1 Finding Jacobian Matrices

One of the calculations in our code that optimises our LQR gain matrices is to calculate the Jacobian matrices of our system. Let us start by stating our states :

$$x = \begin{cases} w \\ \mu \\ v_x \\ v_y \\ r \end{cases} \quad \dot{x} = \begin{cases} \dot{w} = v_x \cdot \sin(\mu) + v_y \cdot \cos(\mu) \\ \dot{\mu} = r - \frac{\kappa(v_x \cdot \cos(\mu) - v_y \cdot \sin(\mu))}{1-\kappa \cdot w} \\ \dot{v}_x = \frac{1}{M_L}(F_x - \texttt{Drag}) + r \cdot v_y \\ \dot{v}_y = \frac{1}{m}(F_{yf} \cdot \cos(\delta) + F_{yr}) - r \cdot v_x \\ \dot{r} = \frac{1}{I_{tot}}(F_{yf} \cdot \cos(\delta) \cdot L_f - F_{yr} \cdot L_r + M_{TV}) \end{cases} \quad y = \begin{cases} w \\ \mu \\ v_x \\ v_y \\ r \end{cases} \quad u = \begin{cases} DC \\ \delta \end{cases}$$

There are a few nested functions that we can define below :

$$\texttt{Drag} = F_{yf} \cdot \sin(\delta) + C_{drag} \cdot v_x^2 + \texttt{Car.DT.Friction} \cdot \arctan(v_x) \tag{1}$$

$$F_x = 2.06 \cdot 9.81 \cdot (m + \texttt{Car.DT.Theta}) \cdot DC \tag{2}$$

$$\alpha_f = \arctan\left(\frac{v_y + L_f \cdot r}{v_x}\right) - \delta \qquad \alpha_r = \arctan\left(\frac{v_y - L_r \cdot r}{v_x}\right) \tag{3}$$

$$\mu_y(\alpha_i) = D \cdot \sin\{C \cdot \arctan[B \cdot (1-E) \cdot \alpha_i + E \cdot \arctan(B \cdot \alpha_i)]\} \quad \text{where } i \text{ is } f \text{ or } r \tag{4}$$

$$F_{zf} = \left(m \cdot g + C_{down} \cdot v_x^2\right) \cdot W_f \qquad F_{zr} = \left(m \cdot g + C_{down} \cdot v_x^2\right) \cdot W_r \tag{5}$$

$$F_{yf} = F_{zf} \cdot \mu_y(\alpha_f) \qquad F_{yr} = F_{zr} \cdot \mu_y(\alpha_r) \tag{6}$$

Now that we want to calculate the state space matrices of our system at certain time zones, let us start by showing their formulation.

$$A = \begin{pmatrix} \frac{\partial \dot{w}}{\partial w} & \frac{\partial \dot{w}}{\partial \mu} & \frac{\partial \dot{w}}{\partial v_x} & \frac{\partial \dot{w}}{\partial v_y} & \frac{\partial \dot{w}}{\partial r} \\ \frac{\partial \dot{\mu}}{\partial w} & \frac{\partial \dot{\mu}}{\partial \mu} & \frac{\partial \dot{\mu}}{\partial v_x} & \frac{\partial \dot{\mu}}{\partial v_y} & \frac{\partial \dot{\mu}}{\partial r} \\ \frac{\partial \dot{v}_x}{\partial w} & \frac{\partial \dot{v}_x}{\partial \mu} & \frac{\partial \dot{v}_x}{\partial v_x} & \frac{\partial \dot{v}_x}{\partial v_y} & \frac{\partial \dot{v}_x}{\partial r} \\ \frac{\partial \dot{v}_y}{\partial w} & \frac{\partial \dot{v}_y}{\partial \mu} & \frac{\partial \dot{v}_y}{\partial v_x} & \frac{\partial \dot{v}_y}{\partial v_y} & \frac{\partial \dot{v}_y}{\partial r} \\ \frac{\partial \dot{r}}{\partial w} & \frac{\partial \dot{r}}{\partial \mu} & \frac{\partial \dot{r}}{\partial v_x} & \frac{\partial \dot{r}}{\partial v_y} & \frac{\partial \dot{r}}{\partial r} \end{pmatrix} \quad B = \begin{pmatrix} \frac{\partial \dot{w}}{\partial DC} & \frac{\partial \dot{w}}{\partial \delta} \\ \frac{\partial \dot{\mu}}{\partial DC} & \frac{\partial \dot{\mu}}{\partial \delta} \\ \frac{\partial \dot{v}_x}{\partial DC} & \frac{\partial \dot{v}_x}{\partial \delta} \\ \frac{\partial \dot{v}_y}{\partial DC} & \frac{\partial \dot{v}_y}{\partial \delta} \\ \frac{\partial \dot{r}}{\partial DC} & \frac{\partial \dot{r}}{\partial \delta} \end{pmatrix} \quad C = I_5 \quad D = 0$$

Obviously, the most complex matrix to calculate will be the A and then B. We have developed the equations for each element of A below :

$$\frac{\partial \dot{w}}{\partial w} = 0$$
$$\frac{\partial \dot{w}}{\partial \mu} = v_x \cdot \cos(\mu) - v_y \cdot \sin(\mu)$$
$$\frac{\partial \dot{w}}{\partial v_x} = \sin(\mu)$$
$$\frac{\partial \dot{w}}{\partial v_y} = \cos(\mu)$$
$$\frac{\partial \dot{w}}{\partial r} = 0$$

$$\frac{\partial \dot{\mu}}{\partial w} = \frac{-\kappa^2 \cdot (v_x \cdot \text{cos}(\mu) - v_y \cdot \text{sin}(\mu))}{(\kappa \cdot w - 1)^2}$$
$$\frac{\partial \dot{\mu}}{\partial \mu} = \frac{-\kappa \cdot (v_y \cdot \text{cos}(\mu) + v_x \cdot \text{sin}(\mu))}{\kappa \cdot w - 1}$$
$$\frac{\partial \dot{\mu}}{\partial v_x} = \frac{\text{cos}(\mu) \cdot \kappa}{\kappa \cdot w - 1}$$
$$\frac{\partial \dot{\mu}}{\partial v_y} = \frac{\text{sin}(\mu) \cdot \kappa}{1 - \kappa \cdot w}$$
$$\frac{\partial \dot{\mu}}{\partial r} = 1$$

$$\frac{\partial \dot{v}_x}{\partial w} = 0$$
$$\frac{\partial \dot{v}_x}{\partial \mu} = 0$$
$$\frac{\partial \dot{v}_x}{\partial v_x} = \text{(function too long to write, see Matlab code for details)}$$
$$\frac{\partial \dot{v}_x}{\partial v_y} = \text{(function too long to write, see Matlab code for details)}$$
$$\frac{\partial \dot{v}_x}{\partial r} = \text{(function too long to write, see Matlab code for details)}$$

$$\frac{\partial \dot{v}_y}{\partial w} = 0$$
$$\frac{\partial \dot{v}_y}{\partial \mu} = 0$$
$$\frac{\partial \dot{v}_y}{\partial v_x} = \text{(function too long to write, see Matlab code for details)}$$
$$\frac{\partial \dot{v}_y}{\partial v_y} = \text{(function too long to write, see Matlab code for details)}$$
$$\frac{\partial \dot{v}_y}{\partial r} = \text{(function too long to write, see Matlab code for details)}$$

$$\frac{\partial \dot{r}}{\partial w} = 0$$
$$\frac{\partial \dot{r}}{\partial \mu} = 0$$
$$\frac{\partial \dot{r}}{\partial v_x} = \text{(function too long to write, see Matlab code for details)}$$
$$\frac{\partial \dot{r}}{\partial v_y} = \text{(function too long to write, see Matlab code for details)}$$
$$\frac{\partial \dot{r}}{\partial r} = \text{(function too long to write, see Matlab code for details)}$$

We have then done the same for the matrix B :

$$\frac{\partial \dot{w}}{\partial DC} = 0$$
$$\frac{\partial \dot{w}}{\partial \delta} = 0$$

$$\frac{\partial \dot{\mu}}{\partial DC} = 0$$
$$\frac{\partial \dot{\mu}}{\partial \delta} = 0$$

$$\frac{\partial \dot{v}_x}{\partial DC} = \frac{2 * 20.2086 \cdot m}{\texttt{Car.DT.Theta} + m}$$
$$\frac{\partial \dot{v}_x}{\partial \delta} = \text{(function too long to write, see Matlab code for details)}$$

$$\frac{\partial \dot{v}_y}{\partial DC} = 0$$
$$\frac{\partial \dot{v}_y}{\partial \delta} = \text{(function too long to write, see Matlab code for details)}$$

$$\frac{\partial \dot{r}}{\partial DC} = 0$$
$$\frac{\partial \dot{r}}{\partial \delta} = \text{(function too long to write, see Matlab code for details)}$$

Having the direct explicit equations in our code instead of making the algorithm calculate increases its efficiency by a substantial margin, along with decreasing its dependancy on other work packages in order to function. These calculations have been implemented within the function `LQR_Opt`. Now that we have the expressions for these matrices, we can easily calculate them in our code.

2 Discretising the System

The next step is to discretise the response of our state space system with a sampling time of T_S. In our algorithm, we choose to do a "zero-order hold" discretisation. Our algorithm will therefore output new matrices A' and B'. These two matrices can be calculated by doing the following :

$$A' = \text{exp}\left(A \cdot T_S\right) \tag{7}$$

$$B' = \int_0^{T_S} \text{exp}\left(A \cdot \tau\right) d\tau \cdot B = \frac{e^{A \cdot T_S} - 1}{A} \cdot B \tag{8}$$

3 Making the Final State Space System

Finally, we can make the final state space system with the matrices $\hat{A}$, $\hat{B}$, $\hat{C}$ and $\hat{D}$. These matrices are simply compositions of the matrices A, B, C and D that we previously calculated :

$$\hat{A} = \begin{pmatrix} A' & B' \\ 0 & I_2 \end{pmatrix} \qquad \hat{B} = \begin{pmatrix} B' \\ I_2 \end{pmatrix} \qquad \hat{C} = \begin{pmatrix} C & I_2 \end{pmatrix} \qquad \hat{D} = D$$

4 Calculating the Gain Matrix of our LQR Controller

Now that we have the matrices $\hat{A}$, $\hat{B}$, $\hat{C}$ and $\hat{D}$, we can move on to finding the gain matrix of our LQR controller. In our algorithm we solve for three different gain matrices, and then later on choose which to use. In order to calculate the gain matrix K_i, we will have to solve the Riccati equation. The matrices Q and R will be provided.

In our algorithm, we want to solve the continuous algebraic Riccati equation (CARE), whose equation is :

$$A^T X + XA - XBR^{-1}B^T X + Q = 0$$

And the control matrix is : $K = -R^{-1}B^T X$.

It is possible to find the solution to this by finding the eigen-decomposition of a larger system. We define the Hamiltonian matrix :

$$Z = \begin{pmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{pmatrix}$$

Our solution X can then be calculated :

$$X = U_2 U_1^{-1}$$

4.1 Gain Matrix K_1

4.2 Gain Matrix K_2

4.3 Gain Matrix K_3