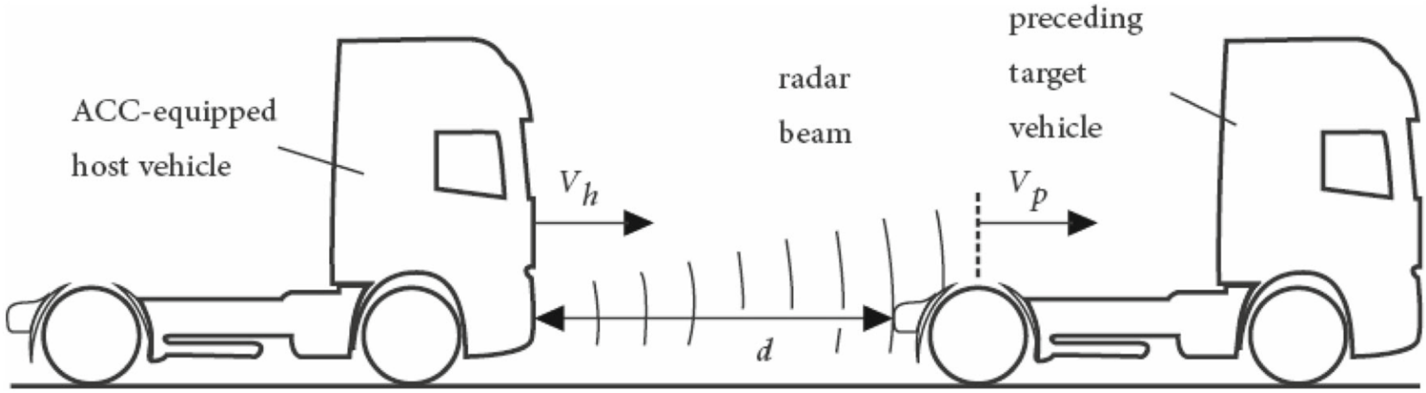
建模

模型图解如下:



参数一览表

符号	实际意义
δ_d	距离误差, Distance error
δ_v	速度误差, Velocity error
$\dot{v_h}$	自车加速度, Host vehicle acceleration
a_f	自车实际加速度, Host vehicle traction force converted to acceleration
a_{hmax}	自车加速度上限, Maximum acceleration
d	自车与前车之间距离, Inter-vehicle distance
d_0	停止后自车与前车距离, Stopping distance
d_r	期望的自车与前车距离, Desired distance
K_{eng}	发动机稳态增益, Steady-state gain
m	自车质量, Vehicle mass
r_{travel}	发动机到加速度输出过程中的阻力因素, Travel resistance
T_{eng}	发动机输出延迟, Time constant of acceleration using engine
T_{hw}	headway 时间常数, Constant time headway
u	输出即加速度, Control input command
v_h	自车速度, Host vehicle velocity
v_p	前车速度, Preceding vehicle velocity

基本的物理公式:

$\delta_v = v_p = v_h$

$\delta_d = d - d_r$

$d_r = T_{hw} v_h + d_0$

$m\dot{v_h} = ma_f - r_{travel}$

本文忽略 r_{travel}, 因此 $m\dot{v_h} = ma_f$

$$\begin{gathered}\dot{v_h} = A_f(t) v_h + B_f(t) u \\ y = a_f = C_f(t)\dot{v_h}\end{gathered}$$

其中:

$A_f(t) = -\frac{1}{T_{eng}}, B_f(t) = -\frac{K_{eng}}{T_{eng}}, C_f(t) = 1$

第一个方程成立是将发动机到速度的过程考虑为一阶延迟系统.

状态模型

状态变量:

$$x = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T = \begin{bmatrix} \delta_d & \delta_v & \dot{v_h} \end{bmatrix}^T$$

状态方程:

$$\begin{gathered}\dot{x} = Ax + Bu \\ y = cx\end{gathered}$$

本模型为线性定常系统.

系数矩阵为

$$A = \begin{bmatrix} 0 & 1 & -T_{hw} \\ 0 & 0 & -1 \\ 0 & 0 & -\frac{1}{T_{eng}} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ -\frac{K_{eng}}{T_{eng}} \end{bmatrix}$$

采用二次型的性能泛函(cost function):

$$\begin{aligned} & V_N(x_0, u) = \sum_{k=0}^{N-1} \{l(x(k), u(k)\} + V_f(x(N)) \\ & \text{s.t. } u \in \mathbb{U}, x \in \mathbb{X} \end{aligned}$$

其中 l 与 V_f 的表示分别如下:

$$
\begin{aligned}
&l(x(k), u(k)) = x(k)^T Q x(k) + u(k)^T R u(k) \\
&V_f(x(N)) = x(N)^T \mathrm{Px}(N)
\end{aligned}
$$

N 为 time horizon.

将权重矩阵 Q 与 R 都设为1, 即 $Q = I, R = 1$.

P 是 Discrete Algebraic Riccati Equation (DARE) 的微分方程组的解.

$$
I = -PA - A^T P + PBB^T P
$$

注意 A 不是满秩, 不能使用 A^{-1} 求解.

本系统的稳态也是期望的最终状态为原点 (origin), 即 $x_e = [0 \quad 0 \quad 0]^T$. 也就是相对前车速度为 0, 距离为期望的距离, 加速度为 0.

离散化

$$
\begin{aligned}
&x_{t+1} = \Pi x_t + \Gamma u_t \\
&y_t = C_d x_t \\
&\Pi = e^{AT_s} \\
&\Gamma = \int_0^{T_t} e^{At} dt \cdot B \\
&C_d = C
\end{aligned}
$$

其中 T_s 为采样时间间隔.

对 Π 的求解可以通过如下常规流程求解:

1. 求解 A 的特征值: $|\lambda I - A| = 0$ 求出 $\lambda_1, \lambda_2, ...\lambda_n$. 本例出现重根:$\lambda_1 = \lambda_2 = 0, \lambda_3 = -\frac{1}{T_{eng}}$.
2. 求解特征向量 P_1, P_2, P_3.
   $AP_1 = \lambda_1 P_1, AP_3 = \lambda_3 P_3$. 因为有重根, 所以 $\lambda_1 P_2 - AP_2 = -P_1$.
3. 求出变换矩阵 $T = [P_1 \ P_2 \ P_3]$ 及其逆矩阵 T^{-1}.
4. 通过 $e^{AT} = Te^{\Lambda t}T^{-1}$ 的形式求得 e^{AT}.
   即

$$
e^{At} = T \begin{pmatrix} e^{\lambda_1 t} & & & \\ & e^{\lambda_2 t} & & 0 \\ 0 & & \ddots & \\ & & & e^{\lambda_n t} \end{pmatrix} T^{-1}
$$

也可以利用泰勒展开忽略高阶项, 近似求解. 对 e^{AT_s} 进行泰勒展开可以得到:

$$\Pi = e^{AT_s} = I + AT_s + \frac{1}{2!}A^2T_s^2 + \cdots = \sum_{i=0}^{\infty}\frac{1}{i!}A^iT_s^i$$

仅取到2阶近似

$$A^2 = \begin{bmatrix} 0 & 0 & -1+\frac{T_{hw}}{T_{eng}} \\ 0 & 0 & \frac{1}{T_{eng}} \\ 0 & 0 & \frac{1}{T_{eng}^2} \end{bmatrix}$$

所以:

$$\Pi = I + AT_s + \frac{1}{2!}A^2T_s^2 =$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & T_s & -T_{hw}T_s \\ 0 & 0 & -T_s \\ 0 & 0 & \frac{-T}{T_{eng}} \end{bmatrix} + \begin{bmatrix} 0 & 0 & \frac{1}{2}(-T_s^2+\frac{T_{hw}T_s^2}{T_{eng}}) \\ 0 & 0 & \frac{1}{2}\frac{T_s^2}{T_{eng}} \\ 0 & 0 & \frac{1}{2}\frac{T_s^2}{T_{eng}^2} \end{bmatrix} =$$

$$\begin{bmatrix} 1 & T_s & \frac{1}{2}(-T_s^2+\frac{T_{hw}T_s^2}{T_{eng}}) - T_{hw}T_s \\ 0 & 1 & \frac{1}{2}\frac{T_s^2}{T_{eng}} - T_s \\ 0 & 0 & 1 - \frac{T}{T_{eng}} + \frac{1}{2}\frac{T_s^2}{T_{eng}^2} \end{bmatrix}$$

考虑到 $e^{\frac{-T_s}{T_{eng}}}$ 的二阶展开为:

$$e^{\frac{-T_s}{T_{eng}}} = 1 + \frac{-T_s}{T_{eng}} + \frac{1}{2}\frac{T_s^2}{T_{eng}^2}$$

可以凑出来 Π 如下, 这种形式与第一种常规做法求出的解是一致的.

$$\Pi = \begin{bmatrix} 1 & T_s & T_{eng}^2 - T_{eng}T_s - T_{hw}T_{eng} - T_{eng}^2e^{\frac{-T_s}{T_{eng}}} + T_{hw}T_{eng}e^{\frac{-T_s}{T_{eng}}} \\ 0 & 1 & T_{eng}e^{\frac{-T_s}{T_{eng}}} - T_{eng} \\ 0 & 0 & e^{\frac{-T_s}{T_{eng}}} \end{bmatrix}$$

利用上面求得的 Π 积分求得 Γ.

$$\Gamma = \int_0^{T_s} e^{At}dt \cdot B = \begin{bmatrix} \int_0^{T_s} 1dt & \int_0^{T_s} tdt & \int_0^{T_s} (T_{eng}^2 - T_{eng}t - T_{hw}T_{eng} - T_{eng}^2e^{\frac{-t}{T_{eng}}})dt \\ \int_0^{T_s} 0dt & \int_0^{T_s} 1dt & \int_0^{T_s} (T_{eng}e^{\frac{-t}{T_{eng}}} - T_{eng})dt \\ \int_0^{T_s} 0dt & \int_0^{T_s} 0dt & \int_0^{T_s} (e^{\frac{-t}{T_{eng}}})dt \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ \frac{1}{T_{eng}} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{-(\frac{T_{eng}T_s^2}{2} - T_{eng}^3(e^{\frac{-T}{T_{eng}}} - 1) - T_{eng}^2 T_s + T_{hw} T_{eng} T_s + T_{hw} T_{eng}^2 (e^{\frac{-T}{T_{eng}}} - 1))}{T_{eng}} \\ \frac{-(T_{eng}T_s + T_{eng}^2(e^{\frac{-T}{T_{eng}}} - 1))}{T_{eng}} \\ 1 - e^{\frac{-T}{T_{eng}}} \end{bmatrix}$$

求解

希望得到的最终结果是在 N 步下的控制序列 u. 然后根据其限值, 求得下一步的控制量 u_{next}.

离散的状态方程如下:

$$x(k+1) = \Pi x(k) + \Gamma u(k)$$
$$x(k)|_{k=0} = x(0)$$

差分的表达形式如下:

$$x(k) = \Pi^k x(0) + \sum_{j=0}^{k-1} \Pi^j \Gamma u(k-j-1)$$

更清晰一点, 使用矩阵形式:

$$\begin{pmatrix} x(1) \\ x(2) \\ x(3) \\ \vdots \\ x(k) \end{pmatrix} = \begin{pmatrix} \Pi \\ \Pi^2 \\ \Pi^3 \\ \vdots \\ \Pi^k \end{pmatrix} x(0)$$
$$+ \begin{pmatrix} \Gamma & 0 & 0 & \cdots & 0 \\ \Pi H & \Gamma & 0 & \cdots & 0 \\ \Pi^2\Gamma & \Pi H & \Gamma & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \Pi^{k-1}\Gamma & \Pi^{k-2}\Gamma & \Pi^{k-3}\Gamma & \cdots & \Gamma \end{pmatrix} \begin{pmatrix} u(0) \\ u(2) \\ \vdots \\ u(k-1) \end{pmatrix}$$

简写为标记:

$$X = \hat{A} x(0) + \hat{B} u$$

求解需要把性能泛函里的权重考虑进去, 先构建增广后的 Q, R 矩阵 Q_E, R_E.

$$R_E = \begin{pmatrix} R & & & \\ & R & & 0 \\ 0 & & \ddots & \\ & & & R \end{pmatrix}$$

考虑最终的状态为 x_e, 需要对最后一帧进行惩罚(如果最终状态不是 x_e 状态时), 也就是 Q_E 的最后一行需要加上上面 DARE 的解 P:

$$Q_E = \begin{pmatrix} Q & & & \\ & Q & & 0 \\ 0 & & \ddots & \\ & & & Q+P \end{pmatrix}$$

这一过程相当于把论文中的 P 与 Q 组合成了 Q_E.

根据离散系统的两点边值问题的求解原理可得:

$$u^* = -[\hat{B}^T Q_E \hat{B} + R_E]^T \hat{B}^T Q_E \hat{A} x(0)$$

控制量边界

考虑到延迟:

$$a_{lb} = \max(\frac{-T_s \cdot jerk}{1 - e^{\frac{-T_s}{T_{eng}}}} + a_{current}, a_{min})$$
$$a_{ub} = \min(\frac{T_s \cdot jerk}{1 - e^{\frac{-T_s}{T_{eng}}}} + a_{current}, a_{max})$$
$$u(1) = \max(a_{lb}, \min(u^*(1)), a_{ub})$$

最终输出控制量:

$$u_{next} = Ax_{current} + Bu(1)$$