

AEB系统控制算法（下）

本章对 AEB 系统的信号处理、安全距离、复杂场景的控制算法和执行逻辑进行了研究。

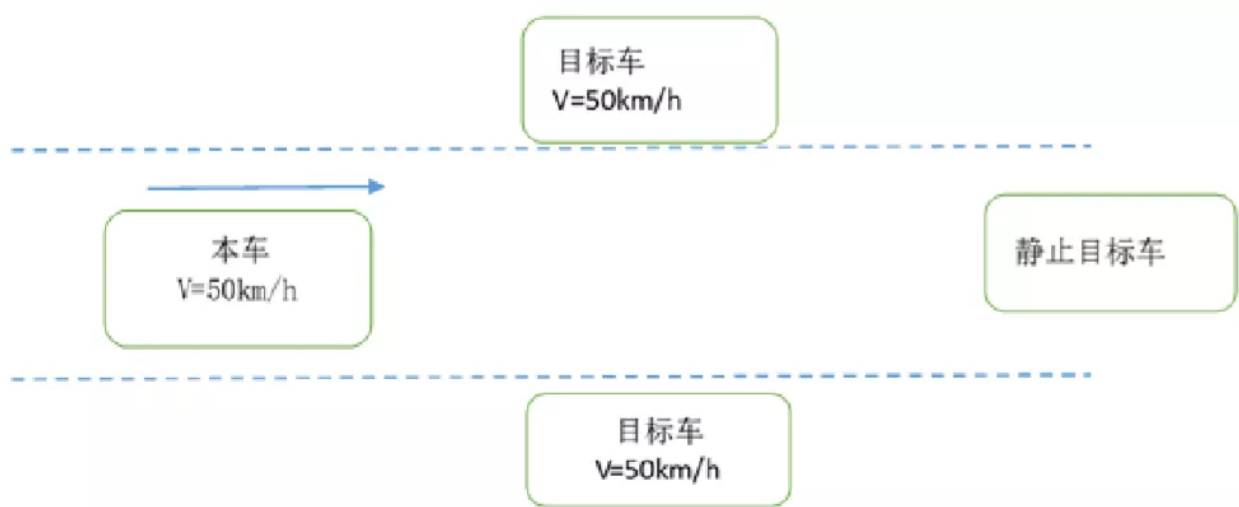
本文来源：牛咯网

复杂场景 AEB 系统控制算法

1. 多个目标车的 AEB 系统控制算法

在实际交通道路中，面对多个可能的潜在碰撞目标，因此测距传感器有必要具有 测量多个目标的能力，避撞算法也要相应能够处理多目标的潜在碰撞工况。

如图 4.11 所示，建立一条直线 3 车道的道路，本车处于中间车道，车速为 50km/h，车道上有三个目标车，和本车同一车道的目标车是静止状态，本车两侧的目标车速度都是 50km/h，三个目标车速度方向与本车相同。



4.11 多个目标车的 AEB 测试场景

在此交通场景下，AEB 系统能够实现多目标车的检测，并且能够避免目标车的 纵向碰撞危险 工况。

在 Simulink 中，多目标的检测信息以列向量的形式存储，因此在处理多目标检 测信息时，应 该将此列向量的值与一个列向量安全阈值进行大小对比，如果小于安全 阈值，则生成一个二 进制列向量标旗（flag），其中 0 表示标旗立起，1 表示旗标倒下。最终用或（OR）模块将 此列向量进行逻辑处理，如果其中有一个及以上立起的旗标，则输出立起的旗标，给出系统 判断当目标车距及车速会导致车辆发生碰撞的危险的信号。

2. 不同车道的车辆识别

在正常驾驶过程中，存在本车与目标车位于不同车道平行行驶的情况。这时候，即使两车的相对速度与相对距离达到危险情况的因素，系统也不应该给出制动的请求。

本车以 50km/h 的速度直线向前行驶，前方左右各两旁车道有一静止目标车，此测试场景意在验证车辆的 AEB 系统是否在不应该制动的情况下进行制动。

$$Xp=\sin \left(\theta \right) R ,$$
$$Yp=\cos \left(\theta \right) R .$$

式 (4-17)

式 (4-18)

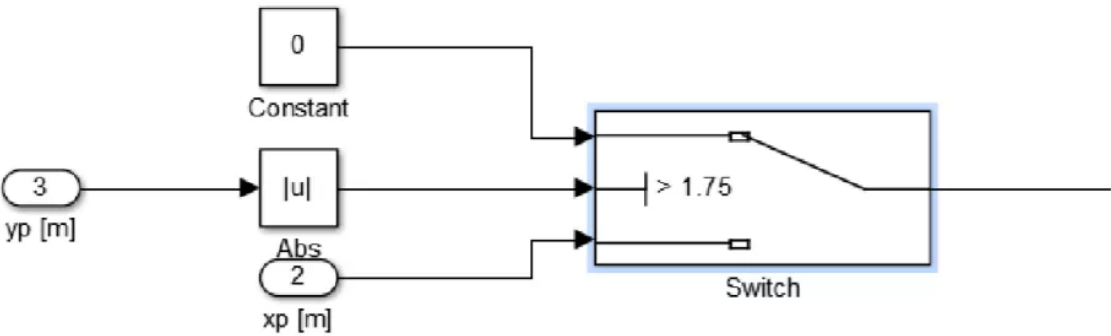


图 4.12 横向避撞场景算法

其中：Xp 表示纵向距离，Yp 表示横向距离，车道宽度为 3.5m，所以横向距离大于半个车道宽度时，可以判断目标车与本车位于不同车道。如果两车的横向距离保持不变，则表明两车平行行驶。

3. 驾驶员变道时 AEB 系统控制算法

当驾驶员有变道的意图时，这时候驾驶员已经有避免碰撞的意图，如果 AEB 系统进行制动动作，附着系数下降，侧向加速度变大，将会导致车辆的不稳定。因此在 驾驶员有变道的动作时，AEB 系统将不会起作用，否则会干扰正常驾驶。

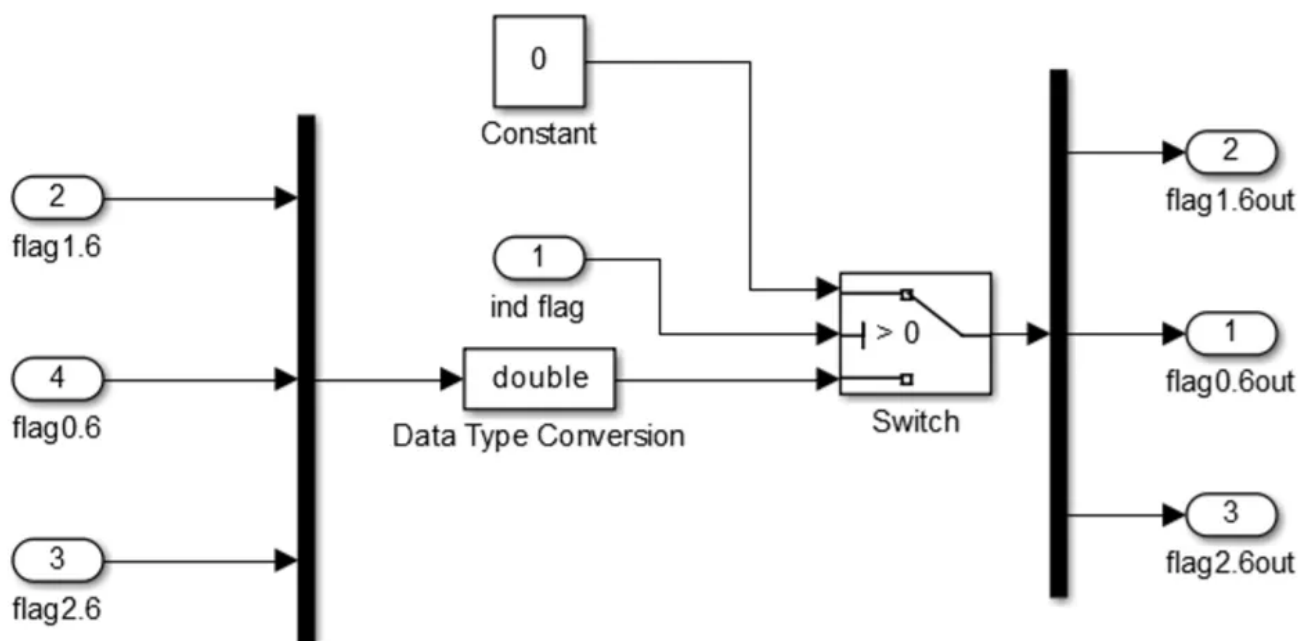


图 4.13 驾驶员变道 AEB 控制算法

因此在 Simulink 中建立一个 Switch 判断逻辑，如果驾驶员的转向角速度大于零，则输入的 ind flag 大于零，此时无论部分自动制动旗标（flag1.6）、全力制动旗标（flag 0.6）、警告旗标（2.6）为多少，输出对应的值则为零。

4. 弯道工况的 AEB 系统控制算法

车辆在弯道行驶过程中，需要判断目标车与本车是否处于同一个车道，用以过滤那些与本车不同车道的目标车。传感器输入信号只有本车的位置信号与 TIS 传感器的信号，在不添更多传感器的情况下，判断目标车与本车是否处于同一车道较为困难。

本文的思路是首先需要判断本车是否处于转弯状态，在本文中我们利用车辆的横摆速率（Yaw_rate）是否大于 1，来判断本车是否处于转弯的情况。利用假设目标车与本车都是做同心圆的车道轨迹运动，那么只需要计算出目标车和本车的运动轨迹的转弯半径之差，就可以判断两车是否在同一车道。如果两车的转弯半径之差在半个车道宽度（1.75m）之内，则判断本车与目标车处于一个车道，有发生碰撞的可能；否则本车与目标车没有碰撞的可能。

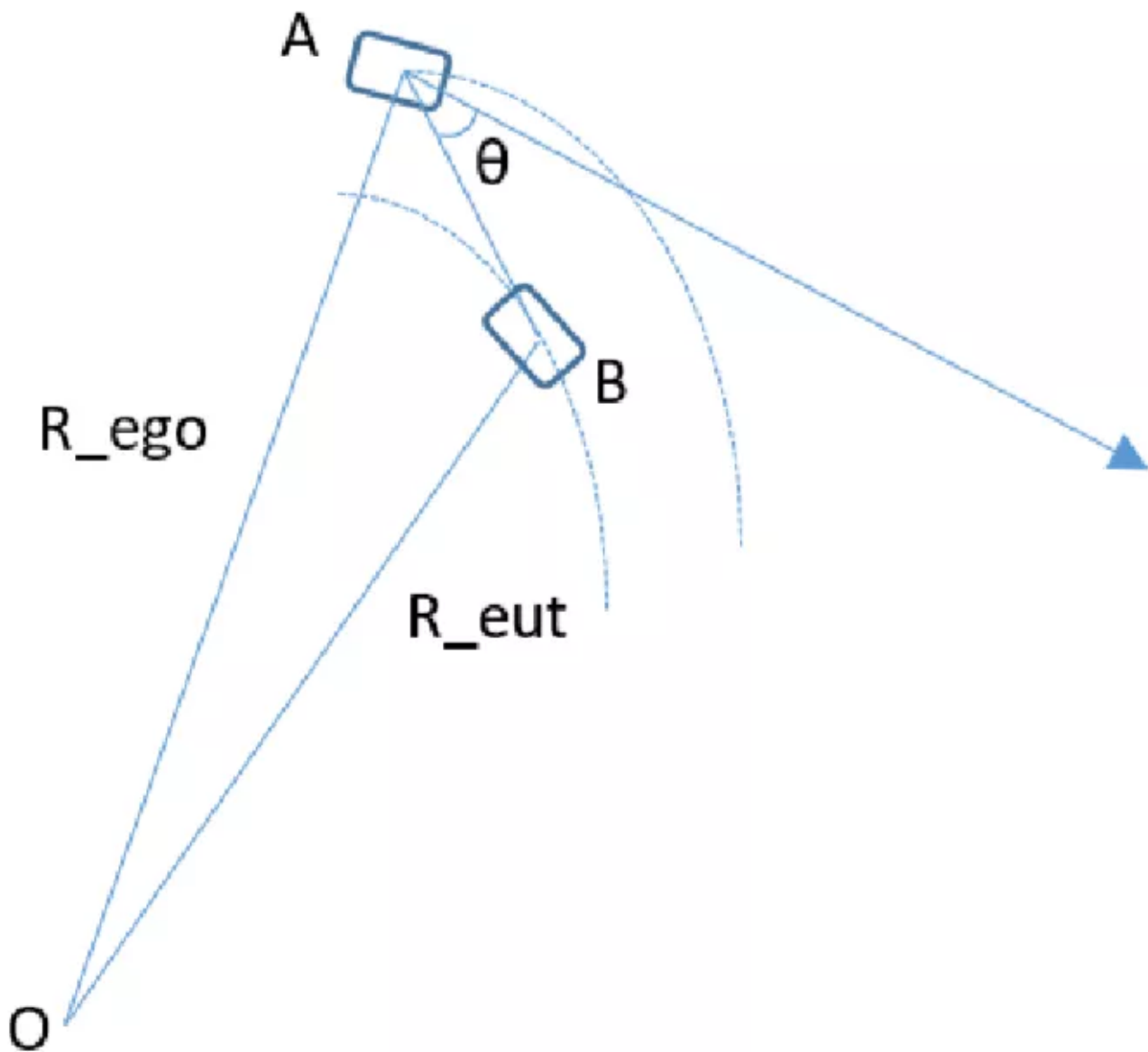


图 4.20 三支分叉路口工况两车运动模型

建立简化后的两车轨迹交叉运动数学模型如图 4.20 所示，在本车位于坐标系原点，目标车假设匀速沿着轨迹运动。只有当目标车与本车在同一时间到达轨迹交叉点 (x_0, y_0) 时，两车才会发生碰撞。所以本文的计划是利用 TIS 雷达测得的目标车辆的距离 R_2 与 θ_2 信号，计算出轨迹交叉点 (x_0, y_0) 的坐标，并且计算本车与轨迹交叉点 (x_0, y_0) 的距离 X_0 ，如果当前本车车速与 X_0 之间的关系超过了前文所讲的安全距离，则需要对本车进行预警或者自动制动的动作。

由雷达检测到的水平方位角 θ 和距离 R 数据可得：

$$y_1 = \sin\theta_1 R_1 \quad \text{式 (4-21)}$$

$$x_1 = \cos\theta_1 R_1 \quad \text{式 (4-22)}$$

$$v_{tgt_x} = (x_2 - x_1) / t_{sample} \quad \text{式 (4-23)}$$

式中， x_1 为目标车与本车的横向距离， y_1 为目标车与本车的纵向距离， v_{tgt_x} 为目标车相对于本车的纵向速度， t_{sample} 为本车传感器的数据采样时间。

目标车的运动线性方程为

当 $y=0$ 时，即目标车进入本车车道，

只有当本车和目标车同时到达交汇点 时,本车才需要进行制动。

当 与 相等时，两车有可能相撞；但是当 大于 时，代表目标车会先于本车穿过轨迹交叉点，两车不会碰撞；当 小于 时，代表本车会先于目标车穿过轨迹交叉点，两车也不会碰撞。

在 PreScan 软件中建立如图 4.21 所示的场景与传感器模型。建立一条 100m 长的五车道直线道路，本车位于中间车道，静止，安装如图所示的两个雷达传感器：TIS传感器 1 探测距离 30m，探测水平角度范围 80° ，探测垂直角度范围 9° ；传感器 2 探测距离 150m，探测水平角度范围 18° ，探测垂直角度范围 9° 。

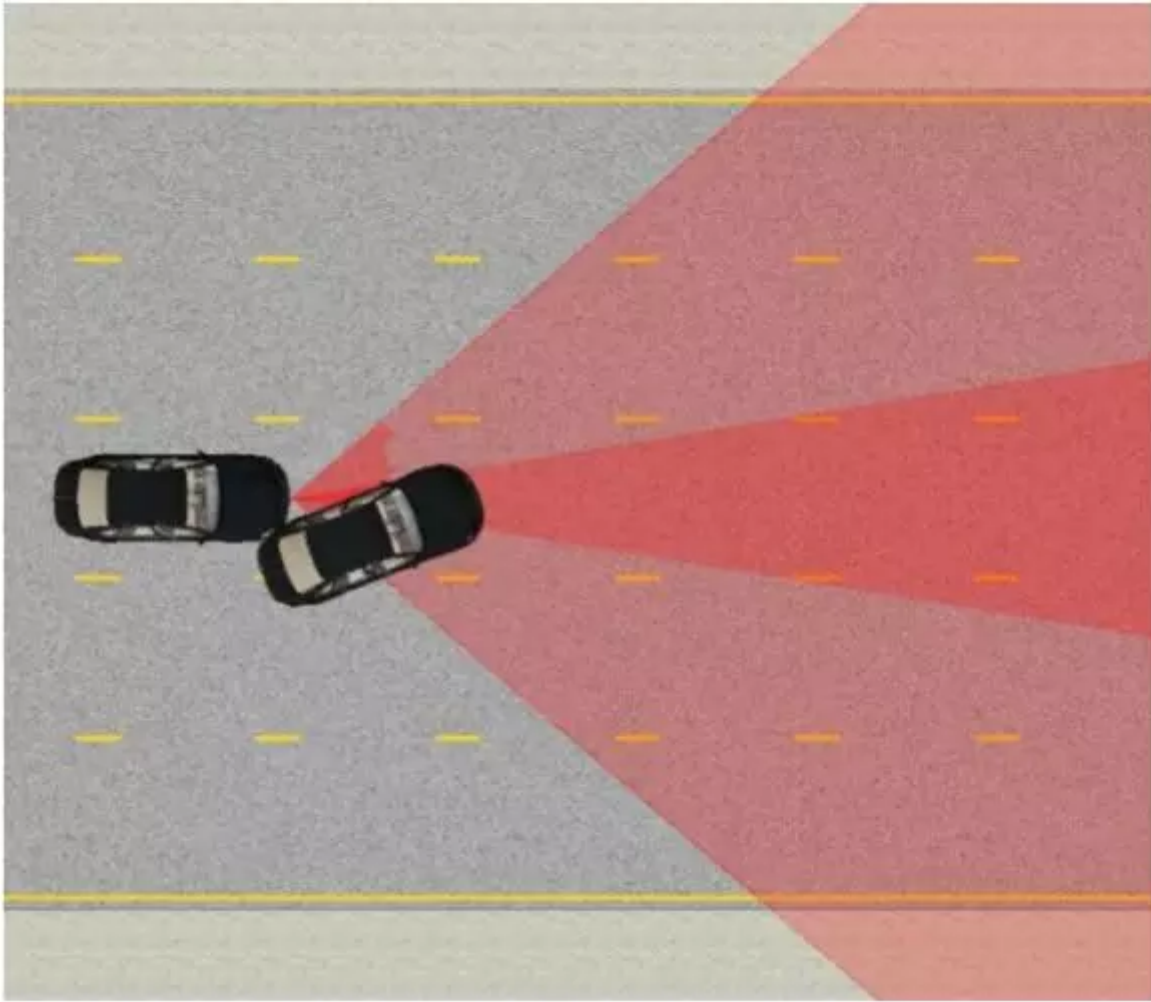


图 4.21 三支分叉路口工况轨迹交叉场景

在 Simulink 中运行的结果发现，TIS 测得的目标车由于车辆距离本车较近，而且 横向距离较大，以至于目标车在 TIS 测量范围之外，因此本文加大了短距离 TIS 雷达 的水平探测范围，更改后的短距离 TIS 雷达的水平探测范围是 160° 。更改后的 PreScan 模型如图 4.22 所示。

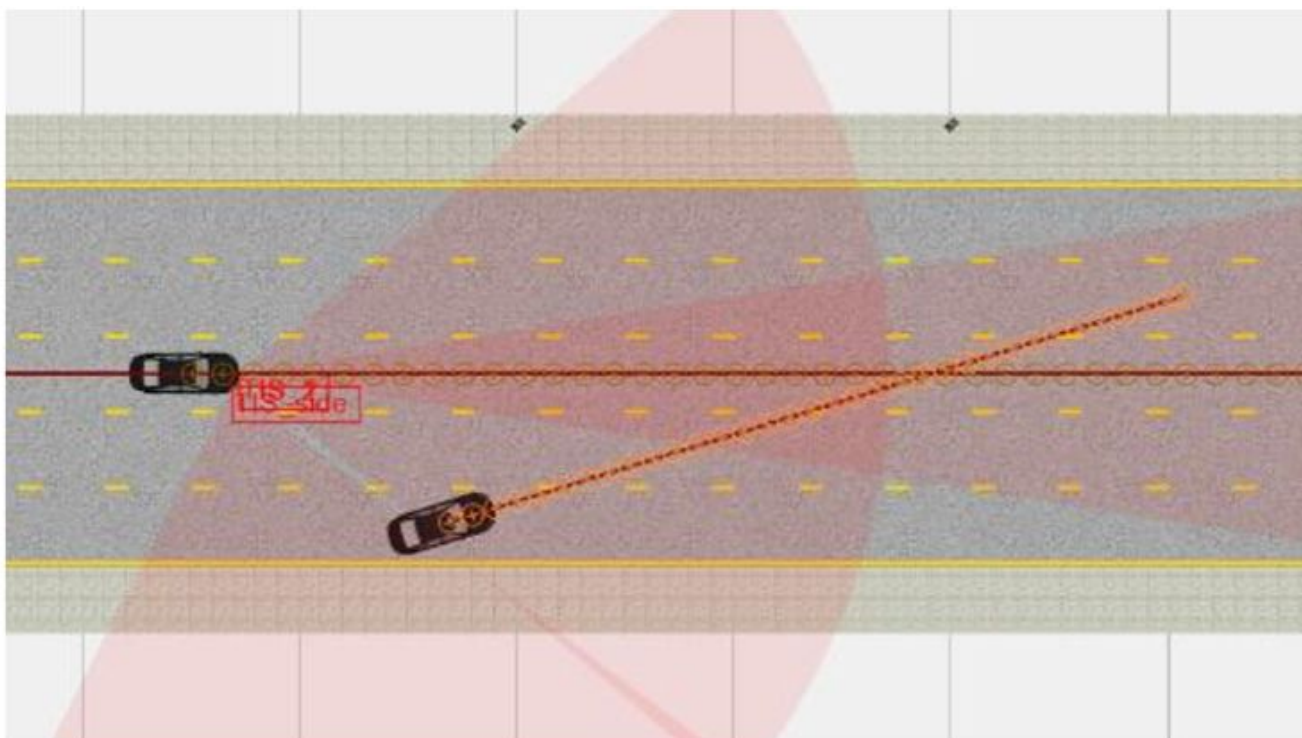


图 4.22 优化后的三支分叉路口工况轨迹交叉场景

根据式 4-21 至 4-27 建立的模型，运算后得到 X_0 点的数值， X_0 点的值如图 4.23 所示：

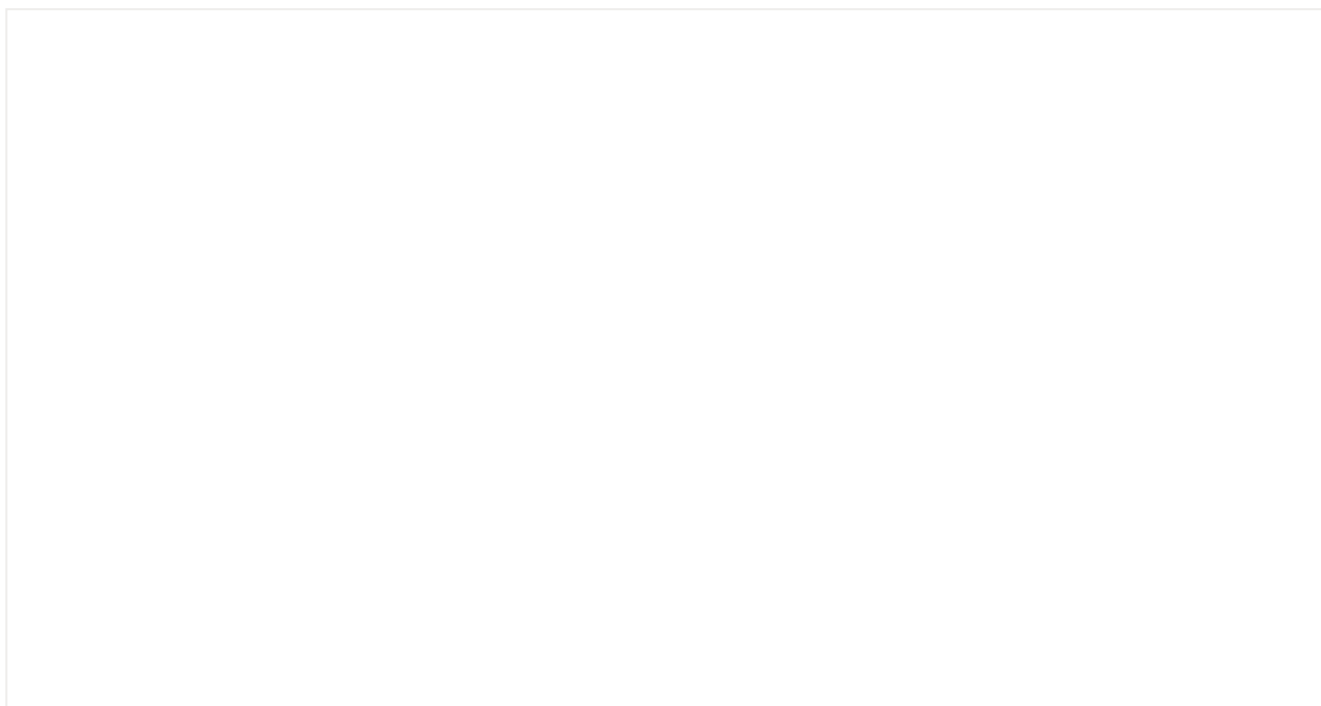


图 4.23 三支分叉路口工况模型计算的 x_0 值

由图 4.23 可见，点估计基本落在 30m 附近，这与实际测量到的 29.6m 误差较小，其误差主要原因是，雷达测量到的目标车距离是根据反射面信号最终确定的反射信号最强的点的位置，而这个点的位置随着车辆角度改变而变化，距离点并不是目标车的运动中心点。而上述所建数学模型是假设目标车是一个理想的质点。

上文提到在交通事故发生的道路类型中除了三支分叉路口，还有四支分叉路口的道路。在四支分叉路口中，两车的运动轨迹是相交垂直的，所以在 PreScan 软件中建立如图 4.24 所示的场景与传感器模型。建立一个长度为 50 米的四支分岔路，车道数为四车道，本车与目标车的位置处于相互垂直的关系。本车静止距离目标车的运动方向的垂直距离为 21m，目标车以 50km/h 的速度匀速直线行驶。



图 4.24 四支分叉路口模型

由此可以看到 Honda 模型有一个碰撞预警系统，而且 Honda 模型对的危险制动距离明显比 Mazda 模型更短。Honda 模型自动制动就介入更晚，这样的算法更加符合驾驶习惯，并且对驾驶员的正常驾驶的影响也更小。

最终得到四支分叉路口工况 点的值如图 4.25 所示：

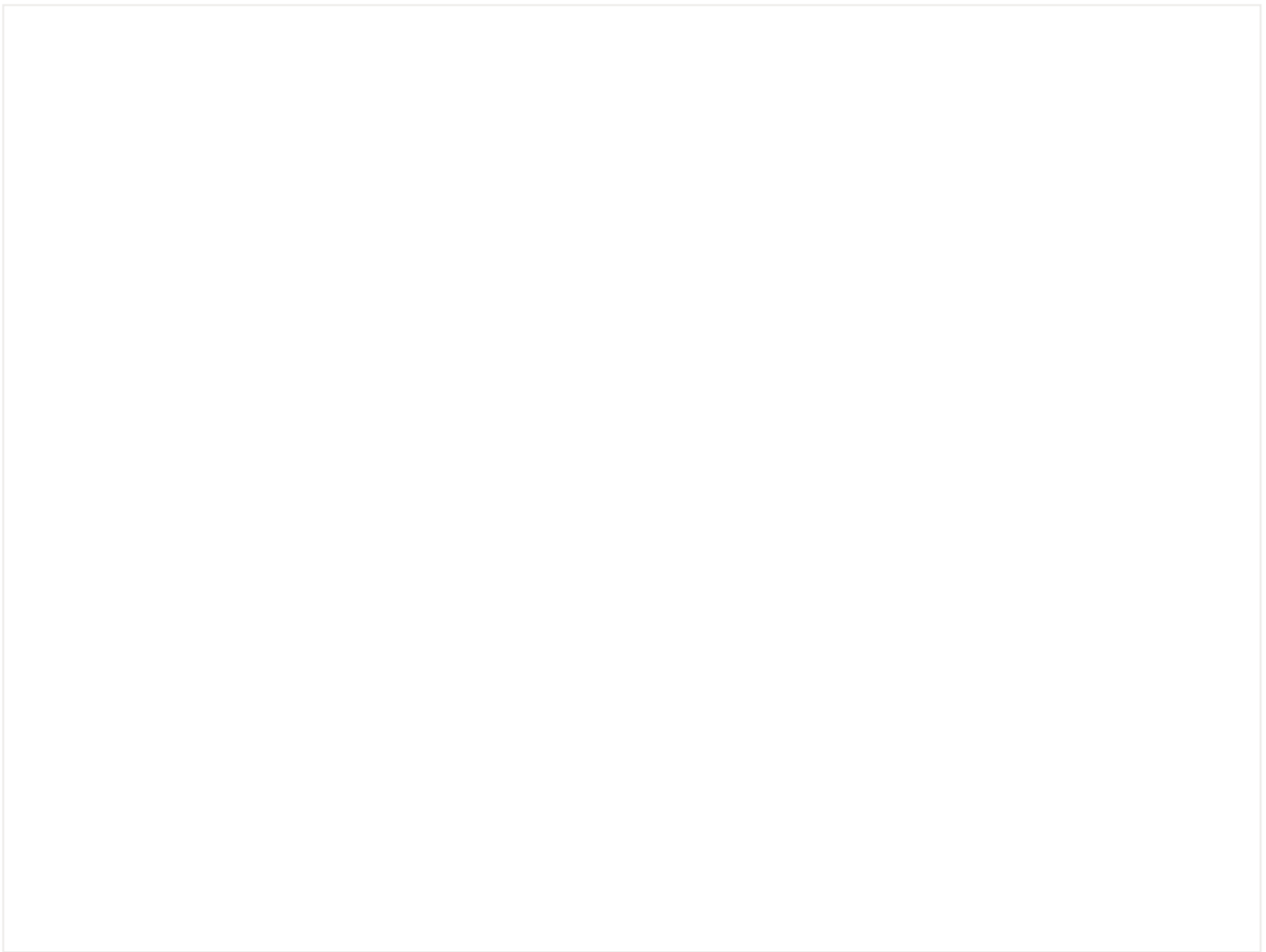


图 4.25 四支分岔路口的轨迹交叉点 x_0 值

由上图 4.25 可得，在此种工况下，点的计算值（21m）与实际将会碰撞的点（21.45-21.65m）估计基本一致。

下一步验证该工况的 AEB 系统控制算法，建立如图 4.26 所示的 PreScan 模型，本车与目标车都是以 50km/h 的速度匀速直线运动，其他的设置和前文的四支分岔路口的设置相同。



图 4.26 本车运动的四支分岔路口场景

运行结果可得计算的碰撞点 发生明显波动，并且 点的数值与实际即将发生 碰撞点的距离大小差异较大。 的计算值与仿真实际值的曲线如图 4.27 所示。

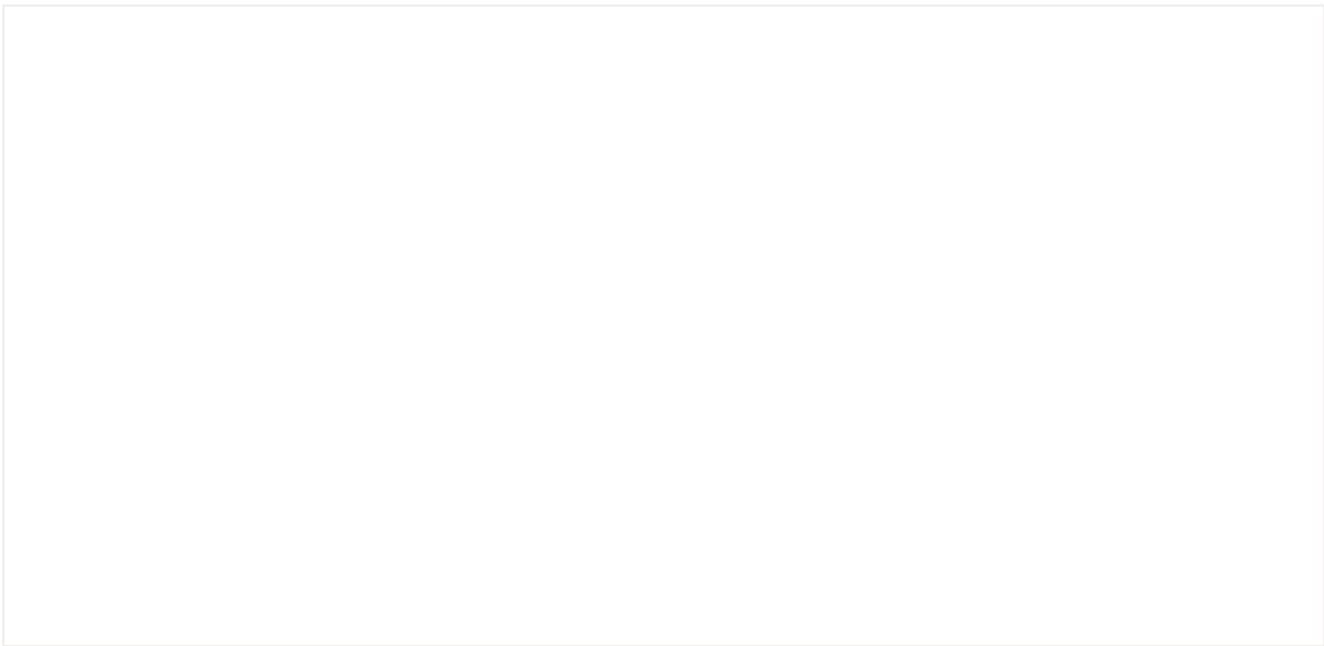


图 4.27 （ a ） x_0 的计算值 （ b ） x_0 仿真实际值

考虑其原因是运动数学模型没有考虑到本车也在发生运动，本文将车辆运动的数学模型中加入本车运动模式，并且也简化为直线匀速运动，这样本车只有 x 方向的运动。

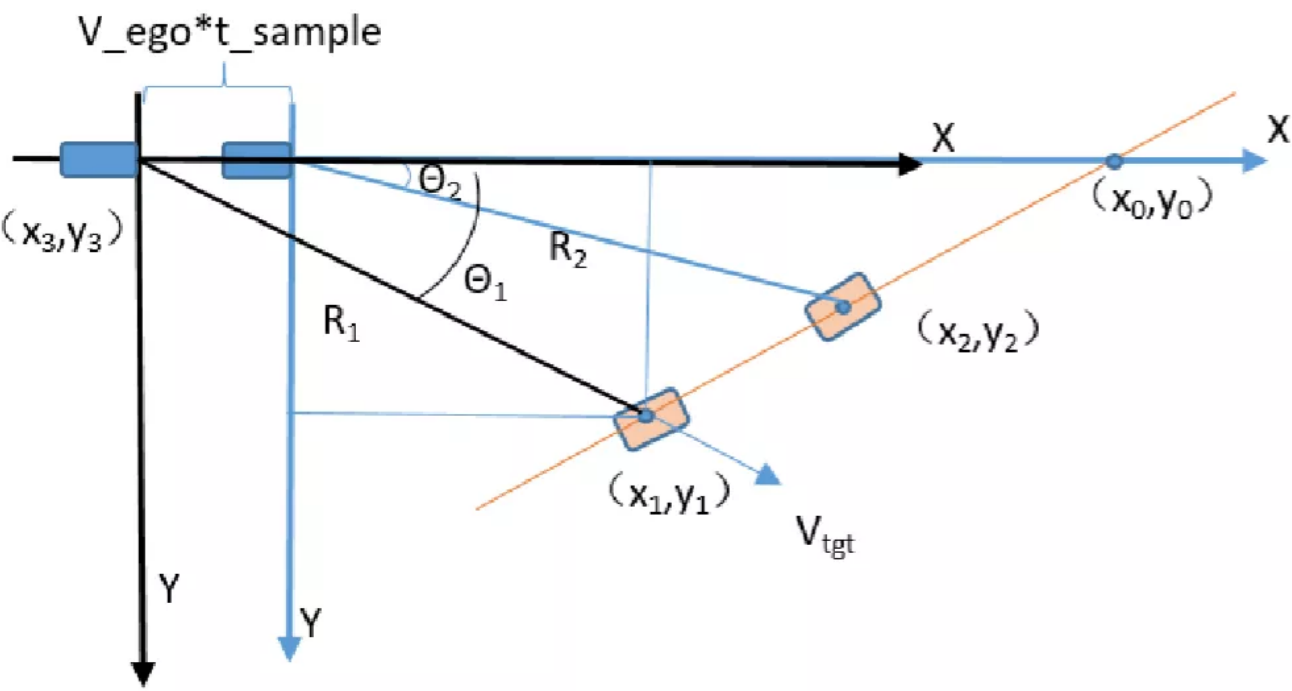


图 4.28 交叉轨迹的数学运动模型

在此运动数学模型中，本车的 TIS 雷达的频率是 25Hz，因此该模型的采样周期（t_sample）是 0.04s，本车与目标车的两个相邻的运动位置时间间隔也是 0.04s。由于本车是直线运动，所以本车的运动是沿着本车坐标系的 X 轴方向运动的。

以(x2,y2)点所处的时间为当前时间

$$x_3=-v_ego \times t_sample \tag{4-28}$$

$$y_3=0 \tag{4-29}$$

$$x_1=R_1 \cos\theta_1+x_3 \tag{4-30}$$

$$y_1=R_1 \sin\theta_1 \tag{4-31}$$

假设目标车的运动轨迹为一阶线性方程，其方程为

当 y=0 时，即目标车进入本车车道，可求得X0

$$x_0=(y_2 \ x_1-y_1 \ x_2)/y_2-y_1 \tag{4-34}$$

和上一个数学模型类似，只有当本车和目标车同时到达交汇点 (x_0,y_0) 时,本车 才需要进行制动。

$$T_{ego}=x_0/v_{ego} \tag{4-35}$$

$$T_{tgt}=(x_0-x_2)/v_{tgt_x} \tag{4-36}$$

建立 Simulink 模型，并运行，最终得到的结果如图 4.29 所示

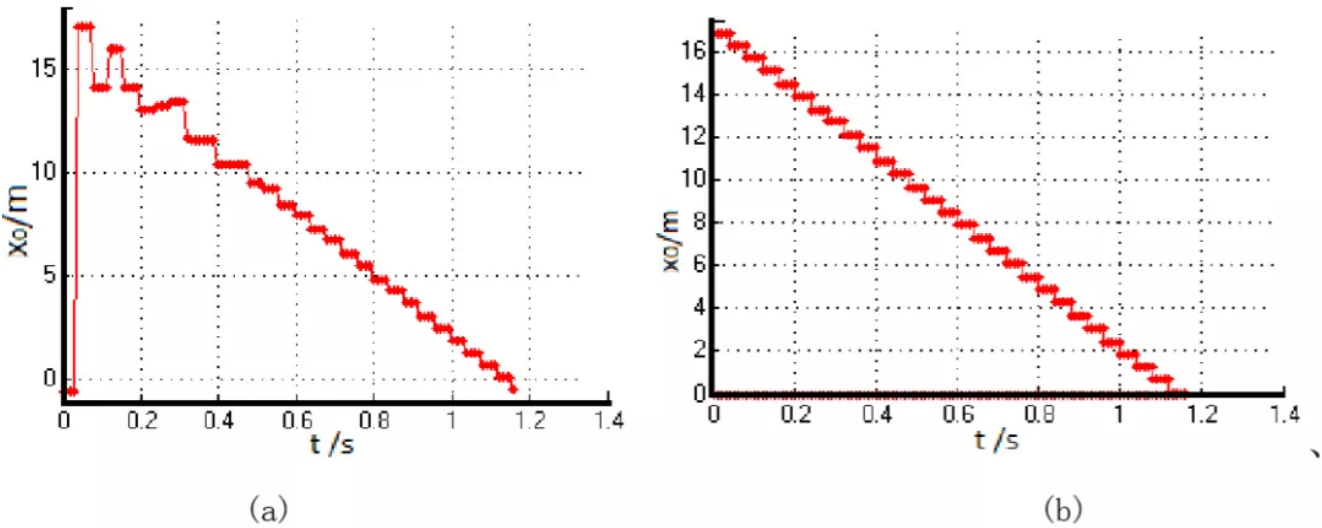


图 4.29 (a)x0计算值 (b)x0仿真实际值

预测碰撞点结果与实际碰撞距离之间的误差很小，证明该数学模型在此种工况 下碰撞点预测正确。

由于当 T_{ego} 与 T_{tgt} 相等时，两车有可能相撞，则设定

$$\Delta T=T_{ego}-T_{tgt} \tag{4-37}$$

当 ΔT 等于零时两车的运动中心将会在同一时间重合，然而实际车辆外形是一个 近似与长方形的形状，在文献[48]提到一般假设研究车辆的运动模型时，会将车辆 简化为为质点、圆形和方形。因此 ΔT 的绝对值在小于摸个设定的阈值范围内的情况 下，车辆有可能发生碰撞。

6. AEB 系统控制算法执行逻辑分析

在 AEB 系统模型中，执行逻辑算法会根据驾驶员输入信号、AEB 系统的安全距 离模型输出，决定最终 AEB 系统输出的信号。

驾驶员操作包括踩制动踏板、踩节气门踏板、方向盘转动。AEB 系统的安全距 离模型输出包括预警旗标（warning flag）、部分力制动旗标（part brake flag）、全力制 动旗标（full

brake flag)。

当仅仅只有驾驶员踩下制动踏板，然而没有任何旗标竖起，则执行逻辑仅仅输出 驾驶员制动力；所有旗帜不竖起，并且驾驶员无制动，则输出制动压力为零；当仅仅 部分力制动旗标竖起，驾驶员无制动或者转向动作，则输出部分制动的制动压力；当 仅仅只有全力制动旗标竖起，而部分制动旗标未竖起时，表明 AEB 系统可能出现错 误信号，因此在这种情况下，如果 驾驶员没有踩下制动踏板，则输出制动压力为零；当部分制动旗标和全力制动旗标竖起的时候，输出全力制动的制动压力；驾驶员踩下 制动踏板，并且部分制动旗标和全力制动旗标竖起，则全力制动的制动压力。

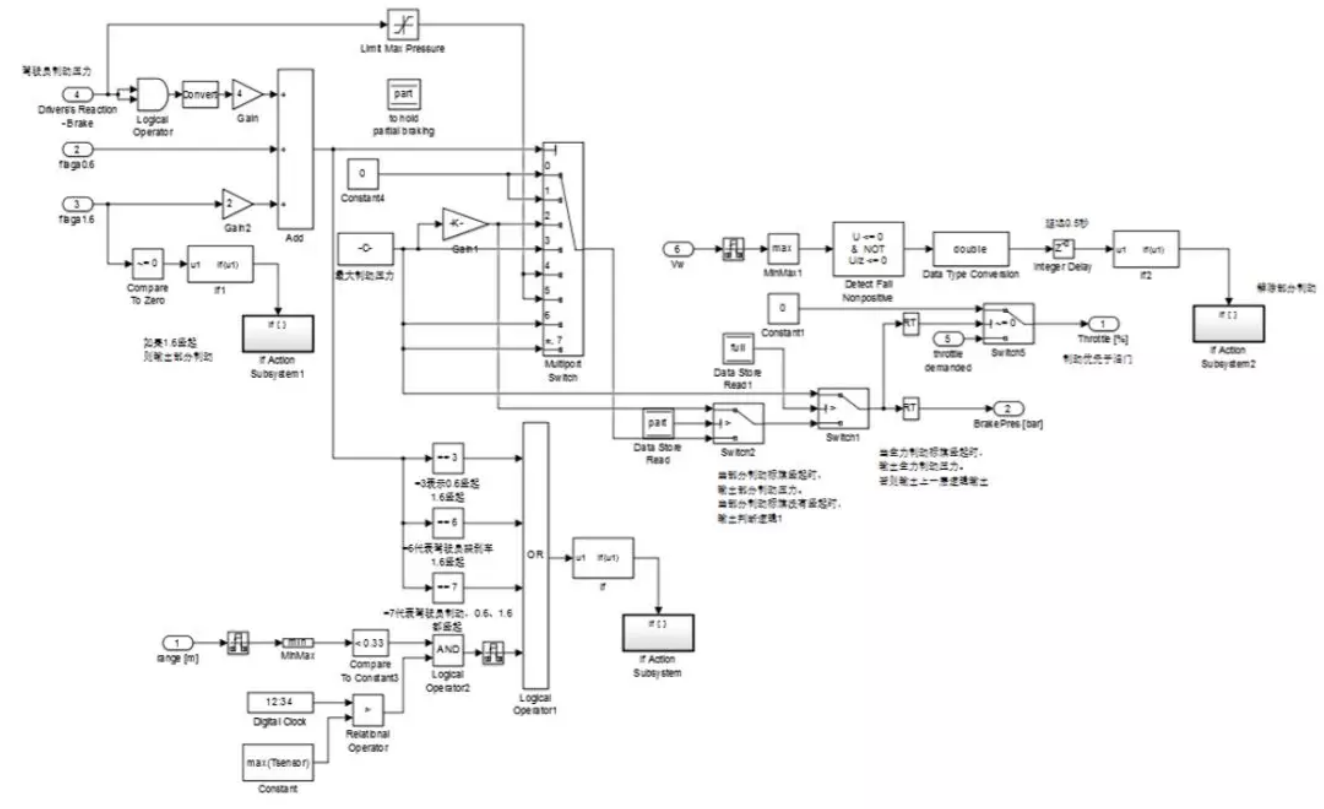


图 4.30 AEB 系统控制算法执行逻辑 Simulink 模型

在运行 AEB 系统模型时，当实际距离小于一定的安全距离时，AEB 系统会自动 制动，本车的速度会降低，则安全距离模型中的安全距离会变大。制动一段时间后， 安全距离会小于两车的相对距离，则 AEB 系统会停止制动，当停止制动后，本车会 继续滑行，而由于没有制动车速变化不大，此时的安全距离模型算出来的安全距离又 会再次大于两车实际相对距离，AEB 系统又会自动制动。由此造成 AEB 系统会往复 地升高与降低制动压力，而导致本车产生严重的抖动。

为了解决 AEB 系统制动压力波动的问题，需要建立稳压模型。部分制动时油压 上升，需要将油压保持一段时间，但是么有必要将车完全制动至停止，因此本文将部分制动的稳压时间设置

为 0.6s；一般车辆情况很危急的时候，AEB 系统才会升高油 压至最大压力，所以在发生 AEB 系统请求全力制动的时候，应保持最大制动压力，直至车辆静止。

在仿真结果中，消除了 AEB 系统制动时本车的制动压力与本车速度的波动。

本章小结

本章对 AEB 系统的信号处理、安全距离、复杂场景的控制算法和执行逻辑进行了研究。信号处理部分是对离散突变信号、测量范围之外的信号等进行了处理；安全 距离模型包括 Mazda 模型、Honda 模型、Berkeley 模型、SeungwukMoon 模型、TTC 模型，最终得出 TTC 模型在不干预驾驶员正常驾驶的方面表现最好；复杂场景包 括多目标车场景、不同车道的目标车场景、驾驶员变道场景、弯道场景、交叉轨迹场景，本文针对这些场景都建立了相应的 AEB 控制算法，并取得了较好的控制效果；逻辑控制算法是综合了驾驶员输入和 AEB 系统的控制逻辑，并最终 AEB 系统控制算 法输出的算法。