doi:10.12068/j.issn.1005－3026.2026.20240211

四轮独立驱动车辆的MPC快速协调控制算法

王宏伟，汪 洵，王力强
（东北大学秦皇岛分校 控制工程学院，河北 秦皇岛 066004）

摘 要：为提高四轮独立驱动车辆的综合性能，提出一种基于模型预测控制的主动前轮转向和直接横摆力矩协调快速控制算法．首先，建立车辆的七自由度非线性模型，设计带有多约束条件的模型预测控制器．然后，针对控制算法中二次规划求解困难的问题，利用拉盖尔函数构造其卡鲁什-库恩-塔克（KKT）条件，再通过斜坡函数将二次规划问题转化为隐式方程的形式，并采用改进的有效活动约束集法进行高效求解．最后，联合MATLAB/Simulink和CarSim搭建仿真平台，在不同的驾驶工况下对提出的快速协调控制算法进行有效性验证．结果表明，快速协调控制算法在保证车辆行驶稳定的前提下，缩减了系统的求解时间，提高了车辆系统的实时性．

关 键 词：四轮独立驱动车辆；模型预测控制；协调控制；二次规划问题；高效求解
中图分类号：TP 273 **文献标志码**：A **文章编号**：1005-3026(2026)02-0001-09

Fast Coordinated Control Algorithm for Four-Wheel Independent Driven Vehicles Based on MPC

WANG Hong-wei, *WANG Xun*, *WANG Li-qiang*
(School of Control Engineering, Northeastern University at Qinhuangdao, Qinhuangdao 066004, China. Corresponding author: WANG Hong-wei, E-mail: wanghw0819@163.com)

Abstract: To enhance the comprehensive performance of four-wheel independent driven vehicles, a fast control algorithm based on model predictive control is proposed to coordinate the active front wheel steering and direct yaw moment control. Firstly, a seven-degree-of-freedom nonlinear model of the vehicle is established, and a model predictive controller with multiple constraints is designed. Then, aiming at the difficulty of solving the quadratic programming problem with multiple constraints in the model predictive control algorithm, the Lagrange function is used to construct its Karush-Kuhn-Tucker (KKT) condition, and the quadratic programming problem is transformed into the form of implicit equation by the ramp function, and the improved effective active constraint set method is used to solve the problem efficiently. Finally, MATLAB/Simulink and CarSim are combined to build a simulation platform, and the effectiveness of the proposed fast model predictive coordinated control algorithm is verified under different driving conditions. The experimental results show that the proposed fast model predictive coordinated control algorithm can greatly reduce the solving time and improve the real-time performance of the vehicle system.

Key words: four-wheel independent driven vehicles; model predictive control; coordinated control; quadratic programming problem; efficient solution

四轮独立驱动车辆由于其精确的动力分配、灵活的轮胎附着分布以及简单的布置方式受到人们越来越多的关注[1]．四轮独立驱动车辆的每个轮胎都配备了电机，使每个轮胎拥有独立可控的特性，这不仅提高了车辆的纵向和横向运动的可操纵性，也大大降低车辆的能源损耗率．这些特点使其成为工业和学术领域关注的焦点，但要满足各个轮胎之间转矩分配以及运动的协调，对

收稿日期：2024-11-08
基金项目：中央高校基本科研业务费专项资金资助项目（N2223029）．
作者简介：王宏伟（1983—），女，辽宁铁岭人，东北大学副教授．
通信作者：王宏伟，E-mail:wanghw0819@163.com.

多个电机的同步协调控制策略的要求较高,同时对系统的控制算法提出更高的要求.

近年来,电子技术在车辆领域的发展尤为迅速,许多车辆主动安全控制系统已经从理论走向了实际应用.主动前轮转向(AFS)系统、直接横摆力矩控制(DYC)系统以及防抱死制动系统(ABS)在车辆稳定性方面得到了广泛的发展,而在车辆横摆稳定性研究上主要是AFS与DYC相结合.AFS通过提供额外转角来实现车辆理想的转向特性,但其控制效果会受轮胎横向力的限制.DYC通过改变轮胎的纵向力,产生车辆需要的直接横摆力矩,但其改变轮胎纵向力的过程会使车速产生波动.单独使用AFS和DYC系统能够实现车辆的稳定性控制,它们各自的缺陷会限制其控制效果[2].

目前AFS和DYC协同控制的方法有很多,包括但不限于模糊控制[3]、滑模控制[4]、鲁棒控制[5]等.这些控制器主要依赖当前车辆状态量来判断车辆稳定性,并且针对AFS和DYC分别设计控制器,然后对整体进行协同控制.然而,这种方法在极端情况下难以确保车辆的稳定,并且子控制器之间存在耦合关系.模型预测控制(MPC)凭借其可对系统未来输出进行预测、处理多约束问题、对系统进行解耦控制等一系列优点,成为车辆系统中稳定性控制的常用方法.文献[6]综合考虑系统的稳定性以及约束指标,利用MPC提出一种基于AFS和DYC的新型控制策略,解决了多目标多约束优化问题.文献[7]基于AFS和DYC的集成控制,建立了混合MPC控制策略,并采用多参数规划进行离线计算,从而提高效率.尽管离线计算可以在一定程度上提高效率,但是模型越复杂,系统的计算负担就越大.目前,越来越多的学者致力于研究如何提高MPC的实时性,相关方法主要集中在二次规划问题的求解方法上,包括改进的内点法[8]、对偶点法[9]、有效集法[10]等.这些方法在理论上对快速求解MPC进行研究,实际车辆系统中的相关研究相对较少.文献[11]设计实时MPC轨迹规划器,并对多约束问题进行序列线性化,实现对约束条件的放松,从而提高计算速率.文献[12]提出一种基于交替方向乘子法的MPC在线优化方法,并将MPC算法分解成多个较小的计算任务,达到加速求解的目标.上述方法处理的都是轨迹跟踪问题,车辆模型简单.而对于车辆稳定性控制,需要更精准的系统模型,这对MPC快速求解算法有更高的要求.

针对上述问题,设计一种快速MPC控制器,协调控制AFS与DYC主动安全系统,提高四轮独立驱动车辆的综合性能.首先,建立7自由度车辆非线性模型.然后,设计基于MPC算法的AFS与DYC协调策略.为提高处理多约束复杂系统时MPC算法的求解速率,提出一种用斜坡函数隐式表达二次规划问题KKT条件的快速求解方法.最后,搭建MATLAB/Simulink与CarSim的联合仿真平台,选取双移线工况和蛇形工况,对提出的快速MPC控制器的控制效果进行仿真验证.

1 系统模型

建立包括车辆纵向、横向、横摆和4个轮胎旋转运动的7自由度模型,如图1所示.根据车辆运动学和牛顿力学的基本原理,构造车辆的动力学方程如下:

$$\left.\begin{aligned}
\dot{\boldsymbol{v}}_x&=\frac{1}{m}\left[(\boldsymbol{F}_{x\mathrm{fl}}+\boldsymbol{F}_{x\mathrm{fr}})\cos\delta-(\boldsymbol{F}_{y\mathrm{fl}}+\boldsymbol{F}_{y\mathrm{fr}})\sin\delta+\boldsymbol{F}_{x\mathrm{dl}}+\boldsymbol{F}_{x\mathrm{dr}}\right]+\boldsymbol{v}_y\boldsymbol{\gamma},\\
\dot{\boldsymbol{v}}_y&=\frac{1}{m}\left[(\boldsymbol{F}_{x\mathrm{fl}}+\boldsymbol{F}_{x\mathrm{fr}})\sin\delta+(\boldsymbol{F}_{y\mathrm{fl}}+\boldsymbol{F}_{y\mathrm{fr}})\cos\delta+\boldsymbol{F}_{y\mathrm{dl}}+\boldsymbol{F}_{y\mathrm{dr}}\right]-\boldsymbol{v}_x\boldsymbol{\gamma},\\
\dot{\boldsymbol{\gamma}}&=\frac{1}{I_z}\left\{a[(\boldsymbol{F}_{x\mathrm{fl}}+\boldsymbol{F}_{x\mathrm{fr}})\sin\delta+(\boldsymbol{F}_{y\mathrm{fl}}+\boldsymbol{F}_{y\mathrm{fr}})\cos\delta]+\frac{d_\mathrm{f}}{2}[(\boldsymbol{F}_{x\mathrm{fr}}-\boldsymbol{F}_{x\mathrm{fl}})\cos\delta+(\boldsymbol{F}_{y\mathrm{fl}}-\boldsymbol{F}_{y\mathrm{fr}})\sin\delta]+\frac{d_\mathrm{r}}{2}(\boldsymbol{F}_{x\mathrm{dr}}-\boldsymbol{F}_{x\mathrm{dl}})-b(\boldsymbol{F}_{y\mathrm{dl}}+\boldsymbol{F}_{y\mathrm{dr}})\right\}.
\end{aligned}\right\}\tag{1}$$

式中:m为车辆总质量;$\boldsymbol{v}_x$和$\boldsymbol{v}_y$分别为车辆的纵向车速和横向车速;$\boldsymbol{\gamma}$为车辆的横摆角速度;I_z为车辆的横摆转动惯量;a和b分别为车辆质心到前轴、后轴的轴距;d_f和d_r分别为车辆前轮、后轮的轮距;δ为车辆前轮的转角;$\boldsymbol{F}_{xwj}$和$\boldsymbol{F}_{ywj}$分别为轮胎受到的纵向力和横向力,下标$w=\mathrm{f},\mathrm{d}$代表车辆的前后轮,$j=\mathrm{l},\mathrm{r}$代表车辆的左右轮,文中其他部分定义相同.

采用魔术轮胎公式拟合实际车辆轮胎模型,其公式为

$$\left.\begin{aligned}
F(x)&=D\sin\{C\arctan[B(\xi+S_\mathrm{h})(1-E)\\
&\quad+E\arctan(B(\xi+S_\mathrm{h}))]\}+S_\mathrm{v},\\
F(y)&=D\sin\{C\arctan[B(\alpha+S_\mathrm{h})(1-E)\\
&\quad+E\arctan(B(\alpha+S_\mathrm{h}))]\}+S_\mathrm{v}.
\end{aligned}\right\}\tag{2}$$

式中:ξ为轮胎的滑移率;D为峰值因子;B为刚度因子;C为曲线形状因子;E为曲线曲率因子;S_h为曲线水平方向的漂移量;S_v为曲线垂直方向的漂移量;α为前轮的侧偏角.

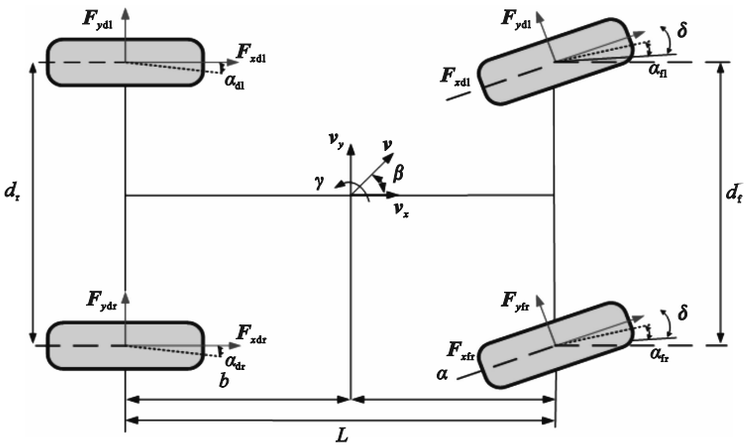


图1　车辆动力学模型

Fig. 1　Vehicle dynamics model

车辆纵向力的滑移率定义如下:

$$\xi_{ij}=\frac{\omega_{ij}R-v_{ij}}{\max(\omega_{ij}R,v_{ij})}. \tag{3}$$

式中:R为车轮有效滚动半径;ω_{ij}和v_{ij}分别为轮胎的滚动角速度和纵向行驶速度.

当电机处于驱动模式时,$\omega_{ij}R>v_{ij}$,此时$\xi_{ij}>0$;当电机处于再生制动模式时,$\omega_{ij}R<v_{ij}$,此时$\xi_{ij}<0$;当电机处于纯滚动状态时,$\omega_{ij}R=v_{ij}$,此时$\xi_{ij}=0$.各个轮胎中心处纵向速度v_{ij}的公式如下:

$$\left.\begin{aligned}
v_{fl}&=\left(v_x-\frac{\gamma d_f}{2}\right)\cos\delta+(v_y+\gamma a)\sin\delta,\\
v_{dl}&=v_x-\frac{\gamma d_f}{2},\\
v_{fr}&=\left(v_x+\frac{\gamma d_f}{2}\right)\cos\delta+(v_y+\gamma a)\sin\delta,\\
v_{dr}&=v_x+\frac{\gamma d_f}{2}.
\end{aligned}\right\} \tag{4}$$

定义4个轮胎的侧滑角如下:

$$\left.\begin{aligned}
\alpha_{fl}&=\arctan\left(\frac{v_y+\gamma a}{v_x-\gamma\frac{d_f}{2}}\right)-\delta,\\
\alpha_{dl}&=\arctan\left(\frac{v_y-\gamma b}{v_x-\gamma\frac{d_r}{2}}\right),\\
\alpha_{fr}&=\arctan\left(\frac{v_y+\gamma a}{v_x+\gamma\frac{d_f}{2}}\right)-\delta,\\
\alpha_{dr}&=\arctan\left(\frac{v_y-\gamma b}{v_x+\gamma\frac{d_r}{2}}\right).
\end{aligned}\right\} \tag{5}$$

4个轮胎垂直载荷的计算公式如下:

$$\left.\begin{aligned}
F_{zfl}&=\frac{mgb}{2L}-\frac{ma_xh_g}{2L}-\frac{ma_yh_g}{d_f}\frac{b}{L},\\
F_{zfr}&=\frac{mgb}{2L}-\frac{ma_xh_g}{2L}+\frac{ma_yh_g}{d_f}\frac{b}{L},\\
F_{zdl}&=\frac{mga}{2L}+\frac{ma_xh_g}{2L}-\frac{ma_yh_g}{d_r}\frac{a}{L},\\
F_{zdr}&=\frac{mga}{2L}+\frac{ma_xh_g}{2L}+\frac{ma_yh_g}{d_r}\frac{a}{L}.
\end{aligned}\right\} \tag{6}$$

式中:$L=a+b$为车辆前轴到后轴的长度;a_x和a_y分别为车辆纵向、横向加速度;h_g为车辆的质心高度;g为重力加速度.

利用魔术轮胎公式计算得到的纵向力和横向力需要在极端条件下满足附着椭圆,修正后的纵向力和横向力如下:

$$\left.\begin{aligned}
F_{xij}&=\frac{|\sigma_{xij}|}{\sigma_{ij}}F(x),\\
F_{yij}&=\frac{|\sigma_{yij}|}{\sigma_{ij}}F(y),\\
\sigma_{ij}&=\sqrt{\sigma_{xij}^2+\sigma_{yij}^2},\\
\sigma_{xij}&=\frac{\xi_{ij}}{1+\xi_{ij}},\\
\sigma_{yij}&=\frac{\tan\alpha_{ij}}{1+\xi_{ij}}.
\end{aligned}\right\} \tag{7}$$

对轮胎的旋转运动进行动力学分析,其方程为

$$I_\omega\dot{\omega}_{ij}=T_{ij}-F_{xij}R. \tag{8}$$

式中:I_ω为轮胎滚动惯量;T_{ij}为电机输出给轮胎的转矩.

2　协调控制的快速MPC控制器设计

2.1　预测模型设计

本文设计的快速MPC控制器,以车辆的前轮

转角以及4个轮胎的纵向力作为系统控制量,表示为 $\boldsymbol{u}=[\delta\ \boldsymbol{F}_{x\mathrm{fl}}\ \boldsymbol{F}_{x\mathrm{fr}}\ \boldsymbol{F}_{x\mathrm{dl}}\ \boldsymbol{F}_{x\mathrm{dr}}]^{\mathrm{T}}$,其中前轮转角代表AFS的控制效果,4个车轮的纵向力代表DYC的控制效果,利用MPC对它们进行协调控制.系统状态变量选择为车辆的纵向速度、横向速度和横摆角速度:$\boldsymbol{x}=[\boldsymbol{v}_x\ \boldsymbol{v}_y\ \boldsymbol{\gamma}]^{\mathrm{T}}$;系统的输出选择为车辆的纵向车速、质心侧偏角$\beta$和横摆角速度:$\boldsymbol{y}=[\boldsymbol{v}_x\ \beta\ \boldsymbol{\gamma}]^{\mathrm{T}}$.

车辆非线性系统动力学模型如下:

$$\left.\begin{aligned}\dot{\boldsymbol{x}}=f(\boldsymbol{x},\boldsymbol{u})=\begin{bmatrix}f_1(\boldsymbol{x},\boldsymbol{u})\\ f_2(\boldsymbol{x},\boldsymbol{u})\\ f_3(\boldsymbol{x},\boldsymbol{u})\end{bmatrix},\\ \boldsymbol{y}=\boldsymbol{E}\boldsymbol{x}.\end{aligned}\right\}\tag{9}$$

式中:

$$f_1(\boldsymbol{x},\boldsymbol{u})=\frac{1}{m}[(\boldsymbol{F}_{x\mathrm{fl}}+\boldsymbol{F}_{x\mathrm{fr}})\cos\delta-(\boldsymbol{F}_{y\mathrm{fl}}+\boldsymbol{F}_{y\mathrm{fr}})\sin\delta+\boldsymbol{F}_{x\mathrm{dl}}+\boldsymbol{F}_{x\mathrm{dr}}]+\boldsymbol{v}_y\boldsymbol{\gamma};$$

$$f_2(\boldsymbol{x},\boldsymbol{u})=-\boldsymbol{v}_x\boldsymbol{\gamma}+\frac{1}{m}[(\boldsymbol{F}_{y\mathrm{fl}}+\boldsymbol{F}_{y\mathrm{fr}})\cos\delta+(\boldsymbol{F}_{x\mathrm{fl}}+\boldsymbol{F}_{x\mathrm{fr}})\sin\delta+\boldsymbol{F}_{y\mathrm{dl}}+\boldsymbol{F}_{y\mathrm{dr}}];$$

$$f_3(\boldsymbol{x},\boldsymbol{u})=\frac{a}{I_z}[(\boldsymbol{F}_{x\mathrm{fl}}+\boldsymbol{F}_{x\mathrm{fr}})\sin\delta+(\boldsymbol{F}_{y\mathrm{fl}}+\boldsymbol{F}_{y\mathrm{fr}})\cos\delta]+\frac{d_{\mathrm{f}}}{2I_z}[(\boldsymbol{F}_{x\mathrm{fr}}-\boldsymbol{F}_{x\mathrm{fl}})\cos\delta+(\boldsymbol{F}_{y\mathrm{fl}}-\boldsymbol{F}_{y\mathrm{fr}})\sin\delta]-\frac{b}{I_z}(\boldsymbol{F}_{y\mathrm{dl}}+\boldsymbol{F}_{y\mathrm{dr}})+$$

$$\frac{d_{\mathrm{r}}}{2I_z}(\boldsymbol{F}_{x\mathrm{dr}}-\boldsymbol{F}_{x\mathrm{dl}});\boldsymbol{E}=\begin{bmatrix}1&0&0\\0&1/\boldsymbol{v}_x&0\\0&0&1\end{bmatrix}.$$

取系统的参考点$(\boldsymbol{x}_0,\boldsymbol{u}_0)$,非线性系统模型写成如下形式:

$$\dot{\boldsymbol{x}}_0=f(\boldsymbol{x}_0,\boldsymbol{u}_0).\tag{10}$$

在参考点$(\boldsymbol{x}_0,\boldsymbol{u}_0)$处进行泰勒级数展开,忽略高阶项只保留一阶项,得到如下方程:

$$\dot{\boldsymbol{x}}=f(\boldsymbol{x}_0,\boldsymbol{u}_0)+\boldsymbol{A}_t(\boldsymbol{x}-\boldsymbol{x}_0)+\boldsymbol{B}_t(\boldsymbol{u}-\boldsymbol{u}_0).\tag{11}$$

式中:$\boldsymbol{A}_t=\dfrac{\partial f(\boldsymbol{x},\boldsymbol{u})}{\partial\boldsymbol{x}}=\begin{bmatrix}\dfrac{\partial f}{\partial\boldsymbol{v}_x}&\dfrac{\partial f}{\partial\boldsymbol{v}_y}&\dfrac{\partial f}{\partial\boldsymbol{\gamma}}\end{bmatrix};\boldsymbol{B}_t=\dfrac{\partial f(\boldsymbol{x},\boldsymbol{u})}{\partial\boldsymbol{u}}=$

$$\begin{bmatrix}\dfrac{\partial f}{\partial\delta_f}&\dfrac{\partial f}{\partial\boldsymbol{F}_{x\mathrm{fr}}}&\dfrac{\partial f}{\partial\boldsymbol{F}_{x\mathrm{fl}}}&\dfrac{\partial f}{\partial\boldsymbol{F}_{x\mathrm{rr}}}&\dfrac{\partial f}{\partial\boldsymbol{F}_{x\mathrm{rl}}}\end{bmatrix}.$$

结合式(10)和式(11),线性化模型如下:

$$\dot{\boldsymbol{x}}-\dot{\boldsymbol{x}}_0=\boldsymbol{A}_t(\boldsymbol{x}-\boldsymbol{x}_0)+\boldsymbol{B}_t(\boldsymbol{u}-\boldsymbol{u}_0).\tag{12}$$

利用前向欧拉法将式(12)离散化,得到系统的状态空间方程:

$$\left.\begin{aligned}\boldsymbol{x}(k+1)=\boldsymbol{A}_{k,t}\boldsymbol{x}(k)+\boldsymbol{B}_{k,t}\boldsymbol{u}(k)+\boldsymbol{d}(k),\\ \boldsymbol{y}(k)=\boldsymbol{E}x(k).\end{aligned}\right\}\tag{13}$$

式中:$d(k)=\boldsymbol{x}_0(k+1)-\boldsymbol{A}_{k,t}\boldsymbol{x}_0(k)-\boldsymbol{B}_{k,t}\boldsymbol{u}_0(k)$;$\boldsymbol{x}(k)$为$k$时刻车辆的状态变量;$\boldsymbol{u}(k)$为$k$时刻系统的控制输入;$\boldsymbol{A}_{k,t}=T\boldsymbol{A}_t+\boldsymbol{I}$;$\boldsymbol{B}_{k,t}=T\boldsymbol{B}_t$;$\boldsymbol{I}$为单位矩阵,$T$为采样周期.

为减小系统误差,定义新的状态变量为

$$\boldsymbol{x}_m(k)=\begin{bmatrix}\boldsymbol{x}(k)\\ \boldsymbol{u}(k-1)\end{bmatrix}^{\mathrm{T}}.\tag{14}$$

式(13)中的控制量在新状态变量下被改写为控制增量的形式如下:

$$\left.\begin{aligned}\boldsymbol{x}_{\mathrm{m}}(k+1)=\boldsymbol{A}_{\mathrm{a}}\boldsymbol{x}_{\mathrm{m}}(k)+\boldsymbol{B}_{\mathrm{a}}\Delta\boldsymbol{u}(k)+\boldsymbol{d}_{\mathrm{a}}(k);\\ \boldsymbol{y}(k)=\boldsymbol{E}_{\mathrm{a}}\boldsymbol{x}_{\mathrm{m}}(k).\end{aligned}\right\}\tag{15}$$

式中:$\boldsymbol{A}_{\mathrm{a}}=\begin{bmatrix}\boldsymbol{A}_z&\boldsymbol{B}_z\\ \boldsymbol{0}&\boldsymbol{I}\end{bmatrix};\boldsymbol{B}_{\mathrm{a}}=\begin{bmatrix}\boldsymbol{B}_z\\ \boldsymbol{I}\end{bmatrix};\boldsymbol{d}_{\mathrm{a}}(k)=\begin{bmatrix}\boldsymbol{d}(k)\\ \boldsymbol{0}\end{bmatrix};\boldsymbol{E}_{\mathrm{a}}=$

$[\boldsymbol{E}_z\quad\boldsymbol{0}]$;$\boldsymbol{A}_z$,$\boldsymbol{B}_z$和$\boldsymbol{E}_z$分别为原始系统的状态转移矩阵、输入矩阵和输出矩阵.

在预测时域N_{p}、控制时域N_{c}的条件下对预测输出进行迭代,输出方程为

$$\boldsymbol{Y}(k)=\boldsymbol{S}_a\boldsymbol{x}_{\mathrm{m}}(k)+\boldsymbol{S}_u\Delta\boldsymbol{U}(k)+\boldsymbol{S}_d\boldsymbol{D}(k).\tag{16}$$

式中:$\boldsymbol{Y}(k)=\begin{bmatrix}\boldsymbol{y}(k+1)\\ \boldsymbol{y}(k+2)\\ \vdots\\ \boldsymbol{y}(k+N_{\mathrm{c}})\\ \vdots\\ \boldsymbol{y}(k+N_{\mathrm{p}})\end{bmatrix};\boldsymbol{S}_{\mathrm{a}}=\begin{bmatrix}\boldsymbol{C}_{\mathrm{a}}\boldsymbol{A}_{\mathrm{a}}\\ \boldsymbol{C}_{\mathrm{a}}\boldsymbol{A}_{\mathrm{a}}^2\\ \vdots\\ \boldsymbol{C}_{\mathrm{a}}\boldsymbol{A}_{\mathrm{a}}^{N_{\mathrm{c}}}\\ \vdots\\ \boldsymbol{C}_{\mathrm{a}}\boldsymbol{A}_{\mathrm{a}}^{N_{\mathrm{p}}}\end{bmatrix};$

$$\boldsymbol{S}_u=\begin{bmatrix}\boldsymbol{C}_{\mathrm{a}}\boldsymbol{B}_{\mathrm{a}}&\boldsymbol{0}&\cdots&\boldsymbol{0}\\ \boldsymbol{C}_{\mathrm{a}}\boldsymbol{A}_{\mathrm{a}}\boldsymbol{B}_{\mathrm{a}}&\boldsymbol{C}_{\mathrm{a}}\boldsymbol{B}_{\mathrm{a}}&\cdots&\boldsymbol{0}\\ \vdots&\vdots&&\vdots\\ \boldsymbol{C}_{\mathrm{a}}\boldsymbol{A}_{\mathrm{a}}^{N_{\mathrm{c}}-1}\boldsymbol{B}_{\mathrm{a}}&\boldsymbol{C}_{\mathrm{a}}\boldsymbol{A}_{\mathrm{a}}^{N_{\mathrm{c}}-2}\boldsymbol{B}_{\mathrm{a}}&\cdots&\boldsymbol{C}_{\mathrm{a}}\boldsymbol{B}_{\mathrm{a}}\\ \boldsymbol{C}_{\mathrm{a}}\boldsymbol{A}_{\mathrm{a}}^{N_{\mathrm{c}}}\boldsymbol{B}_{\mathrm{a}}&\boldsymbol{C}_{\mathrm{a}}\boldsymbol{A}_{\mathrm{a}}^{N_{\mathrm{c}}-1}\boldsymbol{B}_{\mathrm{a}}&\cdots&\boldsymbol{C}_{\mathrm{a}}\boldsymbol{A}_{\mathrm{a}}\boldsymbol{B}_{\mathrm{a}}\\ \vdots&\vdots&&\vdots\\ \boldsymbol{C}_{\mathrm{a}}\boldsymbol{A}_{\mathrm{a}}^{N_{\mathrm{p}}-1}\boldsymbol{B}_{\mathrm{a}}&\boldsymbol{C}_{\mathrm{a}}\boldsymbol{A}_{\mathrm{a}}^{N_{\mathrm{p}}-2}\boldsymbol{B}_{\mathrm{a}}&\cdots&\boldsymbol{C}_{\mathrm{a}}\boldsymbol{A}_{\mathrm{a}}^{N_{\mathrm{p}}-N_{\mathrm{c}}}\boldsymbol{B}_{\mathrm{a}}\end{bmatrix};$$

$$\Delta\boldsymbol{U}(k)=\begin{bmatrix}\Delta\boldsymbol{u}(k)\\ \Delta\boldsymbol{u}(k+1)\\ \vdots\\ \Delta\boldsymbol{u}(k+N_{\mathrm{c}}-1)\end{bmatrix};\boldsymbol{D}(k)=\begin{bmatrix}\boldsymbol{d}_{\mathrm{a}}(k)\\ \boldsymbol{d}_{\mathrm{a}}(k+1)\\ \vdots\\ \boldsymbol{d}_{\mathrm{a}}(k+N_{\mathrm{p}}-1)\end{bmatrix};$$

$$\boldsymbol{S}_d=\begin{bmatrix}\boldsymbol{C}_{\mathrm{a}}&\boldsymbol{0}&\cdots&\boldsymbol{0}\\ \boldsymbol{C}_{\mathrm{a}}\boldsymbol{A}_{\mathrm{a}}&\boldsymbol{C}_{\mathrm{a}}&\cdots&\boldsymbol{0}\\ \vdots&\vdots&&\vdots\\ \boldsymbol{C}_{\mathrm{a}}\boldsymbol{A}_{\mathrm{a}}^{N_{\mathrm{c}}-1}&\boldsymbol{C}_{\mathrm{a}}\boldsymbol{A}_{\mathrm{a}}^{N_{\mathrm{c}}-2}&\cdots&\boldsymbol{C}_{\mathrm{a}}\\ \boldsymbol{C}_{\mathrm{a}}\boldsymbol{A}_{\mathrm{a}}^{N_{\mathrm{c}}}&\boldsymbol{C}_{\mathrm{a}}\boldsymbol{A}_{\mathrm{a}}^{N_{\mathrm{c}}-1}&\cdots&\boldsymbol{C}_{\mathrm{a}}\boldsymbol{A}_{\mathrm{a}}\\ \vdots&\vdots&&\vdots\\ \boldsymbol{C}_{\mathrm{a}}\boldsymbol{A}_{\mathrm{a}}^{N_{\mathrm{p}}-1}&\boldsymbol{C}_{\mathrm{a}}\boldsymbol{A}_{\mathrm{a}}^{N_{\mathrm{p}}-2}&\cdots&\boldsymbol{C}_{\mathrm{a}}\boldsymbol{A}_{\mathrm{a}}^{0}\end{bmatrix};$$

$\boldsymbol{C}_{\mathrm{a}}$为输出系数矩阵.

2.2 代价函数及约束条件设计

为保证车辆在各种工况下都能快速且精准地跟随目标轨迹运动,并且维持车辆的行驶稳

定,定义代价函数如下:

$$J_{\min}(\boldsymbol{x}(k),\boldsymbol{u}(k-1),\Delta\boldsymbol{U}(k))=\sum_{i=1}^{N_p}\left\|\boldsymbol{e}(k+i)\right\|_{\boldsymbol{P}}^2+\sum_{i=0}^{N_c-1}\left\|\Delta\boldsymbol{u}(k+i)\right\|_{\boldsymbol{Q}}^2+\sum_{i=0}^{N_c-1}\left\|\boldsymbol{u}(k+i)\right\|_{\boldsymbol{S}}^2+\rho\varepsilon^2. \quad (17)$$

式中:$\boldsymbol{e}(k+i)=y(k+i)-y_{\mathrm{ref}}(k+i)$描述了系统实际输出与期望输出的偏差,其大小反映了控制器跟踪期望轨迹的能力,$\boldsymbol{y}_{\mathrm{ref}}$为系统的参考输出,$\boldsymbol{P}$为其权重矩阵;$\Delta\boldsymbol{u}(k+i)$描述了控制量变化大小的限制,$\boldsymbol{Q}$为其权重矩阵;$\boldsymbol{u}(k+i)$描述了控制量大小的限制,$\boldsymbol{S}$为其权重矩阵;$\varepsilon$为松弛因子;$\rho$为其权重因子.

定义$\boldsymbol{z}=[\Delta\boldsymbol{U}(k)\quad\varepsilon]^{\mathrm{T}}$,可将式(17)简化如下:

$$J_{\min}=\frac{1}{2}\boldsymbol{z}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{z}+\boldsymbol{g}^{\mathrm{T}}\boldsymbol{z}+\boldsymbol{W}_g. \quad (18)$$

式中:$\boldsymbol{W}_g=\boldsymbol{E}(k)^{\mathrm{T}}\boldsymbol{P}\boldsymbol{E}(k)+\boldsymbol{U}(k-1)^{\mathrm{T}}\boldsymbol{S}\boldsymbol{U}(k-1)$;$\boldsymbol{g}^{\mathrm{T}}=\begin{bmatrix}2(\boldsymbol{E}(k)^{\mathrm{T}}\boldsymbol{P}\boldsymbol{S}_u+\boldsymbol{U}(k-1)^{\mathrm{T}}\boldsymbol{S}\boldsymbol{M}) & 0\end{bmatrix}$;

$\boldsymbol{H}=\begin{bmatrix}2(\boldsymbol{S}_u^{\mathrm{T}}\boldsymbol{P}\boldsymbol{S}_u+\boldsymbol{Q}+\boldsymbol{M}^{\mathrm{T}}\boldsymbol{S}\boldsymbol{M}) & 0\\ 0 & 2\rho\end{bmatrix}$;　$\boldsymbol{M}=\begin{bmatrix}1&0&\cdots&0\\1&1&\cdots&\vdots\\\vdots&\vdots&&0\\1&1&\cdots&1\end{bmatrix}_{N_c\times N_c}\cdot\boldsymbol{I}_5\boldsymbol{U}$为控制量$\boldsymbol{u}$的序列;$\boldsymbol{S}_u$为矩阵;$\boldsymbol{I}_5$为单位矩阵;$\boldsymbol{W}_g$为与控制输入无关的常数项.

为保证车辆的稳定性和舒适性,对车辆控制量和控制增量以及系统的输出量进行限制如下:

$$\left.\begin{aligned}&\boldsymbol{u}_{\min}\leqslant\boldsymbol{u}(k+i)\leqslant\boldsymbol{u}_{\max};\\&\Delta\boldsymbol{u}_{\min}\leqslant\Delta\boldsymbol{u}(k+i)\leqslant\Delta\boldsymbol{u}_{\max};\\&\boldsymbol{y}_{\min}(k+i)\leqslant\boldsymbol{y}(k+i)\leqslant\boldsymbol{y}_{\max}(k+i).\end{aligned}\right\} \quad (19)$$

在滚动迭代优化过程中,忽略与控制输入无关的常数项$\boldsymbol{W}_g$,得到目标函数的标准二次规划形式:

$$\left.\begin{aligned}&\min_z\frac{1}{2}\boldsymbol{z}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{z}+\boldsymbol{g}^{\mathrm{T}}\boldsymbol{z};\\&\text{s.t. }\boldsymbol{F}\boldsymbol{z}\leqslant\boldsymbol{H}.\end{aligned}\right\} \quad (20)$$

式中:$\boldsymbol{F}=\begin{bmatrix}\boldsymbol{M}&0\\-\boldsymbol{M}&0\\\boldsymbol{I}&0\\-\boldsymbol{I}&0\\\boldsymbol{S}_u&-1\\-\boldsymbol{S}_u&1\end{bmatrix}$;$\boldsymbol{H}=\begin{bmatrix}\boldsymbol{U}_{\max}-\boldsymbol{U}(k-1)\\-\boldsymbol{U}_{\min}+\boldsymbol{U}(k-1)\\\Delta\boldsymbol{U}_{\max}\\-\Delta\boldsymbol{U}_{\max}\\\boldsymbol{Y}_{\max}-\boldsymbol{S}_a\boldsymbol{x}_a(k)-\boldsymbol{S}_d\boldsymbol{D}(k)\\-\boldsymbol{Y}_{\min}+\boldsymbol{S}_a\boldsymbol{x}_a(k)+\boldsymbol{S}_d\boldsymbol{D}(k)\end{bmatrix}$.

当前控制时刻的控制量与上一个控制时刻的控制量之间的关系如下:

$$\boldsymbol{u}(k)=\boldsymbol{u}(k-1)+\Delta\boldsymbol{u}(k). \quad (21)$$

在整个预测过程中,系统不断重复上述滚动迭代优化过程来实现对车辆的稳定性控制.

2.3　MPC快速求解算法

MPC将控制问题转化为二次规划问题,以求解最优控制输入,其求解速率将直接影响系统的控制效果.引入拉盖尔乘子λ,将约束条件转化到拉盖尔函数中,并构建KKT条件,将二次规划问题转化为用斜坡函数表示的隐式方程,采用改进的有效活动约束集法进行快速求解,达到提高控制器实时性的目的.

定义1　斜坡函数定义如下:

$$\left.\begin{aligned}&r(l)=0,\ l<0;\\&r(l)=l,\ l\geqslant0.\end{aligned}\right\} \quad (22)$$

引理1[10]　对于任意实数变量l,斜坡函数是其唯一同时满足下面3个条件的函数.

$$\left.\begin{aligned}&\left[r(l)-l\right]r(l)=0,\\&r(l)\geqslant0,\\&\left[r(l)-l\right]\geqslant0.\end{aligned}\right\} \quad (23)$$

定理1　MPC算法中二次规划式(20)的解是关于变量l的分段仿射表示:

$$\boldsymbol{z}=-\boldsymbol{H}^{-1}\boldsymbol{g}-\boldsymbol{H}^{-1}\boldsymbol{F}^{\mathrm{T}}r(l). \quad (24)$$

证明　定义式(20)的拉盖尔函数如下:

$$L(z,\lambda)=\frac{1}{2}\boldsymbol{z}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{z}+\boldsymbol{g}^{\mathrm{T}}\boldsymbol{z}+\lambda(\boldsymbol{F}\boldsymbol{z}-\boldsymbol{H}). \quad (25)$$

同时,构建该二次规划问题的KKT条件如下:

$$\left.\begin{aligned}&\boldsymbol{H}\boldsymbol{z}+\boldsymbol{g}+\boldsymbol{F}^{\mathrm{T}}\boldsymbol{\lambda}=0,\\&\boldsymbol{\lambda}_i(\boldsymbol{H}_i-\boldsymbol{F}_{(i,\cdot)}\boldsymbol{z})=0,\\&\boldsymbol{\lambda}\geqslant0,\\&\boldsymbol{H}-\boldsymbol{F}\boldsymbol{z}\geqslant0.\end{aligned}\right\} \quad (26)$$

式中,$\boldsymbol{F}_{(i,\cdot)}$为第$i$个约束的系数.

根据式(26)的第1个方程得到$\boldsymbol{z}$在不考虑约束条件下的表达式:

$$\boldsymbol{z}=-\boldsymbol{H}^{-1}\boldsymbol{g}-\boldsymbol{H}^{-1}\boldsymbol{F}^{\mathrm{T}}\lambda. \quad (27)$$

将式(27)代入式(26)中,得到如下方程:

$$\left.\begin{aligned}&\boldsymbol{\lambda}_i\left\{\boldsymbol{\lambda}_i-\left[-\boldsymbol{H}_i-\boldsymbol{F}_{(i,\cdot)}\boldsymbol{H}^{-1}\boldsymbol{g}+(\boldsymbol{I}-\boldsymbol{F}\boldsymbol{H}^{-1}\boldsymbol{F}^{\mathrm{T}})_{(i,\cdot)}\boldsymbol{\lambda}\right]\right\}=0,\\&\lambda\geqslant0,\\&\lambda-\left[-\boldsymbol{H}-\boldsymbol{F}\boldsymbol{H}^{-1}\boldsymbol{g}+(\boldsymbol{I}-\boldsymbol{F}\boldsymbol{H}^{-1}\boldsymbol{F}^{\mathrm{T}})\boldsymbol{\lambda}\right]\geqslant0.\end{aligned}\right\} \quad (28)$$

定义:

$$l=-\boldsymbol{H}-\boldsymbol{F}\boldsymbol{H}^{-1}\boldsymbol{g}+(\boldsymbol{I}-\boldsymbol{F}\boldsymbol{H}^{-1}\boldsymbol{F}^{\mathrm{T}})\lambda. \quad (29)$$

将式(29)代入式(28),建立变量l与乘子集λ的联系,并得到式(28)的简化形式如下:

$$\left.\begin{aligned}&\boldsymbol{\lambda}_i(\boldsymbol{\lambda}_i-\boldsymbol{l}_i)=0,\\&\lambda\geqslant0,\\&\lambda-l\geqslant0.\end{aligned}\right\} \quad (30)$$

式(30)满足引理1,将λ用斜坡函数表示如下:

$$\lambda = r(l). \tag{31}$$

将式(31)代入式(27)可以得到式(27)的解,如定理1所示.

结合式(29)与式(31)可得关于变量l的隐式方程:

$$l = -\boldsymbol{H} - \boldsymbol{F}\boldsymbol{H}^{-1}\boldsymbol{g} + (\boldsymbol{I} - \boldsymbol{F}\boldsymbol{H}^{-1}\boldsymbol{F}^{\mathrm{T}})r(l). \tag{32}$$

斜坡函数是典型的分段函数,乘子集r可以表示为$r(l)=\boldsymbol{I}_c l$,式中$\boldsymbol{I}_c$是阶数与变量l的维数相同的矩阵,并且它的元素满足$\boldsymbol{I}_{c(i,i)}\in\{0,1\}$,$c$为活动约束的集合,隐式方程(32)可以表示为

$$l - (\boldsymbol{I} - \boldsymbol{F}\boldsymbol{H}^{-1}\boldsymbol{F}^{\mathrm{T}})\boldsymbol{I}_c l = -\boldsymbol{H} - \boldsymbol{F}\boldsymbol{H}^{-1}\boldsymbol{g}. \tag{33}$$

定义2 $(l,\boldsymbol{I}_c)$如果满足式(33),并且当$\boldsymbol{l}_i \geqslant 0$时,有$\boldsymbol{I}_{c(i,i)}=1$成立;以及当$\boldsymbol{l}_i<0$时,有$\boldsymbol{I}_{c(i,i)}=0$成立,则称是$(l,\boldsymbol{I}_c)$相容的.

引理2 (矩阵反演引理)设矩阵$\boldsymbol{G}\in\mathbf{R}^{N\times N}$可逆,有$(\boldsymbol{G}+\boldsymbol{g}_u\boldsymbol{g}_h\boldsymbol{g}_v)^{-1}=\boldsymbol{G}^{-1}-\boldsymbol{G}^{-1}\boldsymbol{g}_u(\boldsymbol{g}_h^{-1}+\boldsymbol{g}_v\boldsymbol{G}^{-1}\boldsymbol{g}_u)^{-1}\boldsymbol{g}_v\boldsymbol{G}^{-1}$,其中$\boldsymbol{g}_h\in\mathbf{R}\backslash\{0\}$是个向量,$\boldsymbol{g}_h$属于$\mathbf{R}$且非零,$\boldsymbol{g}_u\in\mathbf{R}^N$是列向量,$\boldsymbol{g}_v^{\mathrm{T}}\in\mathbf{R}^N$是行向量.

定理2 利用改进的有效活动集约束法对式(33)进行求解,最后得到控制量$\boldsymbol{z}$如下:

$$\boldsymbol{z} = -\boldsymbol{H}^{-1}\boldsymbol{g} - \boldsymbol{H}^{-1}\boldsymbol{F}^{\mathrm{T}}\boldsymbol{I}_c l. \tag{34}$$

证明 定义$\boldsymbol{G}(c)=(\boldsymbol{I}-\boldsymbol{I}_c)+\boldsymbol{F}\boldsymbol{H}^{-1}\boldsymbol{F}^{\mathrm{T}}\boldsymbol{I}_c$,式(33)简化表示如下:

$$l = \boldsymbol{G}(c)^{-1}(-\boldsymbol{H} - \boldsymbol{F}\boldsymbol{H}^{-1}\boldsymbol{g}). \tag{35}$$

对于活动约束集c有两种情况,考虑$i\notin c$的情况,定义$\boldsymbol{c}_{+i}=c\cup\{i\}$,意味着将$i$加入到原有的活动约束集$c$中,而$i\in c$的情况下定义$\boldsymbol{c}_{-i}=c\backslash\{i\}$,则意味着将$i$从活动约束集$c$中去除.

定义3 $\boldsymbol{v}(c,i)=\boldsymbol{G}(c)^{-1}\left[(\boldsymbol{I}-\boldsymbol{I}_c)-\boldsymbol{F}\boldsymbol{H}^{-1}\boldsymbol{F}^{\mathrm{T}}\right]_{(\cdot,i)}$,当$i\notin c$时,结合引理2,将$i$加入到原有的活动约束集$c$中,并更新可逆矩阵$\boldsymbol{G}(\boldsymbol{c}_{+i})$,可得

$$\boldsymbol{G}(\boldsymbol{c}_{+i})^{-1} = -\left[-1+\boldsymbol{v}(c,i)_i\right]^{-1}\boldsymbol{v}(c,i)\left[\boldsymbol{G}(c)^{-1}\right]_{(i,\cdot)} + \boldsymbol{G}(c)^{-1}. \tag{36}$$

同时更新变量l的表达形式如下:

$$\boldsymbol{l}[+i] = l - \boldsymbol{l}_i\left[-1+\boldsymbol{v}(c,i)_i\right]^{-1}\boldsymbol{v}(c,i). \tag{37}$$

对于另一种情况$i\in c$,则是将i从原有的活动约束的集合c中去除,同时更新可逆矩阵$\boldsymbol{G}(\boldsymbol{c}_{-i})$和变量$l$如下:

$$\boldsymbol{G}(\boldsymbol{c}_{-i})^{-1} = -\left[1+\boldsymbol{v}(c,i)_i\right]^{-1}\boldsymbol{v}(c,i)\left[\boldsymbol{G}(c)^{-1}\right]_{(i,\cdot)} + \boldsymbol{G}(c)^{-1}, \tag{38}$$

$$\boldsymbol{l}[-i] = l - \boldsymbol{l}_i\left[1+\boldsymbol{v}(c,i)_i\right]^{-1}\boldsymbol{v}(c,i). \tag{39}$$

通过上述流程,可以在活动约束集合c的两种情况下对c添加或删除约束,从而更新对应的矩阵$\boldsymbol{I}_c$,最后使$(l,\boldsymbol{I}_c)$相容,再利用式(24)和式(33)得到系统控制量$\boldsymbol{z}$关于变量l的隐式方程的解如定理2所示,并作用于系统,实现对车辆的协调控制.

3 仿真验证

为验证本文提出的快速协调控制算法的稳定性能和求解速度,利用MATLAB/Simulink与CarSim软件搭建联合仿真平台,设置车辆参数,如表1所示,在双移线、蛇形工况下,针对两种不同求解方法的MPC控制器进行对比实验:①采用传统有效集法进行求解的MPC控制器(简称为传统MPC控制器);②利用斜坡函数快速求解的MPC控制器(简称为快速MPC控制器).

表1 车辆参数

Table 1 Vehicle parametrers

参数	数值
整车质量m/kg	1 412
质心到前轴距离a/m	1.015
质心到后轴距离b/m	1.895
质心高度h/m	0.54
前轮轮距d_f/m	1.65
后轮轮距d_r/m	1.65
车轮有效滚动半径R/m	0.325
绕z轴转动惯量I_z/(kg·m^2)	1 536.7

3.1 双移线工况实验

采用标准的双移线(DLC)工况模拟一般行驶工况,设置路面附着系数为0.5,纵向车速为80 km/h.车辆前轮转角输入如图2所示,仿真结果如图3~图6所示.由图3可知,在一般行驶工况下,快速MPC控制器能更好的接近预定轨迹.图4和图5分别是车辆的质心侧偏角和横摆角速度响应对比曲线,可以看出,快速MPC控制器的质心侧偏角最大值为0.03 rad,横摆角速度最大值为0.184 rad/s,相较于传统MPC控制器,质心侧偏角的幅值缩小了14.3%,横摆角速度整体趋势也更接近期望值.图6为车辆的质心侧偏角—质心侧偏角速度相平面图,相平面的面积大小表现了系统的稳定性,对比相平面面积可知,在快速MPC控制器的作用下系统有更好的稳定性.

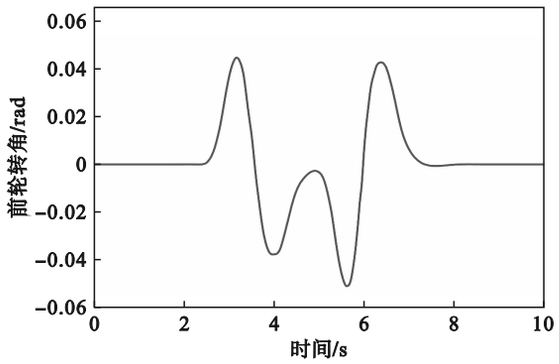


图2　双移线工况下前轮转角输入

Fig. 2　Front wheel steering angle input under double lane change maneuver

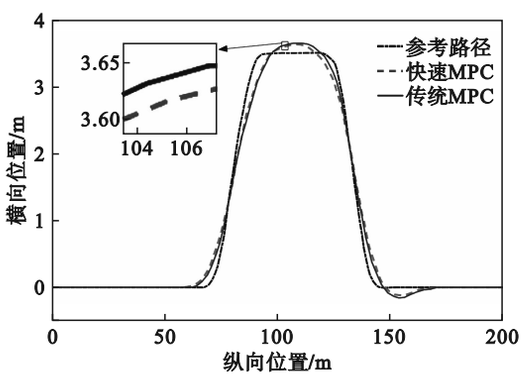


图3　双移线工况下车辆位置

Fig. 3　Vehicle displacement under double lane change maneuver

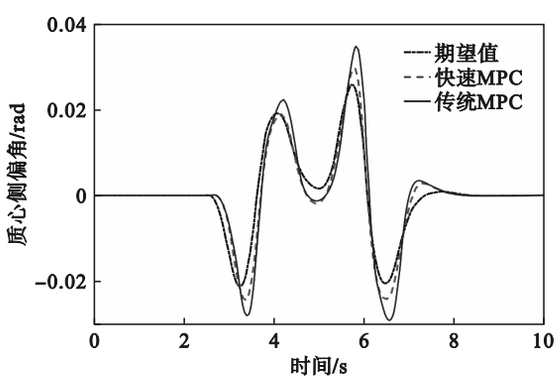


图4　双移线工况下车辆质心侧偏角

Fig. 4　Vehicle sideslip angle under double lane change maneuver

3.2　蛇形工况实验

为进一步验证快速协调控制算法的有效性，选择连续变道的蛇形工况来模拟极限驾驶工况，设置路面附着系数为0.85，纵向车速为100 km/h. 车辆前轮转角输入如图7所示，仿真结果如图8~图11所示. 如图8所示，在此工况下，快速MPC控制器的横向位置误差更小，能更好跟踪目标路径.

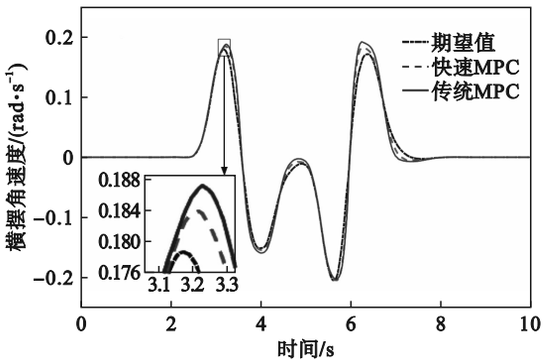


图5　双移线工况下车辆横摆角速度

Fig. 5　Vehicle yaw rate under double lane change maneuver

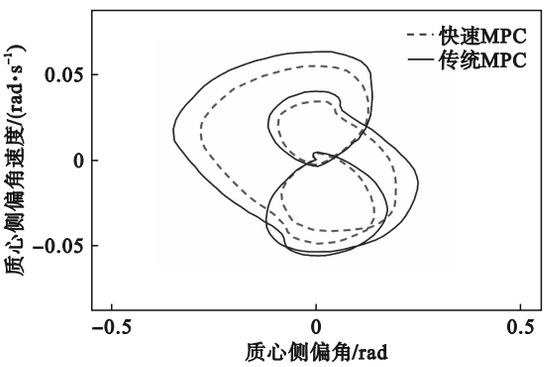


图6　双移线工况下相平面图

Fig. 6　Phase plane diagram under double lane change maneuver

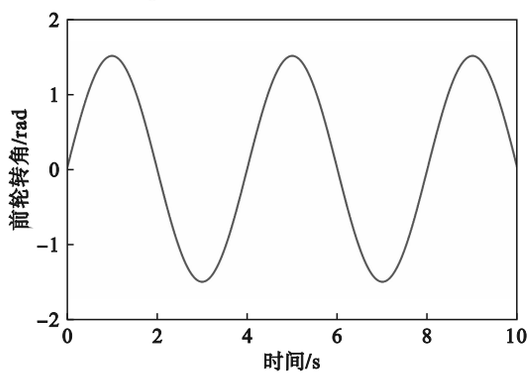


图7　蛇形工况下前轮转角输入

Fig. 7　Front wheel steering angle input under slalom maneuver

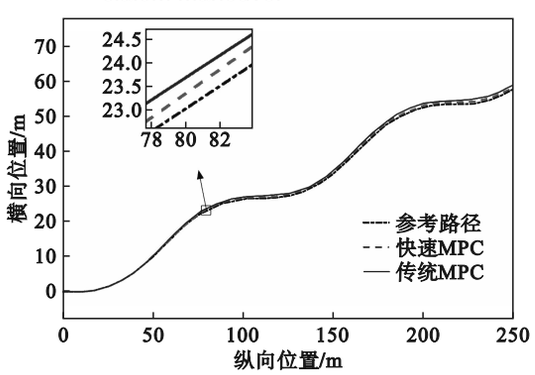


图8　蛇形工况下车辆位置

Fig. 8　Vehicle displacement under slalom maneuver

由图9可以看出,在快速MPC控制器的作用下车辆的质心侧偏角更接近车辆在目标轨迹下的期望值,而且在幅值误差上也小于传统MPC控制器.图10对比了横摆角速度与期望值的偏差,可以看出,快速MPC控制器的横摆角速度更接近设计的期望值,相较于传统MPC控制器有更好的车辆性能.图11为车辆的质心侧偏角-质心侧偏角速度相平面图,由图可知,相较于传统MPC控制器,快速MPC控制器的相平面更接近稳定中心,并且其面积更小,表明快速MPC控制器在蛇形工况下有更好的稳定性表现.

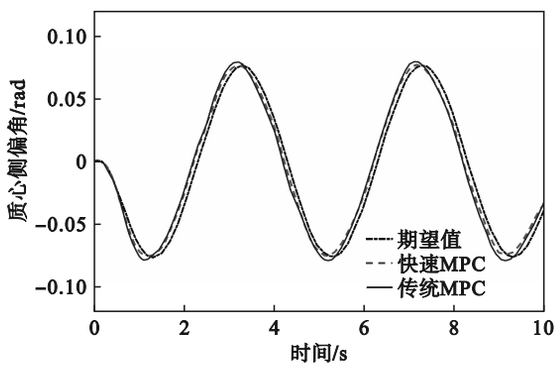


图9 蛇形工况下车辆质心侧偏角

Fig. 9 Vehicle sideslip angle under slalom maneuver

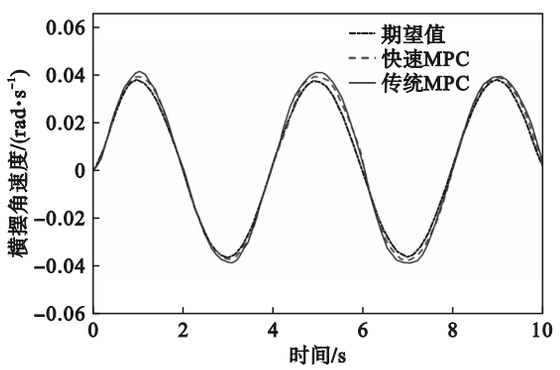


图10 蛇形工况下车辆横摆角速度

Fig. 10 Vehicle yaw rate under slalom maneuver

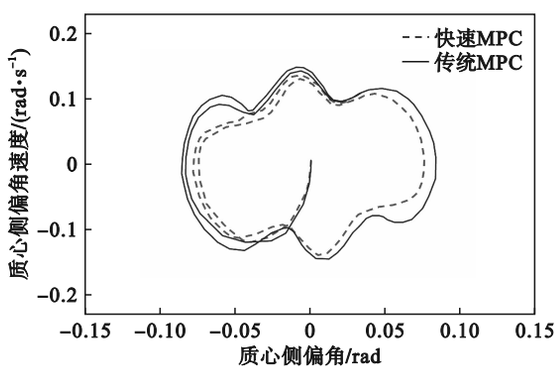


图11 蛇形工况下相平面图

Fig. 11 Phase plane diagram under slalom maneuver

3.3 计算效率验证

为验证快速协调策略的实时性改进效果,利用两种控制器分别完成DLC工况和蛇形工况下的稳定性控制,多次实验得到结果如表2和表3所示.快速MPC控制器在两种工况下,平均计算时间分别比传统MPC控制器减少了1.421 ms和2.235 ms,即在两种工况下其计算效率分别提高32.78%和38.06%;对比两种工况下单次最大计算时间,快速MPC控制器分别比传统MPC控制器减少6.888 ms和9.352 ms,即单次最大计算时间分别减小56.85%和58.08%.在实际道路上行驶过程中将面对更复杂环境的考验,对控制器的实时性有着极高的要求,结果表明,快速MPC控制器大大提高车辆控制系统的响应速度,增强了控制器的实时性,有利于车辆在不同驾驶条件下的稳定行驶.

表2 双移线工况下计算时间

Table 2 Calculation time under double lane change maneuver

控制器类型	平均计算时间/ms	单次最大计算时间/ms
传统MPC	4.336	12.116
快速MPC	2.915	5.228

表3 蛇形工况下计算时间

Table 3 Calculation time under slalom maneuver

控制器类型	平均计算时间/ms	单次最大计算时间/ms
传统MPC	5.873	16.101
快速MPC	3.638	6.749

4 结 语

本文以多约束七自由度非线性车辆模型为研究对象,设计一种基于MPC算法的AFS与DYC快速协调控制策略.针对MPC计算求解速度慢的问题,利用拉盖尔函数构建二次规划问题的KKT条件,通过斜坡函数将其转化为隐式方程,利用改进的有效活动约束集法对其进行求解,从而得到车辆系统最优控制序列.利用MATLAB/Simulink与CarSim进行联合仿真实验,结果表明基于斜坡函数的快速MPC控制器相较于传统MPC控制器,具有更佳的控制性能和稳定性.此外,在两种工况下平均求解速率分别提高了32.78%和38.06%,通过提高控制器的求解速率来增强系统的实时性,有利于车辆在不同驾驶条件下稳定行驶.

参考文献:

[1] 徐向阳, 李光远, 陶斯友, 等. 四轮独立驱动电动汽车轮胎纵横向滑移能耗仿真分析[J]. 机械工程学报, 2021, 57(4): 92-102.
(Xu Xiang-yang, Li Guang-yuan, Tao Si-you, et al. Simulation and analysis on longitudinal and lateral slipping energy consumption of four-wheel independently driven electric vehicle tires [J]. *Journal of Mechanical Engineering*, 2021, 57(4): 92-102.)

[2] Sun X Q, Wang Y L, Quan Z Q, et al. DYC design for autonomous distributed drive electric vehicle considering tire nonlinear mechanical characteristics in the PWA form [J]. *IEEE Transactions on Intelligent Transportation Systems*, 2023, 24(10): 11030-11046.

[3] Elhefnawy A, Ragheb H, Sharaf A M, et al. Optimised coordinated chassis stability control using fuzzy genetic algorithm [J]. *International Journal of Heavy Vehicle Systems*, 2020, 27(6): 800-816.

[4] Lin J M, Zou T, Su L, et al. Optimal coordinated control of active front steering and direct yaw moment for distributed drive electric bus[J]. *Machines*, 2023, 11(6): 640.

[5] Liang J H, Lu Y B, Wang F A, et al. A robust dynamic game-based control framework for integrated torque vectoring and active front-wheel steering system[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2023, 24(7):7328-7341.

[6] Liu H, Liu C, Han L J, et al. Handling and stability integrated control of AFS and DYC for distributed drive electric vehicles based on risk assessment and prediction[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2022, 23(12): 23148-23163.

[7] Yaakoubi H, Haggège J, Rezk H, et al. Explicit hybrid MPC for the lateral stabilization of electric vehicle system [J]. *Energy Reports*, 2024, 11: 1100-1111.

[8] Cipolla S, Gondzio J. Proximal stabilized interior point methods and low-frequency-update preconditioning techniques [J]. *Journal of Optimization Theory and Applications*, 2023, 197(3): 1061-1103.

[9] Liu Y J, Yu J. A semismooth Newton based dual proximal point algorithm for maximum eigenvalue problem [J]. *Computational Optimization and Applications*, 2023, 85(2): 547-582.

[10] Valmorbida G, Hovd M. Quadratic programming with ramp functions and fast online QP-MPC solutions [J]. *Automatica*, 2023, 153: 111011.

[11] Scheffe P, Henneken T M, Kloock M, et al. Sequential convex programming methods for real-time optimal trajectory planning in autonomous vehicle racing[C]// IEEE Intelligent Vehicles Symposium (IV). Jeju: IEEE, 2024: 3144.

[12] Zhou T, Gao R Z, Sun Z Q, et al. Workflow-based fast model predictive cloud control method for vehicle kinematics trajectory tracking problem [J]. *IEEE Transactions on Vehicular Technology*, 2023, 72(12): 15365-15374.