

车辆模型控制

龙锴
University of Pittsburgh

I. 自行车运动学模型

自行车模型是一种常见的车辆动力学简化模型，其原理为将汽车的四个轮子通过两个等效车轮进行简化，从而简化车辆的整体模型分析。在自行车模型中，一般有如下假设：不考虑车辆的垂直运动，车辆为刚体，左右轮的角度和速度一致。于是车辆的状态量可以分为汽车的绝对坐标 (X, Y) 以及汽车的航向角度 φ ，车辆的输入量为后轮的速度 v_r 和前轮的偏转角度 δ_f 。

其整体的结构如图 1 所示

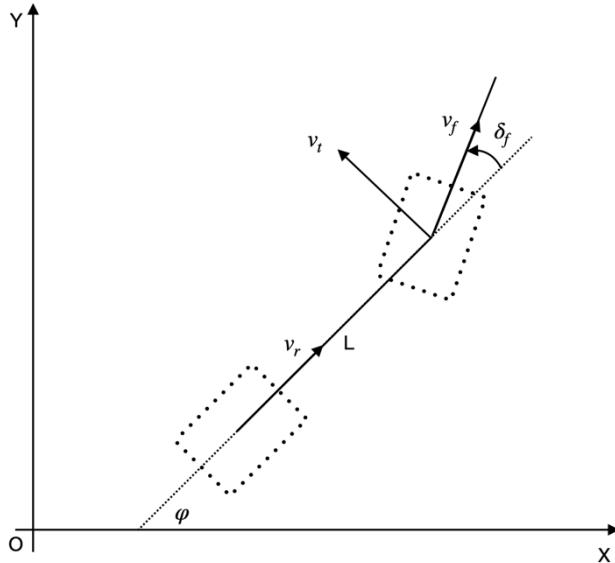


图 1 自行车模型

A. 模型构建

由力学的分析可知

$$v_r = v_f \cos \delta_f$$

$$v_t = v_f \sin \delta_f = v_r \tan \delta_f$$

$$\dot{x} = v_r \cos \varphi$$

$$\dot{y} = v_r \sin \varphi$$

$$\dot{\varphi} = \frac{v_t}{L} = \frac{v_r \tan \delta_f}{L}$$

其中 L 为车身的长度。由此可以得到模型的状态方程。

B. 模型的线性化

通过以上方程可以得到模型的状态空间方程。模型的状态为 $\dot{X} = \begin{bmatrix} x \\ y \\ \varphi \end{bmatrix}$ ，模型的输入为 $U = \begin{bmatrix} v_r \\ \delta_f \end{bmatrix}$ 。状态方程可以表示为 $\dot{X} = f(X, U)$ ，对其使用泰勒展开，可以得到

$$\dot{X} = \frac{\partial f}{\partial X} \tilde{X} + \frac{\partial f}{\partial U} \tilde{U}$$

其中 \tilde{X} 和 \tilde{U} 为参考量和当前量的差值，由此可以得到矩阵 A ：

$$A = \begin{bmatrix} 0 & 0 & -v_r \sin \varphi \\ 0 & 0 & v_r \cos \varphi \\ 0 & 0 & 0 \end{bmatrix}$$

矩阵 B ：

$$B = \begin{bmatrix} \cos \varphi & 0 \\ \sin \varphi & 0 \\ \frac{\tan \delta_f}{L} & \frac{v_r}{L \cos^2 \delta_f} \end{bmatrix}$$

因此自行车模型的线性化状态空间方程为

$$\dot{X} = \begin{bmatrix} 0 & 0 & -v_r \sin \varphi \\ 0 & 0 & v_r \cos \varphi \\ 0 & 0 & 0 \end{bmatrix} X + \begin{bmatrix} \cos \varphi & 0 \\ \sin \varphi & 0 \\ \frac{\tan \delta_f}{L} & \frac{v_r}{L \cos^2 \delta_f} \end{bmatrix} U$$

C. 模型的离散化

在自行车模型中，通常采用的离散化为前向欧拉法，其原理为

$$\frac{x(k+1) - x(k)}{T} = Ax(k) + Bu(k)$$

其中 T 为采样频率，通过转换可以变为

$$x(k+1) = (I + TA)x(k) + TBu(k)$$

于是可以得到离散化的模型为

$$\begin{aligned} X(k+1) &= \begin{bmatrix} 1 & 0 & -T v_r \sin \varphi \\ 0 & 1 & T v_r \cos \varphi \\ 0 & 0 & 1 \end{bmatrix} X(k) \\ &\quad + \begin{bmatrix} T \cos \varphi & 0 \\ T \sin \varphi & 0 \\ \frac{T \tan \delta_f}{L} & \frac{T v_r}{L \cos^2 \delta_f} \end{bmatrix} U(k) \end{aligned}$$

D. 模型的升阶

在实际的工程项目中，MPC 控制器无法直接控制车辆的速度和转角，需要通过控制油门的程度和方向盘转动的速度来变更车辆的速度和转角，因为为了更加贴合实际，将模型进行升阶。

G

$$= [g(0) \quad A(1)g(0) + g(1) \quad \dots \quad \sum_{i=0}^{N_p-1} (\prod_{j=i+1}^{N_p-1} A(j))g(i)]$$

首先将模型的状态从原本的 3 个状态 $[x, y, \varphi]^T$ 变为 5 个状态 $[x, y, \varphi, v_r, \delta_f]^T$, 其相对应的输入则变为 $[\alpha, \omega_\delta]^T$, 则状态空间方程中 A 矩阵为

$$A = \begin{bmatrix} 0 & 0 & -v_r \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & v_r \cos \varphi & \sin \varphi & 0 \\ 0 & 0 & 0 & \frac{\tan \delta_f}{L} & \frac{v_r}{L \cos^2 \delta_f} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

B 矩阵为

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}^T$$

由于原式为非线性系统, 则状态空间方程存在残差项, 则

$$g = f(x_r, u_r) - Ax_c + Bu_c$$

通过前向欧拉法得到离散化的 A 矩阵为

$$A(k+1) = \begin{bmatrix} 1 & 0 & -Tv_r \sin \varphi & T \cos \varphi & 0 \\ 0 & 1 & Tv_r \cos \varphi & T \sin \varphi & 0 \\ 0 & 0 & 1 & T \frac{\tan \delta_f}{L} & T \frac{v_r}{L \cos^2 \delta_f} \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

同理 B 矩阵为

$$B(k+1) = \begin{bmatrix} 0 & 0 & 0 & 0 & T \\ 0 & 0 & 0 & T & 0 \end{bmatrix}^T$$

残差项为

$$g(k+1) = Tg(k)$$

II. MPC COST 设计

A. MPC 状态

在 MPC 控制中, 模型会在预测区间累计展开, 则状态空间方程也会变为状态的累积, MPC 中的状态空间方程为 $X = Ox(k) + MU + G$

其中 X 为状态的累积, 其表达式为

$$X = [x(k+1) \quad x(k+2) \quad \dots \quad x(k+N_p)]^T$$

U 为输入的累积, 其表达式为

$$U = [u(k) \quad u(k+1) \quad \dots \quad u(k+N_c-1)]^T$$

O 为 A 矩阵的累积, 其表达式为

$$O = [A(0) \quad A(1) \quad \dots \quad \prod_{i=0}^{N_p-1} A(i)]^T$$

M 为 AB 矩阵的累积, 其表达式为

$$M = \begin{bmatrix} B(0) & 0 & \dots & 0 \\ A(1)B(0) & B(1) & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \prod_{i=0}^{N_p-1} A(i)B(0) & \dots & A(N_p-1)B(N_c-1) & B(N_c-1) \end{bmatrix}$$

G 为残差的累积, 其表达式为

B. Cost Function

在本 MPC 控制器中, Cost function 由四个部分组成, 其可以表示为

$$J = J_{state} + J_{control} + J_{terminal} + J_{comfort}$$

$$\text{其中 } J_{state} = \sum_{i=0}^{N_p-1} (x(i) - x_{ref}(i))^T Q (x(i) - x_{ref}(i))$$

$$J_{control} = \sum_{i=0}^{N_c-1} u(i)^T R u(i)$$

$$J_{terminal} = (x(N_p) - x_{ref}(N_p))^T P (x(N_p) - x_{ref}(N_p))$$

$$J_{comfort} = \sum_{i=0}^{N_c-1} (\Delta u_{k+i})^T S (\Delta u_{k+i})$$

为了方便求解, 简化代价函数

$$E = X - X_{ref} = (Ox_k + MU + G) - X_{ref} + D$$

D 为扰动项, 用来直接补偿模型偏差, 实现 Offset-free 跟踪控制。

$$\Delta U = D_1 U - E u_{k-1}$$

其中 D_1 为输入差分矩阵。将两个式子代入 Cost function 可以得到

$$\begin{aligned} H &= 2(M^T Q M + R + D_1^T S D_1) \\ F &= 2M^T Q(Ox_0 + G - X_{ref} + D) - 2R U_{ref} \\ &\quad - 2D_1^T S(E u_{k-1}) \end{aligned}$$

由于输入存在约束,

$$u_{min} \leq u(k) \leq u_{max}$$

则可采用 $V = [I, -I]^T, h = [u_{max}, -u_{min}]^T$,

最终得到 $\min_{u} \frac{1}{2} U^T H U + F^T U, GU \leq h$.

C. 延迟模型

车辆在实际工程中由于各传感器之间的交互存在延迟, 当前的状态不符合控制器中存有的状态, 于是需要考虑系统控制的延时。假设控制器中存在的延迟为 τ , Mpc 的采样周期为 T, 则延迟在 Mpc 控制器的步长可以表示为

$$d = \tau/T$$

则真实的离散系统为

$$x_{k+1} = f(x_k, u_{k-d})$$

和 Mpc 原理同理, 利用已知的离散化模型进行 d 步迭代, 可以得到

$$X = [x(k-d+1) \quad x(k-d+2) \quad \dots \quad x(k)]^T$$

$$U = [u(k-d) \quad u(k-d+1) \quad \dots \quad u(k)]^T$$

$$G = [g_{k-d} \quad A_{k-d+1}g_{k-d} + g_{k-d+1} \quad \dots \quad \sum A_{k-d}g_k]$$

矩阵 AB 和同理, 于是可以直接推断出当前车辆的状态为

$$x_k = AAx_{(k-d)} + BBu_{(k-d)} + gg$$

将当前得到的状态输入 MPC 控制器，对其进行预测从而得到合适的输入。

III. 扰动观测补偿

扰动观测补偿是一种先进的控制策略，用于估计和补偿系统中的外部扰动和内部动态变化了通过实施观测并预测扰动，从而提高系统改的鲁棒性和稳定性从而优化系统性能。

其原理为通过原本的状态空间方程 $X_{T+1}^{pred} = Ax + Bu$ 得到的 y 和经过延迟补偿后得到的 X_{T+1}^{meas} 进行比较得到 error，则真实的状态空间方程为 $X_{T+1} = Ax + B(u + d)$ ，在一般的 error 测量中通常含有高频噪声，因此需要对于误差进行过滤。在本项目中采用低通滤波来过滤噪声

$$distance = 0.995 \cdot distance + \alpha \cdot error$$

其中 α 为滤波器的相应系数。在 MPC 控制里面的计算这转变状态空间方程为

$$X = Ox + Mu + G + d$$

将其带入 MPC 控制器的计算即可得到优化后的输入。

代码中的具体实现为

```
arma::mat prevA = car.A_matrix(prevState(3),
prevState(2), prevState(4), T_mpc);
arma::mat prevB =
car.B_matrix(T_mpc);
arma::vec y_pred = prevA *
prevState + prevB * prevInput;
arma::vec y_meas =
compensated_state;
arma::vec error = y_meas -
y_pred;
disturbance = 0.995 *
disturbance + alpha * error;
disturbance =
arma::clamp(disturbance, -0.01, 0.01);
```

IV. 速度与 COST 权重

在不同工况下，系统的动态性能和控制需求不同。在转弯掉头时，汽车需降速速度，增大转弯角度，增加速度的权重，角度角度的权重，在直线路段上，车辆没有转弯的需求，所以增加车辆的角度权重。因此为了适应不同的车况，能够不断调中 COST 的权重就变得非常重要。在本项目中，此方案的实现具体方式为

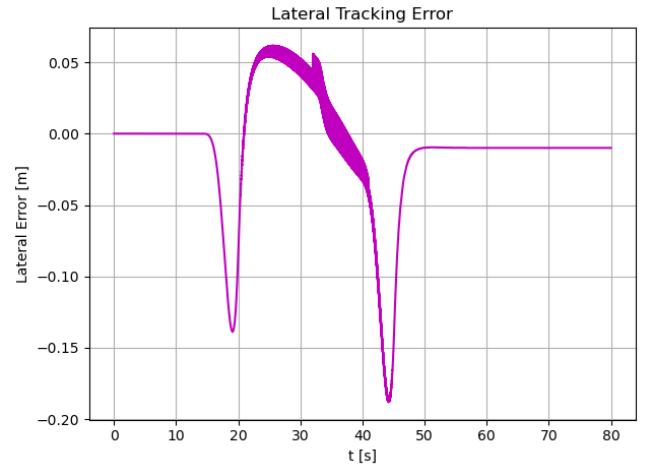
```
double v = car.get_v_r();
double v_low = speed_limit(1);
double v_high = speed_limit(0);
double w = (v - v_low)/(v_high -
v_low);
w = std::clamp(w, 0.0, 1.0);
double scale = 1.0 + 3.0 * w * w;
arma::mat Q_now = Q_base;
arma::mat R_now = R_base;
Q_now(2, 2) *= scale;
Q_now(4, 4) *= scale;
R_now(1, 1) *= scale;
mpc.setWeights(Q_now, R_now);
```

速度最多分为 4 个层次，实现了 MPC cost 权重随着速度的不同而调整，这里首先获得当前的车速，对其进行归一化处理，随后根据车速给予不同的权重，最后更新 MPC 控制器中的 COST 权重。

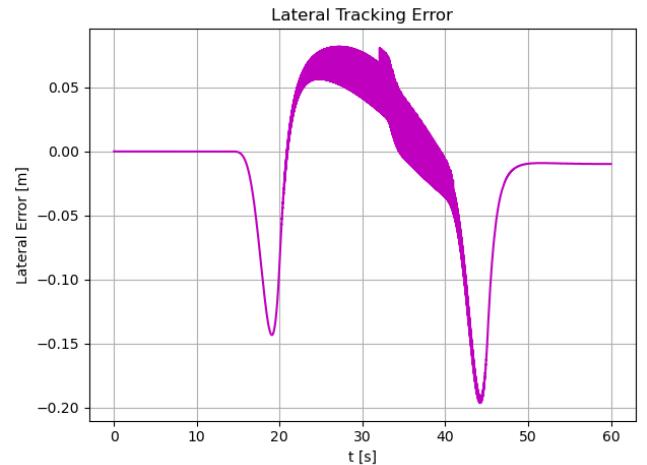
V. 采样频率

由于采样频率对于 MPC 控制器具有很大的影响，在此部分探究不同的采样频率对于 MPC 控制器的影响。此处采用横向误差作为评判标准。

当采样频率为 100Hz，保持固定的预测区间，系统的误差表现为

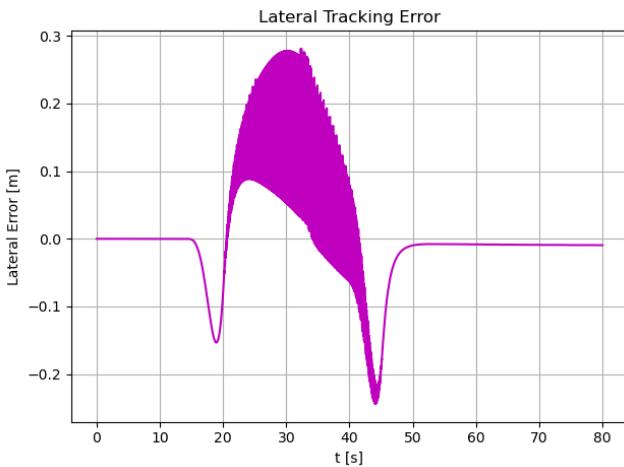


其最高误差为 0.18m，使用 RMSE 作为评估标准，整体的误差表现为 0.0045m，纵向误差为 0.1142m/s
当采样频率为 50Hz 时，系统的误差表现为



其横向最大误差为 0.19m，RMSE 横向表现为 0.0551m，纵向表现为 0.1364m/s

当采样频率为 10Hz 时，系统误差的表现为

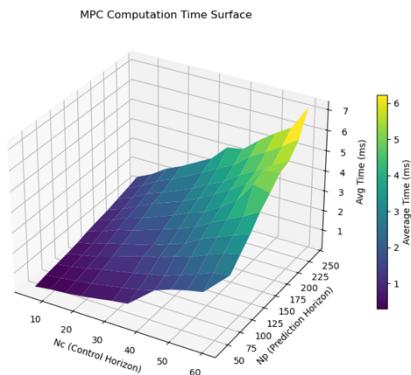


横向峰值误差为 0.29m, RMSE 横向误差为 0.0844m, 纵向误差为 0.1499m/s

综合分析随着 MPC 的控制频率不断上升, 模型整体的精度得到了显著的提高。

VI. 参数的影响

参数的大小会影响 MPC 的计算时间, 过大的参数会导致 MPC 无法在有效的时间内更新自己的参数。对计算时间影响最大的是 NP 和 NC 的大小。



如图所示, 计算的时间随着 NP, NC 的不断变大而上涨。

VII. 误差分析

由于车辆的假设是理想的无摩擦, 刚性完美, 但是在实际中, 轮胎打滑, 空气阻力变化等会让车辆运动偏离模型, 于是车辆在行驶过程中存在过程误差。

同时, 传感器在观测系统状态时, 由于精度有限, 干扰漂移, 采样误差等因素, 会导致观测的量与真实状态不一致, 所以存在着测量误差。因此噪声的实现如代码所示

```
class noiseGenerator {
public:
    noiseGenerator(const arma::vec& mean, const arma::vec&
    stddev) {
        : mean_(mean), stddev_(stddev), dist_(0.0, 1.0) {
        std::random_device rd;
        generator_.seed(rd());
    }
}
```

```
arma::vec sample() {
    arma::vec noise(mean_.n_elem);
    for (arma::uword i = 0; i < mean_.n_elem; ++i)
```

```
        noise(i) = mean_(i) + stddev_(i) * dist_(generator_);
        return noise;
    }
};
```

此段代码创建了一个高斯噪声发生器, 每次调用 sample 函数都会返回一个 $\omega_k \sim N(\mu, \tau^2)$ 的随机向量, 将此逻辑带入到仿真循环中, 可以得到

```
car.update_state(current_u(0), current_u(1), T_model);
arma::vec w_proc = process_noise.sample();
car.add_noise(w_proc);
```

从而让车辆的状态拥有噪声。对于测量的状态, 实现的逻辑为

```
arma::vec measured_state = state_history.front();
arma::vec v_meas = measurement_noise.sample();
noisy_measurement = measured_state + v_meas;
```

measure state 是理想传感器得到的状态, 通过对其增加随机测量误差, 因此可以得到测量误差。

VIII. 滤波器

由于车辆在实际运行过程中会受到各种噪声的影响, 为了减小这些噪声带来的误差, 活得更精准的系统状态信息, 需要在控制环节中使用滤波器。

滤波器的嘴勇士根据车辆的运动学模型和实时测量的数据, 对系统状态进行评估与修正, 从而在噪声的环境下得到尽可能接近真实的状态。常用的方法包括卡尔曼滤波, 它能在随机噪声的情况下融合预测值和测量值, 实现对系统状态的最优估计。

系统通常下的状态方程和测量方程表示为

$$X_{K+1} = A_K X_K + B_K u_k + w_k \\ z_k = H_k x_k = v_k$$

其中 A_k, B_k, H_k 分别为系统矩阵, 输入矩阵和车辆矩阵, w_k 表示过程噪声, 反应模型的不确定性, v_k 表示测量噪声, 用来反映传感器误差。

滤波的过程主要有两个主要步骤构成, 首先是预测阶段, 根据上一时刻的估计值, 预测当前时刻的状态和协方差:

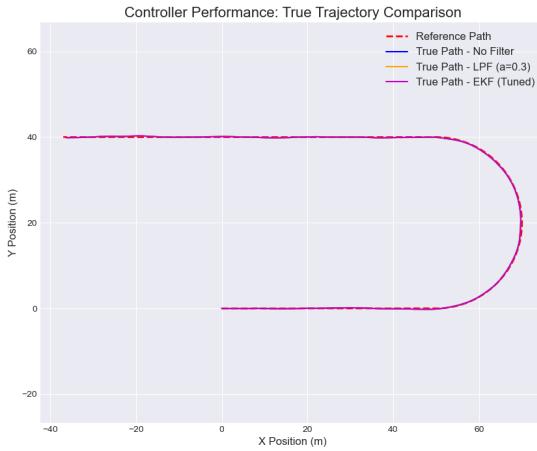
$$x_{k|k-1} = A_{k-1} x_{k-1} + B_{k-1} u_{k-1} \\ P_{k|k-1} = A_{k-1} P_{k-1|k-1} A_{k-1}^T Q$$

第二个阶段为更新阶段利用当前的车辆只对预测结果进行修正

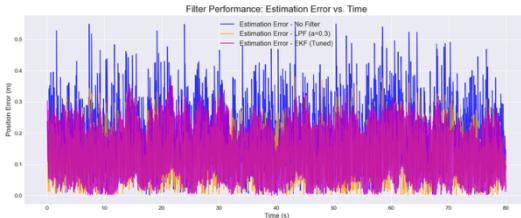
$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R)^{-1} \\ x_{k|k} = x_{k|k-1} + K_k (z_k - H_k x_{k|k-1}) \\ P_{k|k} = (1 - K_k H_k) P_{k|k-1}$$

其中 K_k 为卡尔曼增益, 决定预测预测两之间的均衡, 通过不断迭代上述过程, 卡尔曼滤波能在噪声干扰下获得系统状态的最优估计, 从而让车辆在存在过程误差和测量误差的情况下热能够实现平稳准确的状态跟踪。

如图所示,



尽管有噪声的存在，车辆仍能够紧贴参考轨迹，保持良好的状态。其整体的误差表现为

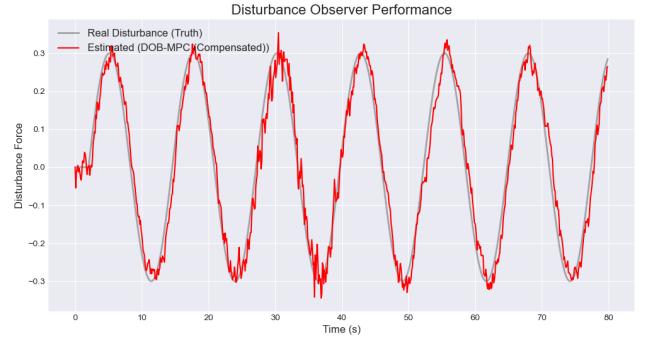


用 RMSE 来评价整体的预测误差表现

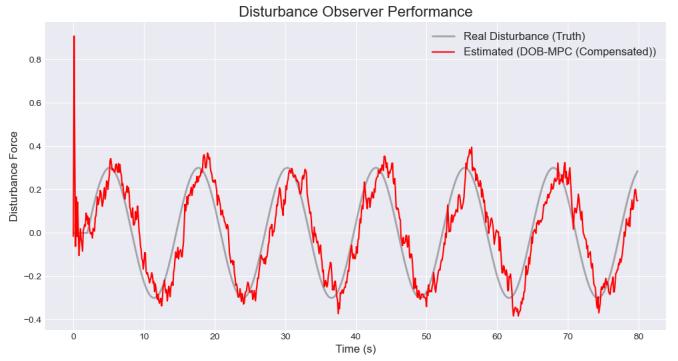
无滤波器下的表现为 0.2074 米，简单的低通滤波器 (LPF) 的误差为 0.1328 米，在卡尔曼滤波 (EKF) 下预测的误差为 0.1518 米。卡尔曼滤波相较于无滤波器提高了 26%，在简单滤波器下提高了 35%。猜测的原因是原函数中的延迟补偿模块使用的是非线性模型预测，LPF 能够直接利用这一精确模型进行外推，而 EKF 中使用雅可比方程线性化原方程然后推断状态，其内部引入的模型偏差在经过延迟补偿后降低了最终的估计精度。同时由于获取的是全状态，EKF 退化成为一个复杂的加权平衡器，导致无法发挥其推断的特性。因此，LPF 的预测精度及追踪精度比 EKF 要高。

IX. 干扰与观测器

在实际测量或任何动态系统中，由于模型的不完美，稳态干扰普遍存在。由于控制器使用的数学模型只能近似真实系统，像轮胎侧偏力，车体惯性误差，风力等外部因素长期作用在系统上，使系统输出在某一方向产生持续的偏移，因此在模拟上添加低频变化来模拟现实中的情况。在本项目中，采用 \sin 函数来模拟低频的变化。系统在受到低频和噪声的联合干扰后，模型的跟踪精度明显变差，无过滤器的跟踪精度由原本的 0.2746 米，变化为 0.4732 米，低通滤波器的精度由原本的 0.2522 米，变化为 0.4571 米，卡尔曼滤波的精度由原本的 0.2872 米，变化为 0.5682 米，其变化的幅度分别为 72%，81%，97%。由于低频扰动是一个外部影响，车辆本质上没有物理状态数量上的改变，因此对于卡尔曼滤波内部的状态进行升维，增加需要估计的数量，同时对 MPC 引入干扰项从而来降低车辆跟踪的误差。对于当几乎没有测量噪声时，如图所示



卡尔曼能够成功的推测低频干扰，观测器几乎没有滞后，通过观测幅值的大小，发现真实的干扰和推测的干扰的极值相同，观测器算出的力的大小是准的，虽然存在一些毛刺，但是极其灵敏，能够瞬时抓住干扰的变化。当测量噪声存在时，如图所示，



观测器反映的大小仍然相近，但是存在一定的滞后，并且整体的曲线更为的陡峭，存在更多的毛刺。其整体的最终精度通过 RMSE 计算为 0.3746 米，相较于完全没有采用观测器来说，精度提高了 34%，因此，基于增广卡尔曼滤波的干扰观测器在提升系统鲁棒性方面展现出了显著的效果。

X. 总结

本项目围绕自动驾驶车辆的路径跟踪控制展开，从最初的模型学习到完整的控制系统实现，构建了一套系统化的控制框架，首先，通过对于直行车辆模型，纵向动力学以及运动学模型的分析，建立了被控对象的连续，离散及线性化模型，为后续的控制器奠定了理论基础。在控制器设计方面，从最初的 PID 控制过度到 MPC 控制，实现了对于多种路径（直线，变道，掉头，转弯）的有效跟踪，并构建了输入约束，舒适性约束，终端约束的优化结构。为了增强系统的现实性，考虑了控制器延迟的现实问题，使得控制器更贴近真实车辆的动态特性。

在系统不确定性处理中，项目分别研究了噪声，滤波器与状态观测器对于精度控制的影响，通过使用比较无滤波器，低通滤波器和卡尔曼滤波器比较了不同场景下状态估计的质量，最终，通过引入扰动观测器对于低频扰动进行估计与补偿，降低了跟踪误差。

总体而言，本项目从模型建立，控制器实现到现实问题的处理（噪声，延迟，干扰，舒适度，跟踪精度），完成了一个较为完整的自动驾驶车辆路径跟踪控制系统，整个过程不仅实现了 MPC 控制器的功能验证，也系统

性的分析了影响控制精度的各种因素，完整了学习了不同场景下对于自动驾驶路径跟踪的挑战。

XI. 致谢

在本项目的学习与推进过程中，我得到了许多宝贵的指导与帮助。在此，我想向 Marvin 表达由衷的感谢。整个项目能够顺利完成，很大程度上得益于他在各个阶段提供的清晰思路与耐心指引。

特别是他为项目搭建的整体结构十分清晰、条理分明，使我在每个阶段都能专注于当前的任务而不至于迷失方向。无论是模型建立、控制器设计，还是观测器与干扰补偿部分，他都给予了关键性的建议与支持，让我对系统控制的理解更加全面、深入。

感谢他在讨论中提供的启发与鼓励，让我能够不断完善自己的方案并稳步推进任务。本项目的完成离不开他的帮助，在此致以最诚挚的谢意。

reference

- [1] Y. Zhang, M. J. Lyu, R. Z. Zhang, W. D. Huang, and M. Y. Qiao, “Research on vehicle lateral control based on model predictive control,” *Modeling and Simulation*, vol. 12, no. 3, pp. 3151–3161, May 2023, doi: 10.12677/mos.2023.123290
- [2] #君#, “MPC（模型预测控制）控制小车沿轨迹移动——C++ 实现,” *CSDN 博客* , Jan. 18, 2025. [Online]. Available: <https://blog.csdn.net/jiesunliu3215/article/details/144831194>. [Accessed: Oct. 28, 2025].