Documentazione Introduttiva Nuovo Task LQR

Filippo Gibertini

Gennaio 2025

Questo è un sistema di equazioni differenziali che rappresenta un modello a bicicletta:

$$\begin{bmatrix} \dot{e}_1 \\ \ddot{e}_1 \\ \dot{e}_2 \\ \ddot{e}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{2C_{\alpha f}+2C_{\alpha r}}{mV_x} & \frac{2C_{\alpha f}+2C_{\alpha r}}{m} & -\frac{2C_{\alpha f}\ell_f-2C_{\alpha r}\ell_r}{mV_x} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{2C_{\alpha f}\ell_f-2C_{\alpha r}\ell_r}{I_zV_x} & \frac{2C_{\alpha f}\ell_f-2C_{\alpha r}\ell_r}{I_z} & -\frac{2C_{\alpha f}\ell_f^2+2C_{\alpha r}\ell_r^2}{I_zV_x} \end{bmatrix} \begin{bmatrix} e_1 \\ \dot{e}_1 \\ e_2 \\ \dot{e}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2C_{\alpha f}}{m} \\ 0 \\ \frac{2C_{\alpha f}\ell_f}{I_z} \end{bmatrix} u$$

Noi utilizziamo questo modello per descrivere la nostra macchina.
Per il momento lasciamo perdere il significato delle singole variabili e concentriamoci sul significato del sistema.

La prima matrice 4×4 si chiama $\mathbf{A}$, il vettore colonna 4×1 si chiama $\mathbf{B}$. Il nostro vettore di stato (che noi chiameremo x) è:

$$\begin{bmatrix} e_1 \\ \dot{e}_1 \\ e_2 \\ \dot{e}_2 \end{bmatrix}$$

che sono rispettivamente:

- Deviazione laterale della macchina dalla traiettoria misurata in m
- Velocità di deviazione laterale in m/s
- Deviazione angolare dello yaw dalla traiettoria in rad
- Velocità di deviazione angolare dello yaw dalla traiettoria, "Angular"

La deviazione laterale è quanto si discosta la macchina dalla traiettoria che deve seguire, la deviazione angolare è quanta differenza angolare c'è fra la direzione della macchina e la direzione in cui dovrebbe essere sulla traiettoria.

In generale un sistema da controllare viene espresso come:

$$\dot{x} = Ax + Bu$$

dove u è l'input che possiamo dare noi al sistema che nel nostro caso specifico è **l'angolo di sterzo**.

Se so che input u dare al mio sistema in ogni singolo momento allora ho vinto. Ma come facciamo a calcolare u ?

$$u = -Kx = -k_1e_1 - k_2e_2 - k_3e_3 - k_4e_4$$

Dove K è la matrice (nel nostro caso specifico è un vettore) di controllo ottima per il nostro sistema. Possiamo quindi sostituire nel nostro sistema:

$$\dot{x} = Ax + (-KB)x \implies \dot{x} = (A - KB)x$$

LQR è una metodologia per trovare K ottimo dati A, B, Q, R.

Cosa sono Q e R? Q e R sono due matrici che servono a definire un problema di minimizzazione di una funzione di costo definita come:

$$J = \int_0^\infty (x^T Qx + u^T Ru)\, dt$$

Quindi Q serve a pesare il nostro vettore di stato x, in altre parole ci dice quanto è importante ogni componente di x. Facciamo un esempio:

$$Q = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0.5 \end{bmatrix}$$

Se questa è la nostra matrice Q allora stiamo dicendo che l'errore di posizione laterale ha un peso di 10, la velocità di deviazione laterale ha un peso di 1, l'errore angolare ha peso 5, la velocità di errore angolare ha peso 0.5. **La matrice Q va tunata**.

La matrice R invece dà un peso al nostro input u che nel nostro caso è solo uno scalare, quindi anche R sarà solo uno scalare. Puoi vederlo un po' come "quanto sforzo faccio a muovere il volante". **Anche R va tunato**.

Quindi a questo punto abbiamo A, B, Q, R. Grazie a MATLAB:

$$K = \texttt{lqr}(A, B, Q, R)$$

Ora che abbiamo K, il resto è semplice nel senso che per sapere che angolo di sterzo dobbiamo dare in ogni secondo basta fare $u = -Kx$ dove x è il nostro vettore di stato.

Problemi

Ci sono ben quattro problemi principali:

1. **Problema**: La matrice A dipende dalla velocità longitudinale (V_x), quindi a ogni velocità abbiamo una matrice A con valori diversi.

   Soluzione: Precalcoliamo tante matrici K offline, una ogni tot metri al secondo (es: ogni 5 m/s). A runtime scegliamo la matrice K relativa alla velocità più vicina.

2. **Problema**: La matrice A ha V_x al denominatore. Se arriviamo quasi a fermarci, questa metodologia si rompe.

   Soluzione: Sotto una certa velocità utilizziamo il pure pursuit e switchiamo online fra LQR e pure pursuit.

   Alternativa tiraculo: if $\{V_x < 1.0\}$: $V_x = 1.0$.

3. **Problema**: Calcolare il vettore di stato x da odometria e traiettoria. Questa è la parte che richiede più lavoro.

   Proposta: Implementare un algoritmo per passare dallo spazio globale allo spazio di Frenet:

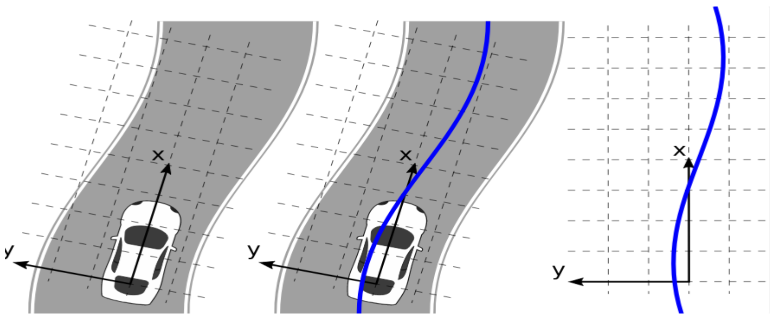

Figura 1: Spazio Globale

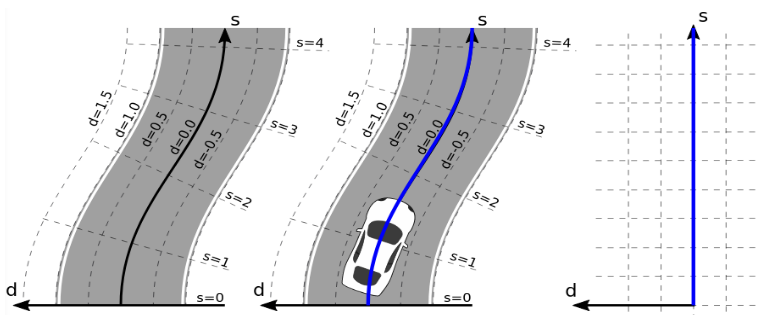

Figura 2: Spazio di Frenet

Lo spazio di Frenet significa semplicemente che tutte le coordinate vengono espresse nel sistema di coordinate della macchina e non nel sistema di riferimento della mappa.
So che questa parte non è facilissima da capire quindi fate domande a manetta, tenete a mente che è tutta roba molto standard quindi qualsiasi cosa che non vi è chiara la potete trovare anche su internet se non volete aspettare la risposta di un DL.

4. **Problema**: I parametri della matrice A sono difficili da stimare, ad esempio:

    - C_f, C_r (Rigidità laterale degli pneumatici):
        - C_f: Rigidità laterale del pneumatico anteriore.
        - C_r: Rigidità laterale del pneumatico posteriore.
    - m: Massa del veicolo.
    - V_x: Velocità longitudinale.
    - ℓ_f, ℓ_r: Distanze dal centro di gravità degli assi rispettivamente frontale e posteriore.
    - $L = \ell_f + \ell_r$: Lunghezza interassiale.
    - I_z: Momento d'inerzia.

    Soluzione: Interfacciarsi con la divisione dinamica per farci dire questi valori.

Dopo che abbiamo risolto brillantemente tutti questi problemi e abbiamo ottenuto $u = -Kx$, aggiungiamo un termine di feedforward, che significa semplicemente che aggiungiamo a u un termine basato su una predizione dell'angolo di sterzo ideale che dovremmo seguire:

$$u = -Kx - \delta_f$$

δ_f può essere calcolato come:

$$\delta_f = \frac{mV_x^2}{RL}\left[\frac{\ell_r}{2C_{\alpha f}} - \frac{\ell_f}{2C_{\alpha r}} + \frac{\ell_f}{2C_{\alpha r}}k_3\right] + \frac{L}{R} - \frac{\ell_r}{R}k_3$$

Dove R è il raggio di curvatura della traiettoria e k_3 è la terza componente del vettore K.

oppure come:

$$\delta_f = a \cdot \delta_m$$

dove δ_m è l'angolo di sterzo ideale calcolato con un modello dinamico sviluppato dalla divisione di dinamica che immagino ci darà un bel formulone da fare solo *""""plug and play""""*

Per chi si occuperà del task:

Ci tengo a ribadire che tutta questa roba è super mega standard quindi trovate tutta la documentazione e le spiegazioni che volete online: cos'è un modello a bicicletta, cos'è lo spazio di Frenet, cos'è il feed forward, ecc...

In particolare il modello a bicicletta è copiato dal libro Vehicle Dynamics and Control di Rajesh Rajamani

I paragrafi che vi interessano sono 2.2 , 2.3 ,2.5, 3.1, 3.2.

La formula che è scritta all'inizio della documentazione la trovate sul libro come **2.45**, potete notare che lui aggiunge un termine moltiplicato per $\dot{\psi}_{des}$ che sarebbe la velocità di deviazione angolare desiderata.
Ma noi non abbiamo una velocità di deviazione angolare desiderata, quindi questo temine viene aggiustato con il feedworward.

Lasciate sempre stare quando dice "road bank angle".

In ogni caso non è fondamentale che conosciate ogni singola formula a memoria però è importante che capiate i concetti.

Se vi interessa imparare di più su teoria dei controlli vi consiglio vivamente di guardare una serie di video su YouTube che si chiama **Control Bootcamp** in cui è spiegato molto bene come si controlla un sistema e come funziona LQR :

Control Bootcamp