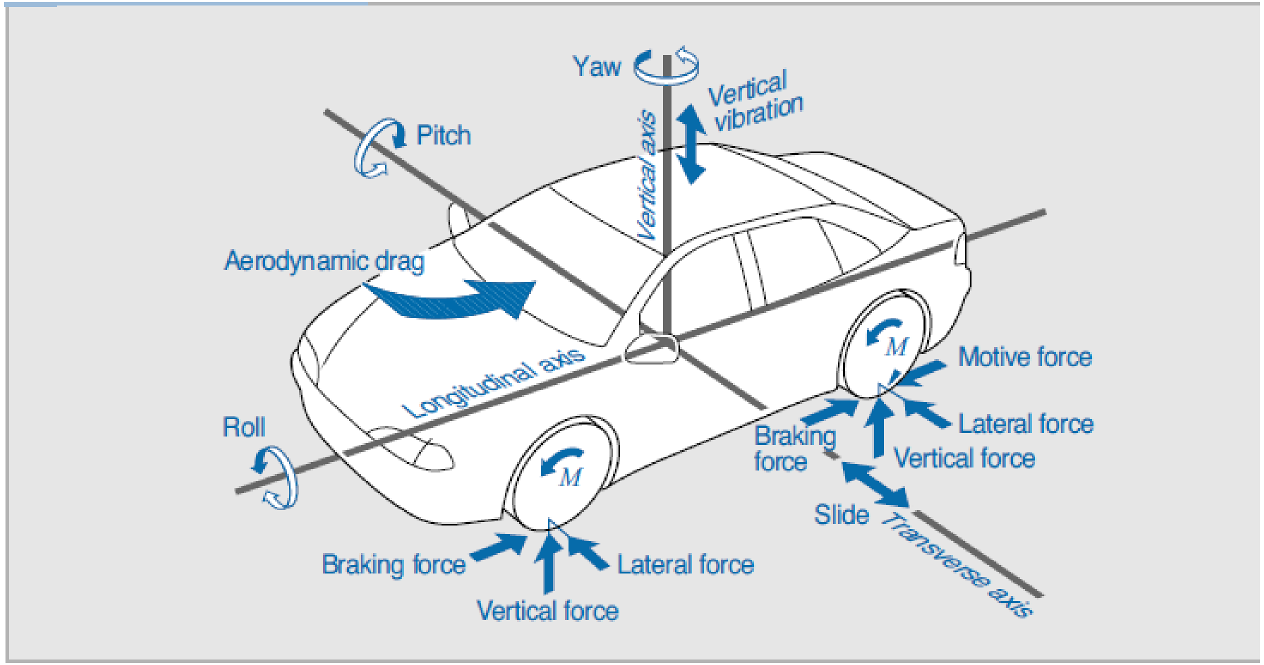


Kinematics Bicycle Model



Tire slip-angle

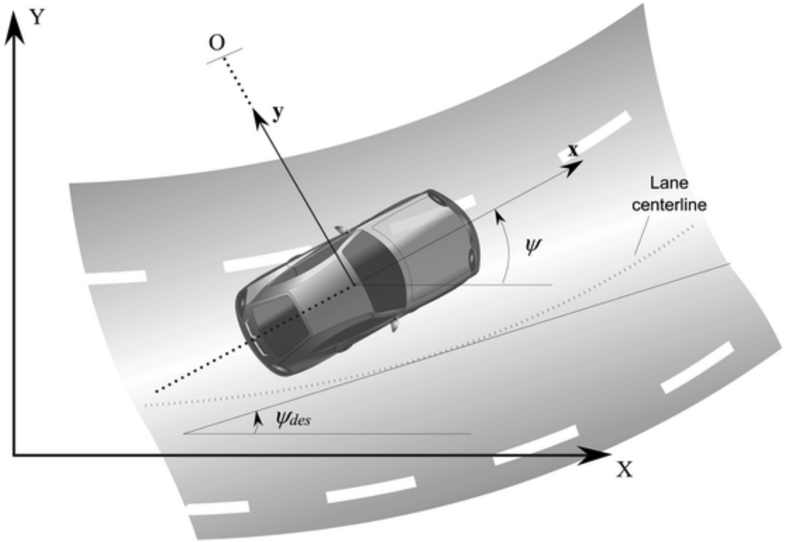
Vehicle Dynamic Model



Dynamic Model of Lateral Vehicle Motion

Dynamic Bicycle Model Assumptions:

- Longitudinal velocity is constant,
- Left and right axle are lumped into a single wheel (bicycle model),
- Suspension movement, road inclination and aerodynamic influences are neglected,
- Decoupled longitudinal and lateral motion.



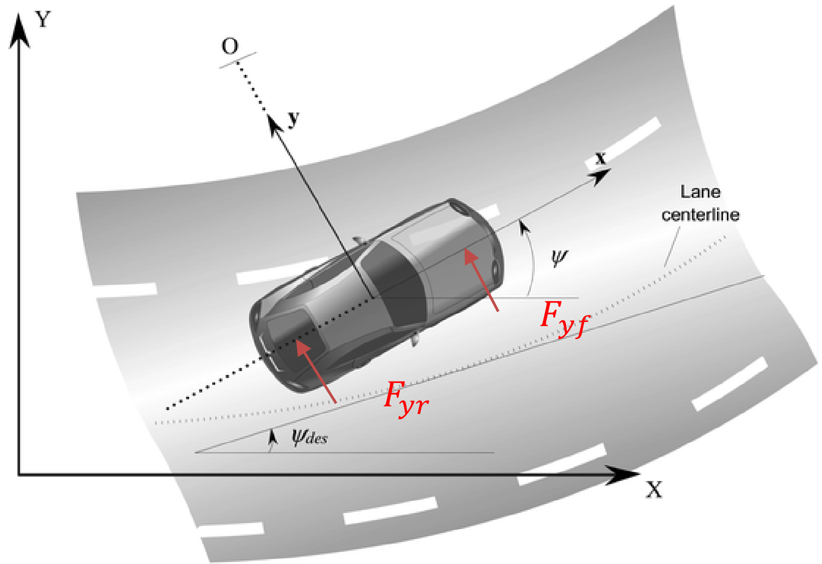
Dynamic Model of Lateral Vehicle Motion

"Bicycle" dynamic model:

$$a_y = \left(\frac{\mathrm{d}^2 y}{\mathrm{d}t^2}\right)_{inertial} = \dot{v}_y + v_x\dot{\psi}$$

$$F_{yf} + F_{y_r} = ma_y = m\left(\dot{v}_y + v_x\dot{\psi}\right)$$

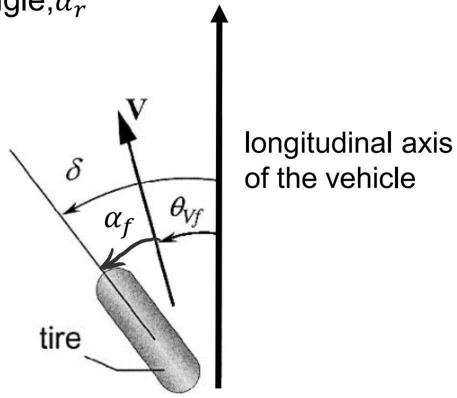
$$l_f F_{yf} - l_r F_{yr} = I_z\ddot{\psi}$$



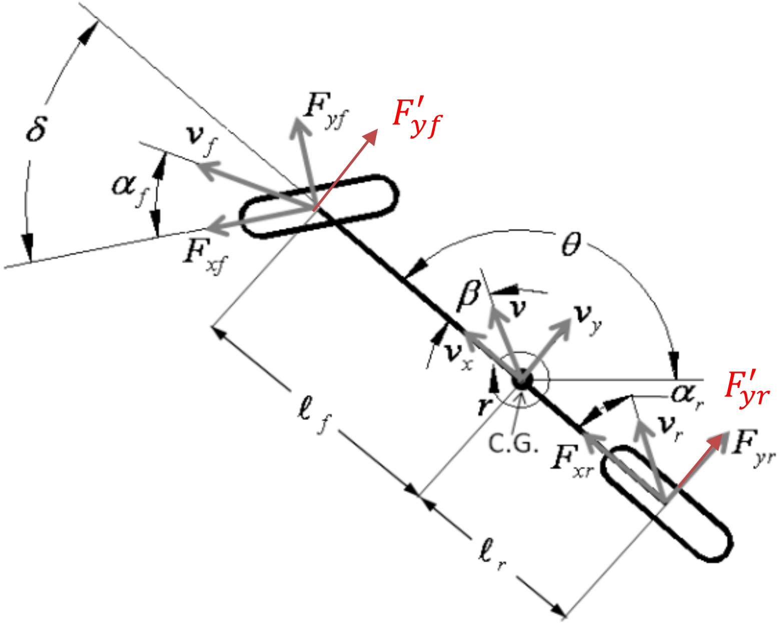
We can also use $r = \dot{\psi}$ to represent the angular rate about the yaw axis.

Tire Slip Angle

- Many different tire slip models
- For *small slip-angles, e*xperimental results show that the linear function of tire slip angle (α)
- Tire variables:
  - Front tire slip angle, α_f
  - Rear tire slip angle, α_r

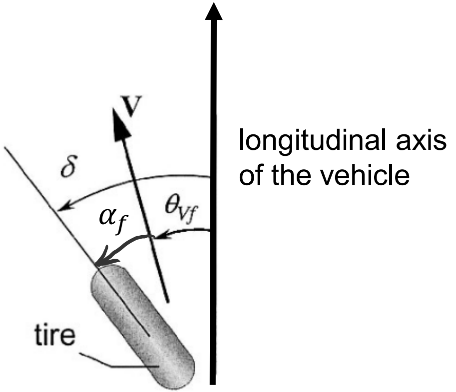


Tire slip-angle

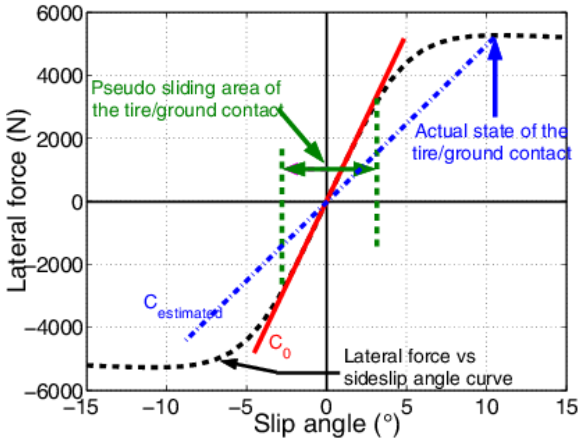


Dynamic Bicycle Model

Front and Rear Tire Forces



Tire slip-angle



- For *small slip-angles, e*xperimental results show that the lateral tire force of a tire is proportional to the "slip-angle“,

$$F_{yf} = 2c_f\alpha_f = 2c_f(\delta - \theta_{vf})$$

$$F_{yr} = 2c_r\alpha_r = 2c_r(-\theta_{vr})$$

where c_f , c_r are the cornering stiffness of tire.

Front and Rear Tire Forces

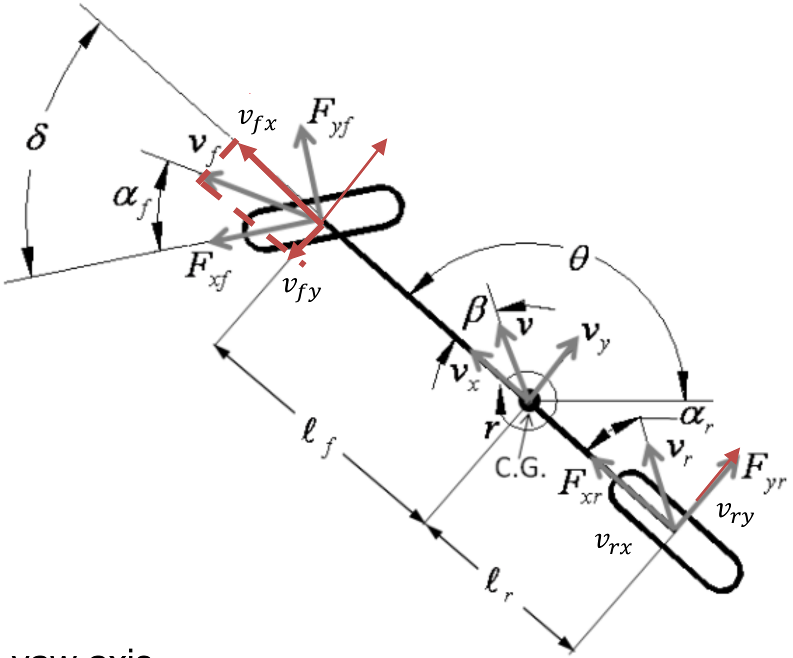
$$F_{yf} = 2c_f(\delta - \theta_{vf})$$

$$F_{yr} = 2c_r(-\theta_{vr})$$

$$\theta_{vf} = \tan^{-1}(\frac{v_{fy}}{v_{fx}}) = \tan^{-1}\left(\frac{v_y + l_f r}{v_x}\right)$$

$$\theta_{vr} = \tan^{-1}(\frac{v_{ry}}{v_{rx}}) = \tan^{-1}\left(\frac{v_y - l_r r}{v_x}\right)$$

where $r = \dot{\psi}$ is the yaw rate / the angular rate about the yaw axis.



Dynamic Bicycle Model

Front and Rear Tire Forces

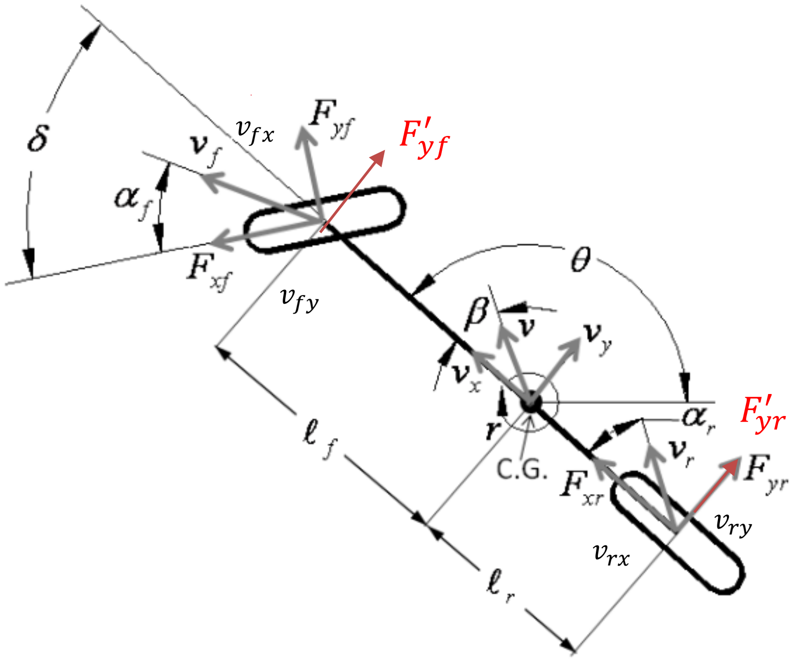
$$F'_{yf} + F'_{yr} = ma_y = m(\dot{v}_y + v_x\dot{\psi})$$

$$l_f F'_{yf} - l_r F'_{yr} = I_z\ddot{\psi}$$

$$\Downarrow$$

$$(F_{yf}\cos(\delta) - F_{xf}\sin(\delta)) + F_{yr} = m(\dot{v}_y + v_x r)$$

$$\ell_f\big(F_{yf}\cos(\delta) - F_{xf}\sin(\delta)\big) - \ell_r F_{yr} = I_z\ddot{\psi} = I_z\dot{r}$$



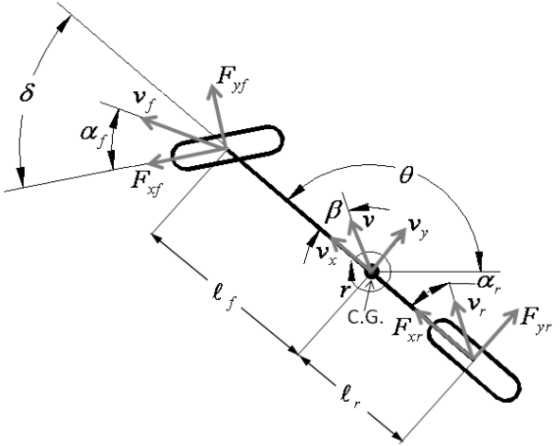
Dynamic Bicycle Model

Lateral and Yaw Dynamics

Summing the lateral forces illustrated (longitudinal velocity is assumed to be controlled separately.):

$$(F_{yf}\cos(\delta) - F_{xf}\sin(\delta)) + F_{yr} = m(\dot{v}_y + v_x r)$$

$$\ell_f\left(F_{yf}\cos(\delta) - F_{xf}\sin(\delta)\right) - \ell_r F_{yr} = I_z\dot{r}$$



where r is the angular rate about the yaw axis.(short form)

Without the constraint on lateral slip from the last section and assume constant speed, we have

$$\alpha_f = \delta - \tan^{-1}\left(\frac{v_y + \ell_f r}{v_x}\right)$$

$$\alpha_r = -\tan^{-1}\left(\frac{v_y - \ell_r r}{v_x}\right)$$

$$F_{yf} = c_f\alpha_f = c_f\left[\delta - \tan^{-1}\left(\frac{v_y + \ell_f r}{v_x}\right)\right]$$

$$F_{yr} = c_r\alpha_r = -c_r\tan^{-1}\left(\frac{v_y - \ell_r r}{v_x}\right)$$

$$\dot{v}_y = \frac{c_f\left[\delta - \tan^{-1}\left(\frac{v_y + \ell_f r}{v_x}\right)\right]\cos(\delta) - c_r\tan^{-1}\left(\frac{v_y - \ell_r r}{v_x}\right) - F_{xf}sin(\delta)}{m} - v_x r$$

$$\dot{r} = \frac{\ell_f c_f\left[\delta - \tan^{-1}\left(\frac{v_y + \ell_f r}{v_x}\right)\right]\cos(\delta) + \ell_r c_r\tan^{-1}\left(\frac{v_y - \ell_r r}{v_x}\right) - l_f F_{xf}sin(\delta)}{I_z}$$

Linearized dynamic bicycle model

1. Applying small angle assumptions

$$\cos(\delta) \approx 1$$

$$sin(\delta) \approx 0$$

$$\tan^{-1}(\theta) \approx \theta$$

$$+$$

$$\dot{v}_y = \frac{c_f\left[\delta - \tan^{-1}\left(\frac{v_y + \ell_f r}{v_x}\right)\right]\cos(\delta) - c_r \tan^{-1}\left(\frac{v_y - \ell_r r}{v_x}\right) - F_{xf} sin(\delta)}{m} - v_x r$$

$$\dot{r} = \frac{\ell_f c_f\left[\delta - \tan^{-1}\left(\frac{v_y + \ell_f r}{v_x}\right)\right]\cos(\delta) + \ell_r c_r \tan^{-1}\left(\frac{v_y - \ell_r r}{v_x}\right) + l_f F_{xf} sin(\delta)}{I_z}$$

$$\Longrightarrow$$

$$\dot{v}_y = \frac{-c_f v_y - c_f \ell_f r}{m v_x} + \frac{c_f \delta}{m} + \frac{-c_r v_y + c_r \ell_r r}{m v_x} - v_x r$$

$$\dot{r} = \frac{-\ell_f c_f v_y - \ell_f^2 c_f r}{I_z v_x} + \frac{\ell_f c_f \delta}{I_z} + \frac{\ell_r c_r v_y - \ell_r^2 c_r r}{I_z v_x} .$$

Linearized dynamic bicycle model

2. Re-group by variables

$$\dot{v}_y = \frac{-c_f v_y - c_f \ell_f r}{m v_x} + \frac{c_f \delta}{m} + \frac{-c_r v_y + c_r \ell_r r}{m v_x} - v_x r$$

$$\dot{r} = \frac{-\ell_f c_f v_y - \ell_f^2 c_f r}{I_z v_x} + \frac{\ell_f c_f \delta}{I_z} + \frac{\ell_r c_r v_y - \ell_r^2 c_r r}{I_z v_x} \quad .$$

$$\Longrightarrow$$

$$\dot{v}_y = \frac{-(c_f + c_r)}{m v_x} v_y + \left[\frac{(l_r c_r - l_f c_f)}{m v_x} - v_x \right] r + \frac{c_f}{m} \delta$$

$$\dot{r} = \frac{l_r c_r - l_f c_f}{I_z v_x} v_y + \frac{-(\ell_f^2 c_f + \ell_r^2 c_r)}{I_z v_x} r + \frac{l_f c_f}{I_z} \delta$$

Linearized dynamic bicycle model

3. Re-write into state space equation by using state $\mathrm{X} = \begin{bmatrix} v_y \\ r \end{bmatrix}$

$$\dot{v_y} = \frac{-(c_f + c_r)}{m v_x} v_y + \left[\frac{\left(l_r c_r - l_f c_f\right)}{m v_x} - v_x\right] r + \frac{c_f}{m}\delta$$

$$\dot{r} = \frac{l_r c_r - l_f c_f}{I_z v_x} v_y + \frac{-(\ell_f^2 c_f + \ell_r^2 c_r)}{I_z v_x} r + \frac{l_f c_f}{I_z}\delta$$

$$\Longrightarrow \boxed{\begin{bmatrix} \dot{v_y} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \dfrac{-(c_f + c_r)}{m v_x} & \dfrac{\left(l_r c_r - l_f c_f\right)}{m v_x} - v_x \\ \dfrac{l_r c_r - l_f c_f}{I_z v_x} & \dfrac{-(\ell_f^2 c_f + \ell_r^2 c_r)}{I_z v_x} \end{bmatrix} \begin{bmatrix} v_y \\ r \end{bmatrix} + \begin{bmatrix} \dfrac{c_f}{m} \\ \dfrac{l_f c_f}{I_z} \end{bmatrix} \delta}$$

Linearized Dynamic Model of Lateral Vehicle Motion

$$\begin{bmatrix} \dot{v}_y \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{-(c_f + c_r)}{m v_x} & \frac{(l_r c_r - l_f c_f)}{m v_x} - v_x \\ \frac{l_r c_r - l_f c_f}{I_z v_x} & \frac{-(\ell_f^2 c_f + \ell_r^2 c_r)}{I_z v_x} \end{bmatrix} \begin{bmatrix} v_y \\ r \end{bmatrix} + \begin{bmatrix} \frac{c_f}{m} \\ \frac{l_f c_f}{I_z} \end{bmatrix} \delta$$

If we use state $X = \begin{bmatrix} y \\ \dot{y} \\ \psi \\ \dot{\psi} \end{bmatrix}$, input δ,

rewrite in state space model $\dot{X} = AX + B\delta$, it is

$$\frac{d}{dt} \begin{bmatrix} y \\ \dot{y} \\ \psi \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-(c_f + c_r)}{m v_x} & 0 & \frac{(l_r c_r - l_f c_f)}{m v_x} - v_x \\ 0 & 0 & 0 & 1 \\ 0 & \frac{l_r c_r - l_f c_f}{I_z v_x} & 0 & \frac{-(\ell_f^2 c_f + \ell_r^2 c_r)}{I_z v_x} \end{bmatrix} \begin{bmatrix} y \\ \dot{y} \\ \psi \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{c_f}{m} \\ 0 \\ \frac{l_f c_f}{I_z} \end{bmatrix} \delta$$

Model Parameter Identification

Unlike the previous models, the dynamic bicycle model has parameters that are not as convenient to directly measure. However, a workable estimate can be obtained using commonly available tools, such as utilizing four scales under each wheel:

C.G. estimation:

$$\ell_f = L(1 - \frac{m_f}{m})$$

$$\ell_f = L(1 - \frac{m_r}{m})$$

$$m = m_{fr} + m_{fl} + m_{rr} + m_{rl}$$

$$m_f = m_{fr} + m_{fl}$$

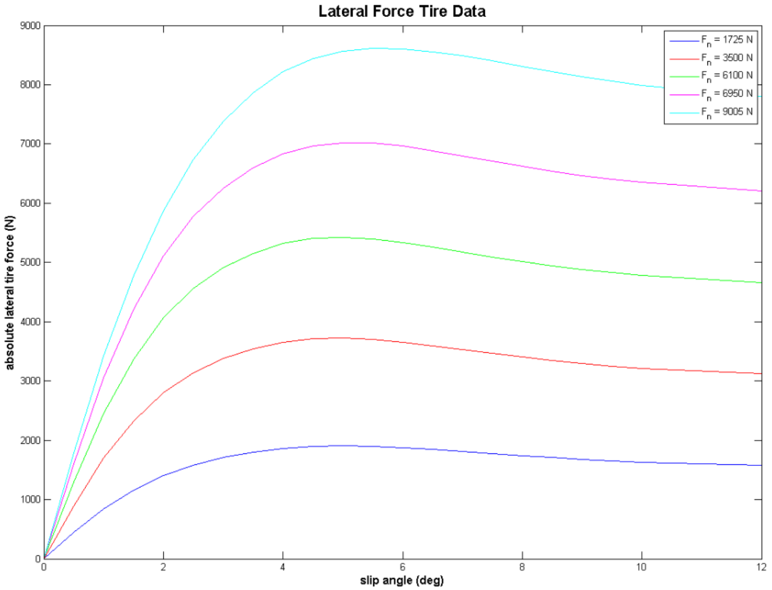
$$m_r = m_{rr} + m_{rl}$$

Vehicle's moment of inertia is approximated by treating the vehicle as two point masses joined by a mass-less rod:

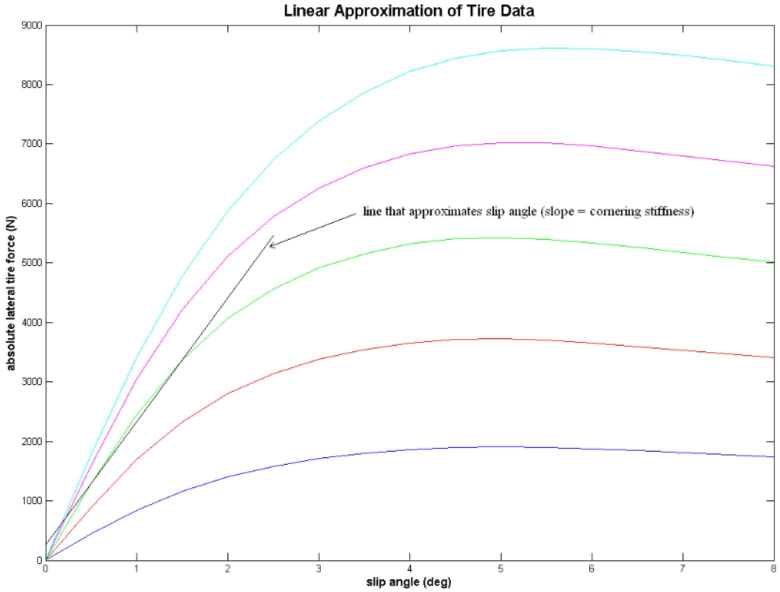
$$I_Z = m_f \ell_f^2 + m_r \ell_r^2$$

Model Parameter Identification

Cornering stiffness parameters: a constant of describing the slope in the most linear region of the data at a nominal normal force.



Example of lateral force tire data: *the slip angle of the tire changes nonlinearly as the lateral force on the tire changes*



Linear approximation of the lateral force tire data

Model Parameter Identification

Cornering stiffness parameters: when the detailed data is not readily available, a method to estimate it is required.

$$\dot{v}_y = \frac{-c_f\left[\tan^{-1}\left(\frac{v_y + \ell_f r}{v_x}\right) - \delta\right]\cos(\delta) - c_r \tan^{-1}\left(\frac{v_y - \ell_r r}{v_x}\right)}{m} - v_x r$$

$$\dot{r} = \frac{-\ell_f c_f\left[\tan^{-1}\left(\frac{v_y + \ell_f r}{v_x}\right) - \delta\right]\cos(\delta) + \ell_r c_r \tan^{-1}\left(\frac{v_y - \ell_r r}{v_x}\right)}{I_z}$$

$$\dot{v}_y + v_x r = \left(\frac{-\ell_f r - v_y}{m v_x} + \frac{\delta}{m}\right) c_f + \left(\frac{\ell_r r - v_y}{m v_x}\right) c_r$$

$$\dot{r} = \left(\frac{-(\ell_f v_y + \ell_f^2 r)}{I_z v_x} + \frac{\ell_f \delta}{I_z}\right) c_f + \left(\frac{\ell_r v_y - \ell_r^2 r}{I_z v_x}\right) c_r$$

Model Parameter Identification

Cornering stiffness estimation:

Assuming that $\dot{v}_y$ and $\dot{r}$ are not directly measured, and that v and r are available. Use Euler method, rewrite

$$\dot{v}_y + v_x r = \left(\frac{-\ell_f r - v_y}{m v_x} + \frac{\delta}{m}\right) c_f + \left(\frac{\ell_r r - v_y}{m v_x}\right) c_r$$

$$\dot{r} = \left(\frac{-(\ell_f v_y + \ell_f^2 r)}{I_z v_x} + \frac{\ell_f \delta}{I_z}\right) C_f + \left(\frac{\ell_r v_y - \ell_r^2 r}{I_z v_x}\right) c_r$$

in a discrete form:

$$v_y(k+1) - v_y(k) + v_x(k) r(k) \Delta t = \left(\frac{-\ell_f r(k) - v_y(k)}{m v_x(k)} + \frac{\delta(k)}{m}\right) \Delta t c_f + \left(\frac{\ell_r r(k) - v_y(k)}{m v_x(k)}\right) \Delta t c_r$$

$$r(k+1) - r(k) = \left(\frac{-(\ell_f v_y(k) + \ell_f^2 r(k))}{I_z v_x(k)} + \frac{\ell_f \delta(k)}{I_z}\right) \Delta t c_f + \left(\frac{\ell_r v_y(k) - \ell_r^2 r(k)}{I_z v_x(k)}\right) \Delta t c_r$$

A least squares problem!

where k is an index of the measurement set and Δt is the time between measurement sets.

Model Parameter Identification

Cornering stiffness estimation: A least squares problem!

Rewrite into the following form:

$$A = B\begin{bmatrix} c_f \\ c_r \end{bmatrix}$$

The least squares formulation then follows as

$$A = \begin{bmatrix} v_y(2) - v_y(1) + v_x(1)\, r(1)\,\Delta t \\ r(2) - r(1) \\ \vdots \\ v_y(n) - v_y(n-1) + v_x(n-1)\, r(n-1)\,\Delta t \\ r(n) - r(n-1) \end{bmatrix} \quad B = \begin{bmatrix} \left(\frac{-\ell_f r(1) - v_y(1)}{m v_x(1)} + \frac{\delta(1)}{m}\right)\Delta t & \left(\frac{\ell_r r(1) - v_y(1)}{m v_x(1)}\right)\Delta t \\ \left(\frac{-(\ell_f v_y(1) + \ell_f^2 r(1)}{I_z v_x(1)} + \frac{\ell_f \delta(1)}{I_z}\right)\Delta t & \left(\frac{\ell_r v_y(1) - \ell_r^2 r(1)}{I_z v_x(1)}\right)\Delta t \\ \vdots & \vdots \\ \left(\frac{-\ell_f r(n-1) - v_y(n-1)}{m v_x(n-1)} + \frac{\delta(n-1)}{m}\right)\Delta t & \left(\frac{\ell_r r(n-1) - v_y(n-1)}{m v_x(n-1)}\right)\Delta t \\ \left(\frac{-(\ell_f v_y(n-1) + \ell_f^2 r(n-1)}{I_z v_x(n-1)} + \frac{\ell_f \delta(n-1)}{I_z}\right)\Delta t & \left(\frac{\ell_r v_y(n-1) - \ell_r^2 r(n-1)}{I_z v_x(n-1)}\right)\Delta t \end{bmatrix}$$

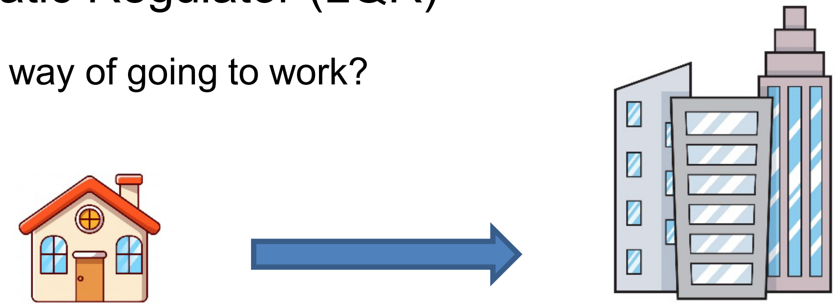
The best results are obtained when the lateral force on the tires are moderate (without exciting the suspension too much) and continuously varying throughout the data collection.

Linear Optimal Control

BACKGROUND

Linear Quadratic Regulator (LQR)

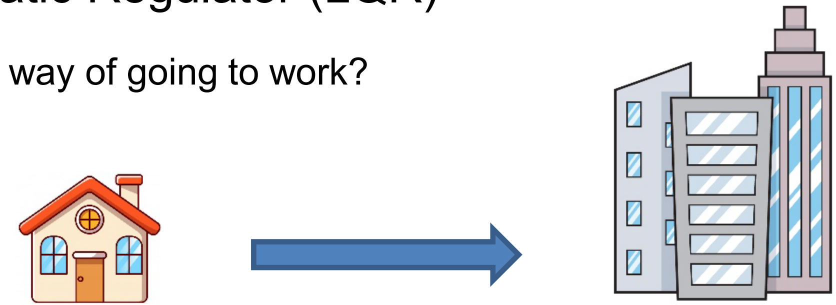
What is the “best” way of going to work?



		time/trip		money/trip	
		20 min		$7	
		75 min		$0	
		30 min		$2	
		4 min		$400	

Linear Quadratic Regulator (LQR)

What is the “best” way of going to work?

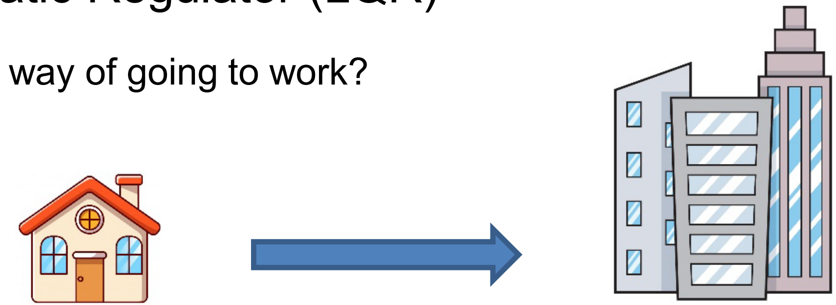


Cost: $J = Q \times time + R \times money$

	Q	time/trip	R	money/trip	Cost
	1	20 min	1	$7	27
	1	75 min	1	$0	75
	1	30 min	1	$2	32
	1	4 min	1	$400	404

Linear Quadratic Regulator (LQR)

What is the “best” way of going to work?



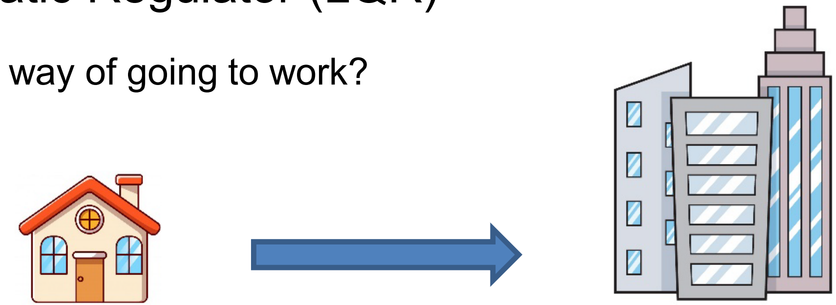
Cost: $J = Q \times time + R \times money$

	Q	time/trip	R	money/trip	Cost
	30	20 min	1	\$7	607
	30	75 min	1	\$0	2250
	30	30 min	1	\$2	902
	30	4 min	1	\$400	520

High cost

Linear Quadratic Regulator (LQR)

What is the “best” way of going to work?



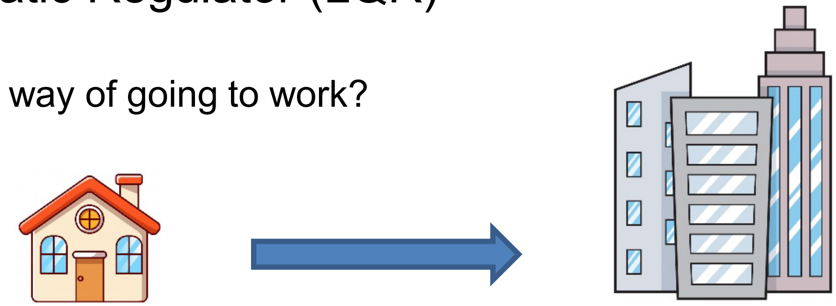
Cost: $J = Q \times time + R \times money$

	Q	time/trip	R	money/trip	Cost
	1	20 min	40	$7	300
	1	75 min	40	$0	75
	1	30 min	40	$2	110
	1	4 min	40	$400	16004

High cost

Linear Quadratic Regulator (LQR)

What is the “best” way of going to work?

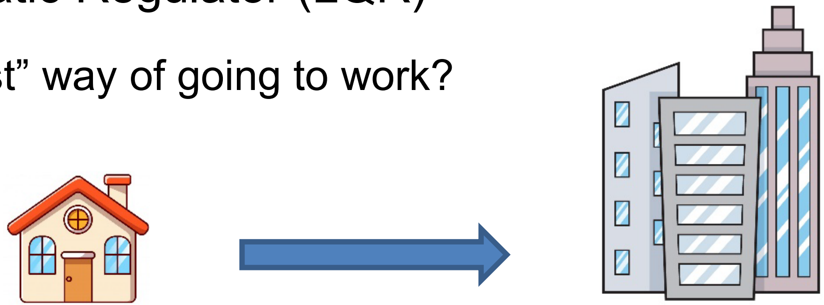


Cost: $J = Q \times time + R \times money$

	Q	time/trip	R	money/trip	Cost
	2	20 min	10	\$7	110
	2	75 min	10	\$0	150
	2	30 min	10	\$2	80
	2	4 min	10	\$400	4008

Linear Quadratic Regulator (LQR)

What is the “best” way of going to work?



Cost: $J = Q \times time + R \times money$

	Q	time/trip	R	money/trip	Cost
	1	20 min	5	$7	55
	1	75 min	5	$0	75
	1	30 min	5	$2	40
	1	4 min	5	$400	2004

Linear Quadratic Regulator (LQR)

A system is described by the standard linear state space model:

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$$

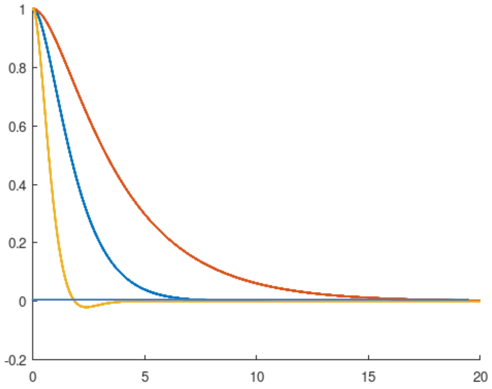
$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x}$$

The objective is to bring the non-zero initial state to zero in the infinite time horizon.

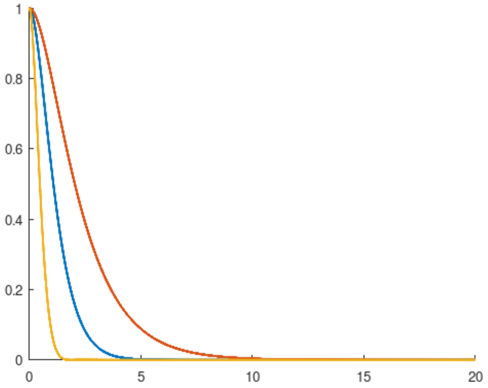
The cost function takes the quadratic form:

$$J = \frac{1}{2}\int_0^{\infty} (\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u})dt$$

Area under the curve indicates the total "cost" of bringing state to 0



We don’t want negative value to subtract the sum. Therefore, we sum the square value of the state



Linear Quadratic Regulator (LQR)

The matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ appear most often in diagonal form. Though there is no inherent restriction to such a form,

$$J = \frac{1}{2}\int_0^{\infty} (\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u})dt$$

Note from $\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} = q_1x_1^2 + q_2x_2^2 + \cdots + q_nx_n^2$,

Where $q_i \geq 0, i = 1, 2, \cdots, n$ and $r_i > 0,\ i = 1, 2, \cdots, m$.

$$Q = \begin{bmatrix} q_1 & & & 0 \\ & q_2 & & \\ & & \ddots & \\ 0 & & & q_n \end{bmatrix}, \qquad R = \begin{bmatrix} r_1 & & & 0 \\ & r_2 & & \\ & & \ddots & \\ 0 & & & r_m \end{bmatrix},$$

- q_i are relative weightings among x_i.
- if q_1 is bigger than q_2, there is higher penalty/price on error x_1 than x_2, and control will try to make smaller than, vice versa.

The same can be said on $\boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u} = r_1u_1^2 + r_2u_2^2 + \cdots + r_mu_m^2$.

LQR: a scalar example

For the plant: $\dot{x} = x + u, x(0) \neq 0$,

we want to regulate the state to $x = u = 0$.

- Assume that we do not wish to apply any more control effort than is necessary.
- For example, we might wish to avoid saturation of the control elements or to use as little power as possible. Thus we should keep u as well as x near zero.

The following extension of the integral squares error (ISE) index expresses this mathematically:

$$J = \frac{1}{2}\int_0^\infty (qx^2 + ru^2)dt$$

- The weighting factors $q \geq 0$ and $r > 0$ express the relative importance of keeping x and u near zero.

LQR: a scalar example

The optimal control is to find the control law that minimizes this J.

- *Let* $u = -Kx,\ r = 1$
- Then we have
$$J = \frac{1}{2}\int_0^x (qx^2 + u^2)dt = \frac{1}{2}(q + K^2)\int_0^\infty x^2 dt$$
$$\dot{x} = x - Kx = -(K-1)x$$
- Its solution for constant K is $x(t) = x(0)e^{-(K-1)t}$.
- And the system is stable for $K > 1$. Substitute $x(t)$ into J, we have:
$$J = \frac{1}{2}(q + K^2)x^2(0)\int_0^\infty e^{-2(K-1)t}\,dt = \frac{q + K^2}{4(K-1)}x^2(0)$$
- To minimize J for fixed q and $x(0)$, we let $\frac{\partial J}{\partial K} = 0$, this gives $K^2 - 2K - q = 0$.

  Its roots are: $K_1 = 1 + \sqrt{1+q}, \quad K_2 = 1 - \sqrt{1+q}$.

LQR: a scalar example

For a minimum we require that $\partial^2 J/\partial^2 K \geq 0$, this implies that $K - 1 \geq 0$. Because $q \geq 0$, K_1, will satisfy this condition. In this case, $x(t) \rightarrow 0$ and J has a minimum value of

$$J_{min} = \frac{1}{2}\left(1 + \sqrt{1+q}\right)x^2(0)$$

$$K_1 = 1 + \sqrt{1+q}$$

It says that the value of K that minimizes J must be such that the closed-loop system will be stable.

Finally, we note that the design problem has been transformed into one of selecting a value for q.

- The larger q is, the larger will be the gain K, and the faster will $x(t)$ approach zero.
- However, the peak magnitude of u will be larger.
- The parameter q is selected to achieve a compromise between these effects. We will indicate a general procedure for doing this.

$$x(t) = x(0)e^{-(K-1)t}$$

$$u = -Kx$$

LQR: General Solution (Raccati Equation)

For a LQR problem defined as

System
$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$$
$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x}$$

State feedback $\boldsymbol{u} = -\boldsymbol{K}\boldsymbol{x}$

The closed loop system is: $\dot{\boldsymbol{x}} = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K})\boldsymbol{x} = \boldsymbol{A}_{\boldsymbol{c}}\boldsymbol{x}$.

We assume that $\boldsymbol{K}$ is such that this system is stable.

Cost function
$$J = \frac{1}{2}\int_0^\infty (\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u})\,dt$$

Will bring $x(\infty) \to 0$ and $\mathrm{u}(\infty) \to 0$.

Let's bring $\boldsymbol{u} = -\boldsymbol{K}\boldsymbol{x}$ Into cost function, we have $J = \frac{1}{2}\int_0^\infty \boldsymbol{x}^T(\boldsymbol{Q} + \boldsymbol{K}^T\boldsymbol{R}\boldsymbol{K})\boldsymbol{x}\,dt$

LQR: General Solution (Raccati Equation)

$$J = \frac{1}{2}\int_0^\infty \boldsymbol{x}^T(\boldsymbol{Q} + \boldsymbol{K}^T\boldsymbol{R}\boldsymbol{K})\boldsymbol{x}\,dt$$

Let $\boldsymbol{P}$, an $n \times n$ symmetric matrix, and

$$\frac{d}{dt}(x^T P x) = -\boldsymbol{x}^T(\boldsymbol{Q} + \boldsymbol{K}^T\boldsymbol{R}\boldsymbol{K})\boldsymbol{x}$$

So that we have

$$J = -\frac{1}{2}\int_0^\infty \frac{d}{dt}(x^T P x)$$

$$= -\frac{1}{2}(x^T(\infty)Px(\infty) - x^T(0)Px(0))$$

$$= \frac{1}{2}(x^T(0)Px(0))$$

LQR: General Solution (Raccati Equation)

Let $\boldsymbol{P}$, an $n \times n$ symmetric matrix, and $\frac{d}{dt}\left(\boldsymbol{x}^T\boldsymbol{P}\boldsymbol{x}\right) = -\boldsymbol{x}^T(\boldsymbol{Q} + \boldsymbol{K}^T\boldsymbol{R}\boldsymbol{K})\boldsymbol{x}$

Differentiate both sides, we will have

$$\dot{\boldsymbol{x}}^T\boldsymbol{P}\boldsymbol{x} + \boldsymbol{x}^T\boldsymbol{P}\dot{\boldsymbol{x}} + \boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{x}^T\boldsymbol{K}^T\boldsymbol{R}\boldsymbol{K}\boldsymbol{x} = \mathbf{0}$$

We know the closed loop system is: $\dot{\boldsymbol{x}} = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K})\boldsymbol{x} = \boldsymbol{A}_c\boldsymbol{x}$

$$\boldsymbol{x}^T\boldsymbol{A}_c^T\boldsymbol{P}\boldsymbol{x} + \boldsymbol{x}^T\boldsymbol{P}\boldsymbol{A}_c\boldsymbol{x} + \boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{x}^T\boldsymbol{K}^T\boldsymbol{R}\boldsymbol{K}\boldsymbol{x} = \mathbf{0}$$

$$\boldsymbol{x}^T\left(\boldsymbol{A}_c^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}_c + \boldsymbol{Q} + \boldsymbol{K}^T\boldsymbol{R}\boldsymbol{K}\right)\boldsymbol{x} = \mathbf{0}$$

This is a quadratic form, only have solution when $\boldsymbol{A}_c^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}_c + \boldsymbol{Q} + \boldsymbol{K}^T\boldsymbol{R}\boldsymbol{K} = \mathbf{0}$

$$(\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K})^T\boldsymbol{P} + \boldsymbol{P}(\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}) + \boldsymbol{Q} + \boldsymbol{K}^T\boldsymbol{R}\boldsymbol{K} = \mathbf{0}$$

$$\boldsymbol{A}^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} + \boldsymbol{Q} + \boldsymbol{K}^T\boldsymbol{R}\boldsymbol{K} - \boldsymbol{K}^T\boldsymbol{B}^T\boldsymbol{P} - \boldsymbol{P}\boldsymbol{B}\boldsymbol{K} = \mathbf{0}$$

Let $\boldsymbol{K} = \boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P}$, we have $\boldsymbol{A}^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} + \boldsymbol{Q} + \boldsymbol{K}^T\boldsymbol{R}(\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P}) - \boldsymbol{K}^T\boldsymbol{B}^T\boldsymbol{P} - \boldsymbol{P}\boldsymbol{B}(\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P}) = \mathbf{0}$

$$\boldsymbol{A}^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} + \boldsymbol{Q} = \boldsymbol{P}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P}$$

This equation is called the *Algebraic Riccati Equation (CARE)*.

LQR: General Solution (Raccati Equation)

For a LQR problem defined as

System $$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$$ $$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x}$$

State feedback $\boldsymbol{u} = -\boldsymbol{K}\boldsymbol{x}$

The closed loop system is: $\dot{\boldsymbol{x}} = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K})\boldsymbol{x} = \boldsymbol{A}_{\boldsymbol{c}}\boldsymbol{x}$.

Cost function

$$J = \frac{1}{2}\int_0^\infty (\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u})\, dt$$

Control law: $\boldsymbol{u} = -\boldsymbol{K}\boldsymbol{x} = -\boldsymbol{R}^{-1}\boldsymbol{B}^{\boldsymbol{T}}\boldsymbol{P}\boldsymbol{x}$, will bring $x(\infty) \to 0$ and $\mathrm{u}(\infty) \to 0$

where $\boldsymbol{A}^{\boldsymbol{T}}\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} + \boldsymbol{Q} = \boldsymbol{P}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^{\boldsymbol{T}}\boldsymbol{P}$

that is, the optimal control law is a linear feedback of the state vector $\boldsymbol{x}$, as assumed.

Case study: LQR tuning

Consider a simple mass system (with friction b) connected to a linear motor to maintain its position (e.g. a UAV hovering at a fixed position).

At time $t = 0$, a disturbance (wind) makes the mass block to displace to new location x_0. We would like to design a LQR controller to make the mass block move back to its original location in an "optimal" way.

We choose system states to be , therefore, the state space representation of the system can be written as:

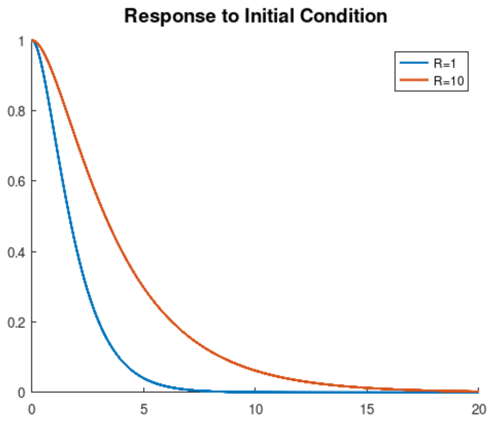
$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{b}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} f$$

Case study: LQR tuning

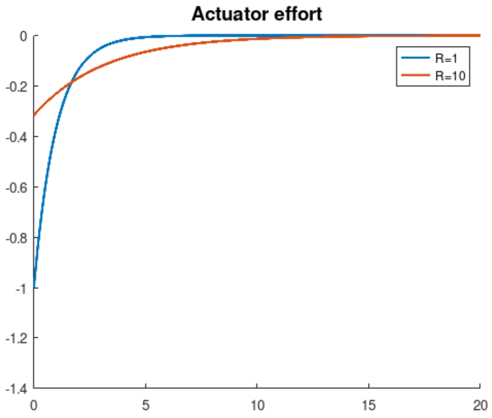
$$J = \frac{1}{2}\int_0^{\infty} (\boldsymbol{x}^T \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^T \boldsymbol{R}\boldsymbol{u})\, dt$$

Let $Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, we choose different R

- When $R = 10$
  - Penalty on control effort is large → Less control effort is used → slower response
- When $R = 1$
  - Penalty on control effort is small → More control effort is used → faster response



Position error response to initial condition



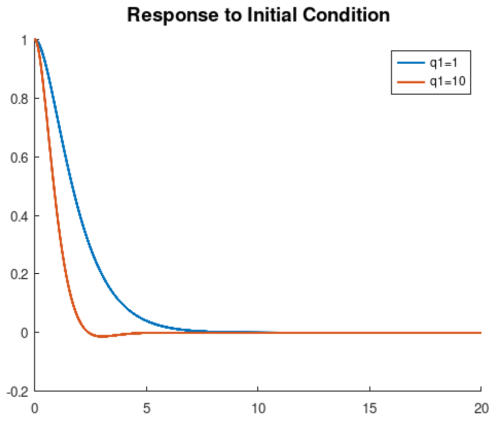
Control effort

Case study: LQR tuning

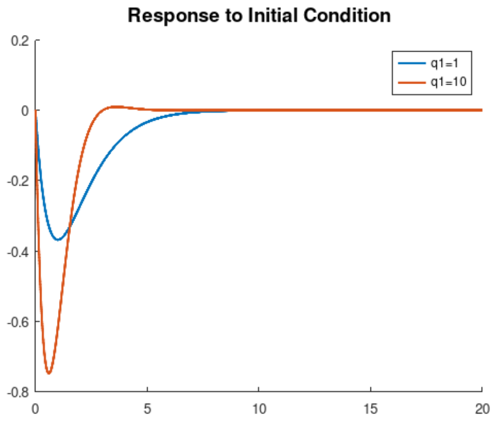
$$J = \frac{1}{2}\int_0^{\infty} (\boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} + \boldsymbol{u}^T \boldsymbol{R} \boldsymbol{u})\, dt$$

Let $R = 1$, we choose different Q

- When $Q = \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix}$
  - Penalty on position error > Penalty on speed error → faster response
- When $Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
  - Penalty on position error = Penalty on speed error → slower response



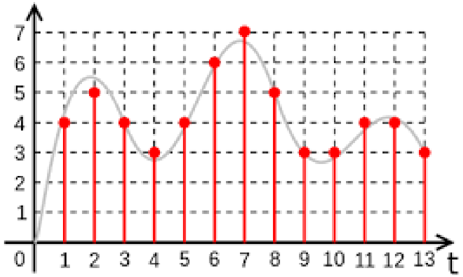
Position error response to initial condition

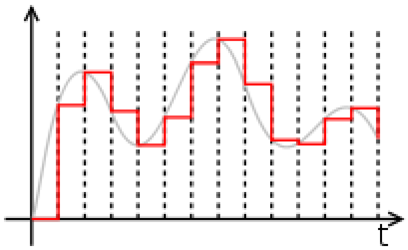


Speed error response to initial condition

From continuous time to discrete time

Continuous time model to discrete time model





$$\dot{x} = Ax + Bu \qquad \xrightarrow{\text{zero order hold}} \qquad x(k+1) = \mathrm{A}_d x(k) + B_d u(k)$$

$$y = Cx \qquad\qquad y(k) = C_d x(k)$$

- Where $A_d = e^{A\Delta t}, B_d = \left(\int_0^{\Delta t} e^{A\tau} d\tau\right) B$
- How to calculate e^{At}? $e^{At} = L^{-1}((sI - A)^{-1})$

[From Continuous Time to Discrete Time]()

Discrete time LQR

Apply LQR in this situation, we have

$$\boldsymbol{u}^*(k) = -\boldsymbol{K}\boldsymbol{x}(k)$$

Where $\boldsymbol{K} = (\boldsymbol{R} + \boldsymbol{B}_d^T\boldsymbol{P}\boldsymbol{B}_d)^{-1}\ \boldsymbol{B}_d^T\boldsymbol{P}\boldsymbol{A}_d$.

Objective cost function to be minimized by the control is

$$J = \sum_{k=0}^{\infty} \boldsymbol{x}^T(k)\boldsymbol{Q}\boldsymbol{x}(k) + \boldsymbol{u}^T(k)\boldsymbol{R}\boldsymbol{u}(k)$$

where $\boldsymbol{P}$ satisfies the matrix difference Riccati equation (DARE)

$$\boldsymbol{P} = \boldsymbol{A}_d^T\boldsymbol{P}\boldsymbol{A}_d - \boldsymbol{A}_d^T\boldsymbol{P}\boldsymbol{B}_d(\boldsymbol{R} + \boldsymbol{B}_d^T\boldsymbol{P}\boldsymbol{B}_d)^{-1}\ \boldsymbol{B}_d^T\boldsymbol{P}\boldsymbol{A}_d + \boldsymbol{Q}$$

- $\boldsymbol{Q}$ is a diagonal weighting matrix with an entry for each state corresponding to the performance aspects contributing to the cost function and $\boldsymbol{R}$ is weighting value corresponding to the control effort contributing to the cost function.

https://stanford.edu/class/ee363/lectures/dlqr.pdf

Linear Quadratic Regulator (LQR) Summary

$$\boxed{J = \frac{1}{2}\int_0^{\infty} (\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u})\, dt}$$

What is LQR?

- LQR represents a class of control problems well: balance/tradeoff between minimization of system errors and minimization of control efforts

Why LQR?

- LQR has a neat solution: linear state feedback
- LQR is about easiest to solve among all optimal control problems $\boldsymbol{K} = \boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P}$
- LQR provides means for tuning.
- LQR guarantees a good robustness (phase marge > 60 degrees)

Trajectory tracking with LQR

Path Coordinates Model (Error dynamics)

对于路径跟踪，相对于其长度 s 路径函数和恒定纵向速度假设来表达自行车模型是有用的。

For path tracking, it is useful to express the bicycle model with respect to the path function of its length s and with the constant longitudinal velocity assumption.

We can choose $\boldsymbol{x} = \left(e_{cg}\ \ \dot{e}_{cg}\ \ e_{\theta}\ \ \dot{e}_{\theta}\right)^T$ as our system state and $\mathrm{u} = \delta$.

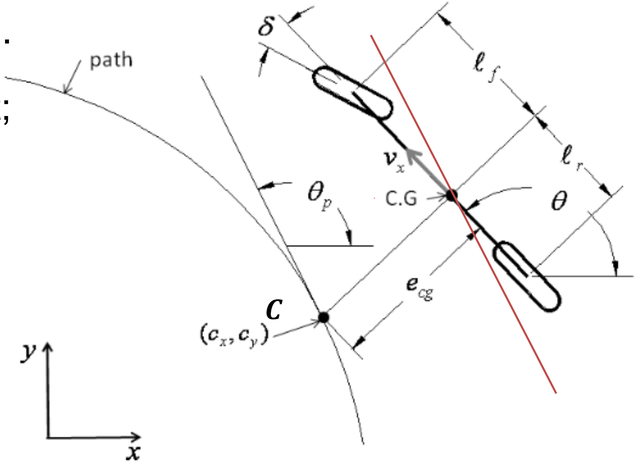
- e_{cg}: Orthogonal distance of the C.G. to the nearest path waypoint;
  车辆重心和路径之间的相对速度
- $\dot{e}_{cg}$: Relative speed between vehicle C.G and path;
- e_{θ}: Heading/Yaw difference between vehicle and path,

  $$e_{\theta} = \theta - \theta_p(s)$$

- $\dot{e_{\theta}}$: Relative yaw rate between vehicle C.G and path，

  $$\dot{e}_{\theta} = r - r(s)$$

  where $r(s) = \dot{\theta}(s)$ is the yaw rate derived from the path



Dynamic Bicycle Model in path coordinates

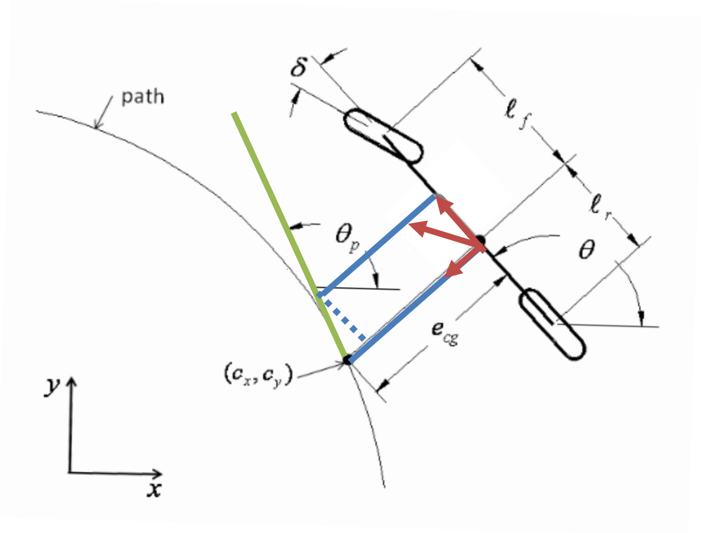
Path Coordinates Model (Error dynamics)

With the constant longitudinal velocity assumption,

$$\dot{e}_{cg} = v_y + v_x \tan(\theta - \theta_p(s))$$
$$= v_y + v_x \tan(e_\theta)$$

Thus, the acceleration of C.G. is:

$$\ddot{e}_{cg} = (\dot{v}_y + v_x r) - \dot{v}_y(s)$$
$$= \dot{v}_y + v_x(r - r(s))$$
$$= \dot{v}_y + v_x \dot{e}_\theta.$$



Dynamic Bicycle Model in path coordinates

Path Coordinates Model (Error dynamics) (Cont'd)

Convert lateral dynamic to error dynamics:

$$v_y = \dot{e}_{cg} - v_x \sin(e_\theta)$$

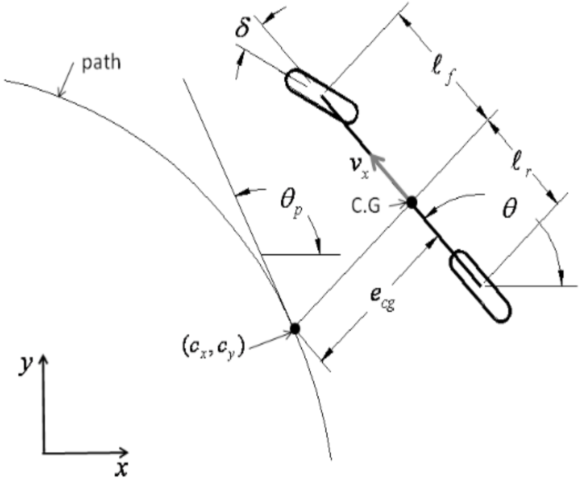
$$\dot{v}_y = \ddot{e}_{cg} - v_x \dot{e}_\theta$$

$$\theta = e_\theta + \theta_p(s)$$

$$r = \dot{e}_\theta + r(s)$$

$$\dot{r} = \ddot{e}_\theta + \dot{r}(s)$$

$$\boxed{\begin{aligned} \dot{v_y} &= \frac{-(c_f + c_r)}{m v_x} v_y + \left[\frac{(l_r c_r - l_f c_f)}{m v_x} - v_x\right] r + \frac{c_f}{m}\delta \\ \dot{r} &= \frac{l_r c_r - l_f c_f}{I_z v_x} v_y + \frac{-(\ell_f^2 c_f + \ell_r^2 c_r)}{I_z v_x} r + \frac{l_f c_f}{I_z}\delta \end{aligned}}$$



Dynamic Bicycle Model in path coordinates

Recall part 1

Path Coordinates Model (Error dynamics) (Cont'd)

$$\dot{v_y} = \frac{-(c_f + c_r)}{mv_x} v_y + \left[\frac{\left(l_r c_r - l_f c_f\right)}{mv_x} - v_x\right] r + \frac{c_f}{m}\delta$$

$$\dot{r} = \frac{l_r c_r - l_f c_f}{I_z v_x} v_y + \frac{-(\ell_f^2 c_f + \ell_r^2 c_r)}{I_z v_x} r + \frac{l_f c_f}{I_z}\delta$$

$$\dot{v_y} = \ddot{e}_{cg} - v_x \dot{e}_\theta = \frac{-\left(c_f + c_r\right)}{mv_x}(\dot{e}_{cg} - v_x e_\theta) + \left[\frac{\left(l_r c_r - l_f c_f\right)}{mv_x} - v_x\right](\dot{e}_\theta + r(s)) + \frac{c_f}{m}\delta$$

$$\dot{r} = \ddot{e}_\theta + \dot{r}(s) = \frac{l_r c_r - l_f c_f}{I_z v_x}(\dot{e}_{cg} - v_x e_\theta) + \frac{-(\ell_f^2 c_f + \ell_r^2 c_r)}{I_z v_x}(\dot{e}_\theta + r(s)) + \frac{l_f c_f}{I_z}\delta$$

$$\ddot{e}_{cg} = \frac{-\left(c_f + c_r\right)}{mv_x}\dot{e}_{cg} + \frac{\left(c_f + c_r\right)}{m} e_\theta + \frac{\left(l_r c_r - l_f c_f\right)}{mv_x}\dot{e}_\theta + \left[\frac{\left(l_r c_r - l_f c_f\right)}{mv_x} - v_x\right] r(s) + \frac{c_f}{m}\delta$$

$$\ddot{e}_\theta = \frac{l_r c_r - l_f c_f}{I_z v_x}\dot{e}_{cg} + \frac{l_r c_r - l_f c_f}{I_z} e_\theta + \frac{-\left(\ell_f^2 c_f + \ell_r^2 c_r\right)}{I_z v_x}\left(\dot{e}_\theta + r(s)\right) + \frac{l_f c_f}{I_z}\delta - \dot{r}(s)$$

Trajectory tracking with LQR

From the first section, we have the linear lateral dynamic model.

Rewrite it as: $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}_1\delta + \boldsymbol{B}_2 r_{des}$, where $\boldsymbol{x} = \left(e_{cg} \ \ \dot{e}_{cg} \ \ e_\theta \ \ \dot{e}_\theta\right)^T$.

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\dfrac{(c_f + c_r)}{mv} & \dfrac{c_f + c_r}{m} & \dfrac{l_r c_r - l_f c_f}{mv} \\ 0 & 0 & 0 & 1 \\ 0 & \dfrac{l_r c_r - l_f c_f}{I_z v} & \dfrac{l_r c_r - l_f c_f}{I_z} & -\dfrac{l_f^2 c_f + l_r^2 c_r}{I_z v} \end{bmatrix}, \boldsymbol{B}_1 = \begin{bmatrix} 0 \\ \dfrac{c_f}{m} \\ 0 \\ \dfrac{l_f c_f}{I_z} \end{bmatrix}, \boldsymbol{B}_2 = \begin{bmatrix} 0 \\ \dfrac{l_r c_r - l_f c_f}{mv} - v \\ 0 \\ -\dfrac{l_f^2 c_f + l_r^2 c_r}{I_z v} \end{bmatrix}$$

Basic work flow:

- Check the controllability matrix has full rank: $[\boldsymbol{B}_1, \boldsymbol{A}\boldsymbol{B}_1, \boldsymbol{A}^2\boldsymbol{B}_1, \boldsymbol{A}^3\boldsymbol{B}_1]$.
- Convert the continuous time system to discrete time.

$$\boldsymbol{x}(\boldsymbol{k}+\mathbf{1}) = \boldsymbol{A}_d\boldsymbol{x}(\boldsymbol{k}) + \boldsymbol{B}_{1d}\delta(k) + \boldsymbol{B}_{2d} r_{des}(k)$$

- Use the full state feedback law: $\delta = -\boldsymbol{K}\boldsymbol{x} = -k_1 e_{cg} - k_2 \dot{e}_{cg} - k_3\, e_\theta - k_4\, \dot{e}_\theta$.

Trajectory tracking with LQR

Apply LQR in this situation, we have

$$\delta^*(k) = -\boldsymbol{K}\boldsymbol{x}(k)$$

Where $\boldsymbol{K} = (R + \boldsymbol{B}_d^T\boldsymbol{P}\boldsymbol{B}_d)^{-1}\ \boldsymbol{B}_d^T\boldsymbol{P}\boldsymbol{A}_d$.

Objective cost function to be minimized by the control is

$$J = \sum_{k=0}^{\infty} \boldsymbol{x}^T(k)\boldsymbol{Q}\boldsymbol{x}(k) + \delta(k)R\delta(k)$$

其中p满足矩阵差分黎卡提方程

where $\boldsymbol{P}$ satisfies the matrix difference Riccati equation

$$\boldsymbol{P} = \boldsymbol{A}_d^T\boldsymbol{P}\boldsymbol{A}_d - \boldsymbol{A}_d^T\boldsymbol{P}\boldsymbol{B}_d(R + \boldsymbol{B}_d^T\boldsymbol{P}\boldsymbol{B}_d)^{-1}\ \boldsymbol{B}_d^T\boldsymbol{P}\boldsymbol{A}_d + \boldsymbol{Q}$$

- $\boldsymbol{Q}$ is a diagonal weighting matrix with an entry for each state corresponding to the performance aspects contributing to the cost function and $\boldsymbol{R}$ is weighting value corresponding to the control effort contributing to the cost function.

是一个对角加权矩阵，每个状态对应于对成本函数有贡献的性能方面，而 是对应于对成本函数有贡献的控制努力的加权值。

LQR summary

- The Linear Quadratic Regulator (LQR) method can be used with the dynamic bicycle model to design a path tracker. The model and resulting control law are easy to understand and implement.
- Tuning the LQR tracker is more complicated because the solution to the optimal control problem must be solved to obtain the gains.
- Curvy road performance is not that good. The dynamic bicycle model approximates the lateral dynamics of the vehicle, but in order to enable the use of linear control techniques the path coordinate model is linearized about the forward direction. In other words, the model excludes the non-linear path dynamics and best approximates a vehicle following a straight path.

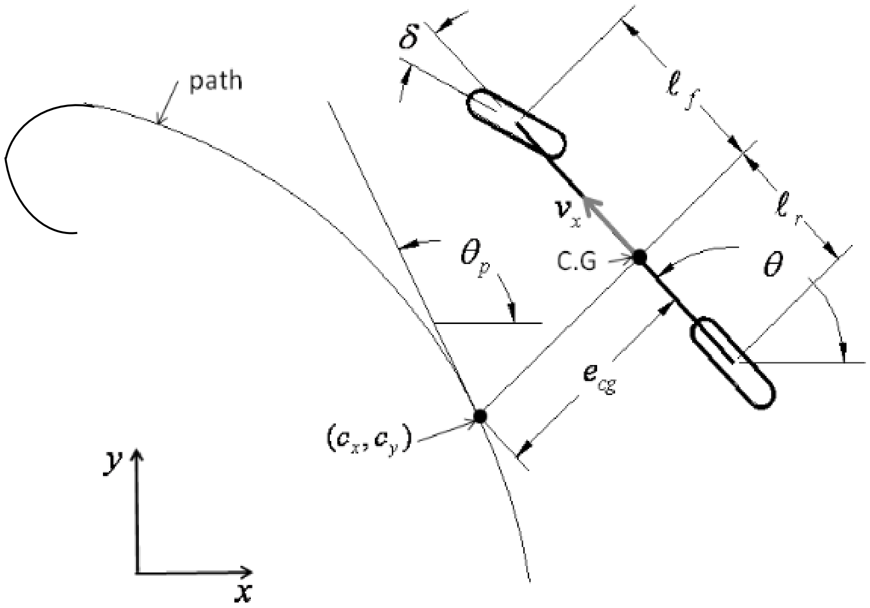
Trajectory tracking with Preview Control

Trajectory tracking with Preview Control

Why preview control?

- What if the path looks like this: a U-turn, can the previously designed LQR handle it?
- How would a human drive handle U-turn?
  - ✓ Starts to prepare before reaching the U-turn
- Why not use all available information on path but only look at "current"?
- "Best" utilizing all existing knowledge: a preview of the path ahead of the vehicle.

利用所有现有知识的“ 最佳 ”:车辆前方路径的预览

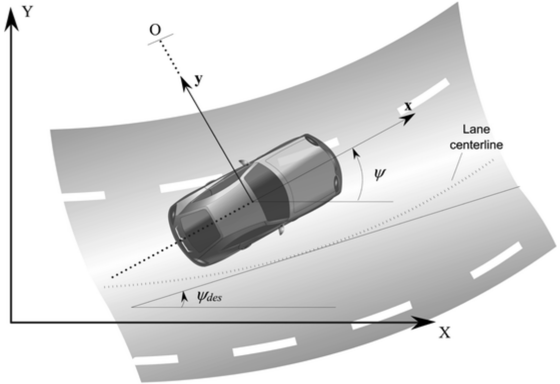


Dynamic Bicycle Model in path coordinates

Trajectory tracking with Preview Control

1. Translated vehicle lateral dynamics the state space to the discrete time-form:

$$\dot{\boldsymbol{x}}(\mathrm{t}) = \boldsymbol{A}_v \boldsymbol{x}(t) + \boldsymbol{B}_v \delta(t)$$
$$y(t) = \boldsymbol{C}_v \boldsymbol{x}(t) + \boldsymbol{D}_v \delta(t)$$
$$\Downarrow$$
$$\boldsymbol{x}(k+1) = \boldsymbol{A}_d \boldsymbol{x}(k) + \boldsymbol{B}_d \delta(k)$$
$$\mathrm{y}(k) = \boldsymbol{C}_d \boldsymbol{x}(k) + \boldsymbol{D}_d \delta(k)$$



Where $\boldsymbol{x}(\boldsymbol{k}) = (y(k) \quad \dot{y}(k) \quad \psi(k) \quad \dot{\psi}(k))^T$ and $\boldsymbol{A}_d, \boldsymbol{B}_d, \boldsymbol{C}_d, \boldsymbol{D}_d$ stand for the discretized vehicle model.

The lateral profile of the path is considered in discrete sample value form, with sample values from past observations of the path ahead being stored as states of the full vehicle/path system.

道路的横向轮廓被认为是离散样本值形式，来自前方道路的过去观察的样本值被存储为整个车辆/道路系统的状态

Trajectory tracking with Preview Control

2. The Reference Path Preview Model:

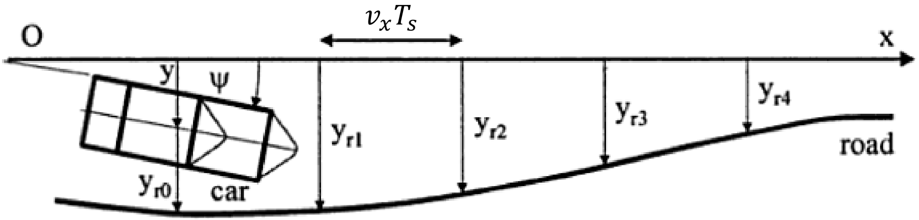
当车辆在时间上向前移动时，先前的预瞄参考将被丢弃，并且可以在每个采样时间基于路径规划模块获得一组新的预瞄参考

As the vehicle moves forward in time, the previous preview reference will be discard and a new set of preview reference can be obtained based on path planning module at each sampling time.

The preview reference $\boldsymbol{y}_r$ can be found as:

$$\boldsymbol{y}_r(k+1) = \boldsymbol{A}_r \boldsymbol{y}_r(k) + \boldsymbol{B}_r \, y_{rN}$$

$$A_r = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}_{(N \times N)} \qquad B_r = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}_{(N \times 1)}$$



Current road angle =$(y_{r1} - y_{r0})/v_x T_s$

Trajectory tracking with Preview Control

增强模型:动态自行车模型和在线路径预览模型通过QP(二次规划)代价函数耦合。模型是:

3. Augmented model: the dynamic bicycle model and online path preview model is coupled by the QP (Quadratic Programming) cost function. The model is:

$$\widetilde{\boldsymbol{X}}(k+1) = \widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{X}}(k) + \widetilde{\boldsymbol{B}}\delta + \widetilde{\boldsymbol{B}}_r y_{rN}$$

$$\widetilde{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{A}_d & 0 \\ 0 & \boldsymbol{A}_r \end{bmatrix}, \widetilde{\boldsymbol{B}} = \begin{bmatrix} \boldsymbol{B}_d \\ 0 \end{bmatrix}, \widetilde{\boldsymbol{B}}_r = \begin{bmatrix} 0 \\ \boldsymbol{B}_r \end{bmatrix}$$

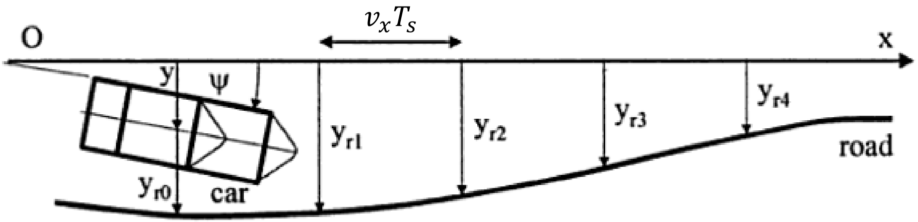
Where $\widetilde{\boldsymbol{X}} = [y \quad \dot{y} \quad \psi \quad \dot{\psi} \quad y_{r0} \quad y_{r1} \quad \dots \quad y_{rN}]^T$.

The cost function is:

$$J = \sum_{k=0}^{N} \widetilde{\boldsymbol{X}}^T(k)\boldsymbol{R}_1\widetilde{\boldsymbol{X}}(k) + R_2\delta^2(k)$$

Trajectory tracking with Preview Control

Use cost function to link the vehicle model and road model:



Current road angle $=(y_{r1} - y_{r0})/v_x T_s$

$$J = \sum_{k=0}^{N} \widetilde{\boldsymbol{X}}^T(k)\boldsymbol{R}_1\widetilde{\boldsymbol{X}}(k) + R_2\delta^2(k) \Longrightarrow \; J = \sum_{k=0}^{N} \widetilde{\boldsymbol{X}}^T(k)\boldsymbol{C}^T\boldsymbol{Q}\boldsymbol{C}\widetilde{\boldsymbol{X}}(k) + R_2\delta^2(k)$$

$$\boldsymbol{R}_1 = \boldsymbol{C}^T\boldsymbol{Q}\boldsymbol{C}; \;\; \boldsymbol{Q} = diag[q_1 \;\; q_2]; \; \boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \cdots 0 \\ 0 & 0 & 1 & 0 & \frac{1}{v_x T_s} & \frac{-1}{v_x T_s} & 0 & 0 \cdots 0 \end{bmatrix}$$

$$\boldsymbol{C}\widetilde{\boldsymbol{X}}(k) = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \cdots 0 \\ 0 & 0 & 1 & 0 & \frac{1}{v_x T_s} & \frac{-1}{v_x T_s} & 0 & 0 \cdots 0 \end{bmatrix} \begin{bmatrix} y(k) & \dot{y}(k) & \psi(k) & \dot{\psi}(k) & y_{r0}(k) & y_{r1}(k) & \ldots & y_{rN}(k) \end{bmatrix}^T$$

$$= \begin{bmatrix} y(k) - y_{r0}(k) \\ \psi(k) - \frac{y_{r1}(k) - y_{r0}(k)}{v_x T_s} \end{bmatrix} = \begin{bmatrix} e_y(k) \\ e_\varphi(k) \end{bmatrix}$$

Trajectory tracking with Preview Control

The steering angle will be

$$\delta_{opt} = -\boldsymbol{K}\widetilde{\boldsymbol{X}}(k),$$

where $\boldsymbol{K} = (R_2 + \widetilde{\boldsymbol{B}}^T\boldsymbol{P}\widetilde{\boldsymbol{B}})^{-1}\widetilde{\boldsymbol{B}}^T\boldsymbol{P}\widetilde{\boldsymbol{A}}$,

where $\boldsymbol{P}$ satisfies the discrete-time-Riccati equation (DARE),

$$\boldsymbol{P} = \widetilde{\boldsymbol{A}}^T\boldsymbol{P}\widetilde{\boldsymbol{A}} - \widetilde{\boldsymbol{A}}^T\boldsymbol{P}\widetilde{\boldsymbol{B}}\left(R + \widetilde{\boldsymbol{B}}^T\boldsymbol{P}\widetilde{\boldsymbol{B}}\right)^{-1}\widetilde{\boldsymbol{B}}^T\boldsymbol{P}\widetilde{\boldsymbol{A}} + \boldsymbol{C}^T\boldsymbol{Q}\boldsymbol{C}.$$

回忆一下连续时间代数黎卡提方程

Recall the continuous time Algebraic Riccati Equation (CARE/ARE),

Where $K = \boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P}$

and where $\boldsymbol{P}$ satisfies $\boldsymbol{0} = \boldsymbol{P}\boldsymbol{A} + \boldsymbol{A}^T\boldsymbol{P} - \boldsymbol{P}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P} + \boldsymbol{Q}.$

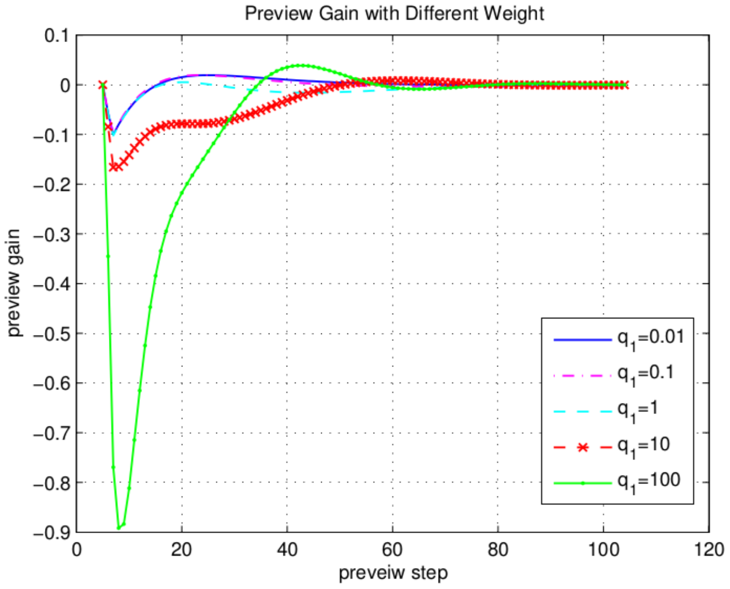
Trajectory tracking with Preview Control

The final problem convert to

$$J = \sum_{k=0}^{N} \widetilde{\boldsymbol{X}}^T(k)\boldsymbol{C}^T\boldsymbol{Q}\boldsymbol{C}\widetilde{\boldsymbol{X}}(k) + R_2\delta^2(k)$$

$$\boldsymbol{Q} = diag[q_1 \;\; q_2]$$

$$\boldsymbol{C}\widetilde{\boldsymbol{X}}(k) = \begin{bmatrix} e_y \\ e_\varphi \end{bmatrix}$$



where,

对应于横向参考跟踪性能的权重,

q_1 : the weights corresponding to the performance of lateral reference tracking,

q_2 : the weights corresponding to the performance of orientation reference tracking.

对应于角度参考跟踪性能的权重。

How to tune Preview Control in different scenarios

Cost function:

$$J = \sum_{k=0}^{N} \widetilde{\boldsymbol{X}}^T(k)\boldsymbol{C}^T\boldsymbol{Q}\boldsymbol{C}\widetilde{\boldsymbol{X}}(k) + R_2\delta^2(k)$$

$\boldsymbol{Q}$ is the weight on vehicle tracking error. It is used to avoid the positions that are not suitable for the vehicle. Therefore, if to track the preplanned reference is desired, the weight in $\boldsymbol{Q}$ will be heavier. 是车辆跟踪误差的权重。它用于避开不适合车辆的位置。因此，如果要跟踪预先计划好的

考， 的权重会更重。

Use case: For environments with strict constraints, such as static obstacle, narrow road with curb.

R_2 is the weight on control effort (steering angle). This may help the vehicle to balance the unexpected disturbance. Therefore, if it is required to be more robust, the weighted in R_2 will be heavier. 是操纵力(转向角)的权重。这可以帮助车辆平衡意外的干扰。因此，如果要求更稳健，在

权重会更重。

Use case: Highway driving.

Preview control summary

最优预瞄方法为LQR方法提供了对即将到来的路径的前瞻或预览

The Optimal Preview method provides the LQR method with a lookahead, or preview, of the upcoming path.

1. Work well in rapid change curvy road: The idea is that if a rapid change in curvature is known ahead of time the tracker can react earlier to minimize error. The proactive nature of the Preview method often sacrifices some error entering a curve to minimize the overall error through the curve.

在快速变化的曲线道路上工作良好:这个想法是，如果提前知道曲率的快速变化，跟踪器可以更早地做出反应，以最小化误差。预瞄方法的主动性质经常避免一些进入曲线的误差，以最小化通过曲线的整体误差。

2. This approach results in similar performance as Model Predictive Control (MPC) under certain circumstances, but using less computational resources. It is a little more complicated than the LQR method, but it is still easy to understand and implement.

这种方法在某些情况下产生与模型预测控制(MPC)相似的性能，但是使用更少的计算资源。它比LQR方法稍微复杂一点，但是仍然很容易理解和实现。

Different controller comparison

Different controller comparison

Controller name	Cross-track error (Overshoot)	Planner requirement	Robustness	Suitable Application
Pure Pursuit	High steady state error (high speed)	No	High	Slow driving; discontinuous path
Stanley Method	Better than PP but still when speed increase, error increase	Continuous	Mid	Smooth high speed; Parking
LQR	Large error in the curvy road, good tracking in straight road	Continuous	Low	High way drive
Preview Control	Good tracking performance with the preview information.	Low	Mid	High way and urban drive

深蓝学院
shenlanxueyuan.com

Thanks for Listening!