

基于改进 GMRES 算法的水冷壁运动机器人路径跟踪模型预测控制^①黄巍^② 占红武^③ 胥芳

(浙江工业大学特种装备制造与先进加工技术教育部/浙江省重点实验室 杭州 310023)

摘要 针对水冷壁爬壁机器人(WWCB)路径跟踪问题,提出一种快速非线性模型预测控制(NMPC)算法。定义 WWCB 与虚拟目标的跟踪误差,将跟踪问题转化为 Bolza 形式的 NMPC 最优控制问题。通过引入半平滑函数,对不等式进行优化处理,构建最优化必要条件。基于庞特里亚金极小值原理建立广义哈密顿函数,采用连续/广义极小残量算法(C/GMRES)计算求解,并引入多重打靶法提高运算精度。同时对 GMRES 算法残量求解方式进行了优化以提高运行效率。对伯努利双纽线和圆形路径进行了跟踪仿真,验证了本文的跟踪控制快速算法的有效性。

关键词 水冷壁爬壁机器人(WWCB);非线性模型预测控制(NMPC);半平滑函数;连续/广义极小残量算法(C/GMRES);多重打靶法;路径跟踪

0 引言

近年来,非完整轮式移动机器人^[1](non-holonomic wheeled mobile robot, NWMR)在运输、探测、救援等领域引起了广泛的关注。如在火电厂中,其应用范围覆盖了水冷壁面巡查检测、缺陷定位^[2]和主辅机关键部件裂化评估维护^[3]。人们对 NWMR 的运动控制^[4]进行了更加深入的研究,提出了各种控制方法,其中包括神经网络控制法^[5-6]、滑膜控制^[7]和非连续反馈控制法^[8]。但当考虑控制输入以及约束处理时,以上算法很难达到较为直观的效果。然而,非线性模型预测控制法(nonlinear model predictive controller, NMPC)却可以很好地解决这个问题,该控制方法以当前时间间隔下 NWMR 状态的预测量为初始条件,基于反馈控制律对系统的未来轨迹进行预测,利用控制序列对性能指标函数进行优化。在此框架下构造的最优化问题具有高度的灵活性^[9]、良好的动态性及鲁棒性,因此在处理多变量多约束问题时表现优秀,吸引了大量研究者的目

光。文献[10]设计了一种非线性模型预测控制算法(NMPC)用于解决误差动力学的轨迹跟踪控制问题。文献[11]针对自主水面舰艇的非线性 MPC 轨迹跟踪控制问题,为提高算法效率引入了启发式方法来处理输入约束。文献[12]针对 NMPC 求解优化问题的滚动可行性问题,将参考轨迹作为性能指标的终端约束以此保证算法收敛。

然而,NMPC 所带来的高计算负荷不可避免地及较小的采样周期产生冲突。为了提高运算速度,满足实时性要求,许多研究者都致力设计有效的在线算法,诸如快速梯度下降^[13]、计算除法^[14]、延迟补偿^[15]及数值连续法^[16]。作为结合连续法^[17]和 Krylov 子空间法^[18]的算法,连续/广义极小残余算法^[19](continuation/general minimum residual, C/GMRES)已经发展成为一种求解 NMPC 的快速算法。该算法通过确定控制序列在每个时间间隔的增量,而非迭代求解,提高了运算速率。

本文研究了水冷壁爬壁机器人(water wall climbing robot, WWCR)的非线性模型预测控制

① 国家重点研发计划(2018YFB1309404, 2018YFB1309401)资助项目。

② 男,1994年生,硕士生;研究方向:机电一体化控制,数字邮政装备;E-mail: 15728043728@sina.cn

③ 通信作者,E-mail: waltzhan@hotmail.com

(收稿日期:2020-07-15)

(NMPC)路径跟踪问题。利用 WWCR 实时状态与虚拟目标状态之间的误差为模型,建立路径跟踪问题 NMPC 框架,基于庞特里亚金极小值原理,转换为 Bolza 形式的 NMPC 最优控制问题。鉴于文献[20]的 C/GMRES 算法同时具备数值连续法与 Krylov 子空间方法的优点,引入该算法对 NMPC 进行计算求解。然而其算法未能完全考虑 NWMR 路径跟踪问题中状态与输入向量不等式约束,本文通过引入正则平滑函数(Fischer-Burmeister, FB),将不等式约束与其互补条件联立构建最优化条件方程,引入多重打靶法,将误差分布到状态向量与共态向量之中,以此提升算法的控制精度。在求解过程中,对约束和控制输入向量进行延拓,提出一种压缩变量的优化方法,对方程进行简化处理,减小线性方程大小,同时对 C/GMRES 算法中的残量求取进行优化,避免 Hessenberg 矩阵的分解运算,降低了算法的计算负载,加快算法运行速度。最后进行仿真,验证该算法的可行性。

1 非线性模型

本文实验对象主要针对履带式爬壁机器人(NWMR),定义图 1 所示的世界坐标系 $\{q\}$, 建立路径跟踪模型,工作环境如图 2 所示。

定义 NWMR 所处位置为 (s_q, y_q) , 其速度为 v , 角速度为 ω 。

$$\dot{\mathbf{X}} = \begin{bmatrix} \dot{s}_q \\ \dot{y}_q \\ \dot{\theta}_q \end{bmatrix} = \begin{bmatrix} v \cos \theta_q \\ v \sin \theta_q \\ \omega \end{bmatrix} = \begin{bmatrix} \cos \theta_q & 0 \\ \sin \theta_q & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (1)$$

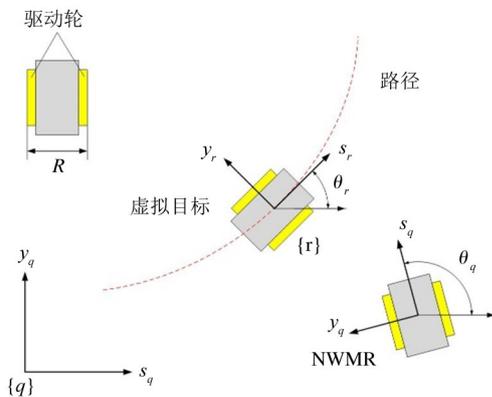


图 1 路径跟踪模型

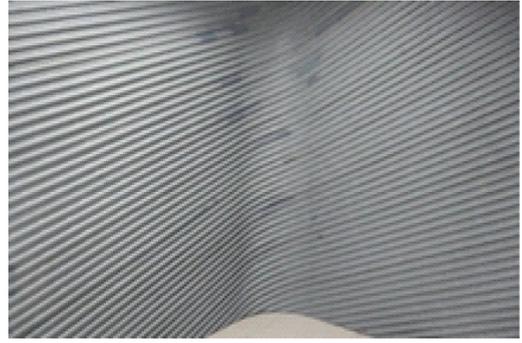


图 2 水冷管壁图

其中 $\mathbf{X} = [s_q \ y_q \ \theta_q]^T$ 为状态向量, $\mathbf{U} = [v \ \omega]^T$ 为控制输入向量。

定义虚拟目标所处坐标系为 $\{r\}$, 需要跟踪的路径为 $[s_r \ y_r \ \theta_r]^T$ 。 $\mathbf{U}_r = [v_r \ \omega_r]$ 表示参考控制信号矢量。考虑到在实际跟踪时由于外部因素导致的局部跟踪误差, 定义误差状态由 $\mathbf{X}_e = [s_e \ y_e \ \theta_e]^T$ 表示, 由此可得 NWMR 的动态误差模型:

$$\begin{cases} \dot{s}_e = \omega y_e - v + v_r \cos \theta_e \\ \dot{y}_e = -\omega x_e + v_r \sin \theta_e \\ \dot{\theta}_e = \omega_r - \omega \end{cases} \quad (2)$$

考虑到动力学方程:

$$\begin{cases} \dot{v} = \frac{f_l + f_r}{mR} \\ \dot{\omega} = \frac{L(f_l - f_r)}{IR} \end{cases} \quad (3)$$

其中, f_l 与 f_r 分别表示左右履带电机的输入驱动转矩, m 与 I 分别为 NWMR 的质量及其惯性矩。在实际工作中, 考虑到由于外部因素导致速度突变引起的运行不稳定, 利用虚拟控制量表示 NWMR 的加速度与角加速度, 对动力学系统简化为

$$\dot{v} = u_1, \dot{\omega} = u_2 \quad (4)$$

根据式(2)~式(4)扩展系统状态, 可得闭环系统状态方程为

$$\begin{cases} \dot{s} = \omega y - v + v_r \cos \theta \\ \dot{y} = -\omega s + v_r \sin \theta \\ \dot{\theta} = \omega_r - \omega \\ \dot{v} = u_1 \\ \dot{\omega} = u_2 \end{cases} \quad (5)$$

令 $\mathbf{x} = [s, y, \theta, v, \omega]^T$ 为 NWMR 的状态向量, 控制输入向量为 $\mathbf{u} = [u_1, u_2]^T$ 。

则式(5)中的闭环系统可用如下非线性表达式表示。

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \quad (6)$$

根据上述建立的模型, 便可将路径跟踪问题转换为寻根问题, 即通过寻找相对最优的虚拟输入 \mathbf{u} 合适的控制率, 满足 $\mathbf{X}_e = 0$ 。

2 快速非线性模型预测控制方法

基于目标函数的内点^[21] (inner point, IP) 重新构造的转换方法, 将潜在的优化问题转化为等式约束进行求解, 是处理 NMPC 的常用方法之一。虽然在一定程度上导致了两点边值问题 (twopoint boundary value problem, TP-BVP), 但是这种基于 PMP 的间接法仍然是求解 OCP 问题较为有效且精度较高的方法。作为一种非线性滚动时域控制的快速数值算法, GMRES 算法避免了传统的牛顿型数值算法中复杂的 Riccati 微分方程, 其基于序列公式, 在每个采样时间仅进行一次求解, 同时利用前向差分替代部分方程近似值, 以此来提高运算速度。针对算法内部的标准基构造进行改进, 加快了计算速度。对于约束问题, 引入 Fischer-Burmeister 正则化平滑算法 (FBRS)^[22], 利用 FB 函数将最优性的必要条件映射到非光滑方程组, 以增加算法的稳定性。融合多重打靶法来提高计算精度, 并对残量求取进行优化处理。

2.1 问题描述

针对模型建立的 NMPC 问题, 在不失一般性的情况下, 可将其描述为 Bolza 形式的最优控制问题, 同时考虑在预测范围内的阶段目标函数 L 以及出现在范围末端的终端目标函数 V 。

$$J = \min_{\mathbf{u}} V(\mathbf{x}(t+T)) + \int_t^{t+T} L(\mathbf{x}(\tau), \mathbf{u}(\tau)) d\tau \quad (7)$$

s. t.

$$\frac{\partial \mathbf{x}}{\partial \tau}(t) = f(\mathbf{x}(t), \mathbf{u}(t)) \quad (8)$$

$$\mathbf{x}(t_0) = \mathbf{x}_0, \mathbf{x}(t_f) = \mathbf{x}_f \quad (9)$$

$$g(\mathbf{x}, \mathbf{u}) \leq 0 \quad (10)$$

其中, 式(9)作为边界条件, \mathbf{x}_0 和 \mathbf{x}_f 分别表示 NMWR 的初始状态和终止状态。式(10)表示约束, 考虑虚拟输入向量 \mathbf{u} 的引入, 定义非线性系统的不等式约束 $g(\mathbf{x}, \mathbf{u})$ 如下所示。

$$\begin{cases} v_{\min} \leq v \leq v_{\max} \\ |\omega| \leq \omega_{\max} \\ u_{1\min} \leq u_1 \leq u_{1\max} \\ u_{2\min} \leq u_2 \leq u_{2\max} \end{cases} \quad (11)$$

式(7)中, $V(\mathbf{x}(t+T))$ 表示终端约束, 表达式为

$$V(\mathbf{x}(t+T)) = \frac{1}{2} [(\mathbf{x}(t+T) - \mathbf{x}_f)^T \mathbf{Q}_f (\mathbf{x}(t+T) - \mathbf{x}_f)] \quad (12)$$

其中, \mathbf{Q}_f 为终端状态权重矩阵。右侧第 2 项为拉格朗日被积函数, 表达式为

$$L(\mathbf{x}(\tau), \mathbf{u}(\tau)) = \frac{1}{2} [(\mathbf{x} - \mathbf{x}_f)^T \mathbf{Q} (\mathbf{x} - \mathbf{x}_f) + \mathbf{u}^T \mathbf{R} \mathbf{u}] \quad (13)$$

上式中 $\mathbf{Q} = \text{diag}\{q_1, \dots, q_5\}$ 、 $\mathbf{R} = \text{diag}\{r_1, r_2\}$ 均表示权重矩阵, q_i 、 r_i 为非负常数。在上述性能指标函数中, 第 1 项表示 NWMR 系统模型, 第 2 项为其边界条件。

2.2 不等式约束

针对不等式约束, 对计算求解带来很大的负担, 许多文献通过各种办法将不等式转变为等式约束来计算, 例如引入外部障碍函数^[23]、势垒函数^[24]、辅助变量法^[25]。本文通过引入 FBRS, 同时使用来自对偶空间的信息生成不同的搜索方向, 即算法同时在原始空间和对偶空间中运行找到原始 OCP 及其对偶的解。使用非线性互补 (NCP) 函数将互补条件式(10)映射到方程组。定义广义半光滑函数为

$$\phi_{SFB}(a, b) = \sqrt{a^2 + b^2 + \varepsilon} - (a + b) \quad (14)$$

其中 $\varepsilon > 0$, 为正则参数。将拉格朗日算子 μ 及不等式约束 $g(\mathbf{x}, \mathbf{u})$ 作为互补条件代入半光滑函数, 得到:

$$\phi_{SFB}(\mu_j, -g_j(\mathbf{x}, \mathbf{u})) = 0 \quad (15)$$

$$\text{s. t. } g_j(\mathbf{x}, \mathbf{u}) < 0$$

需要注意的是附加的拉格朗日乘子 μ 必须加入到输入变量 U_i 中迭代计算, 记为

$$\boldsymbol{\mu} = [\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\mu}_3, \boldsymbol{\mu}_4] = [u_3, u_4, u_5, u_6] \quad (16)$$

为避免当虚拟输入变量为 0 时导致的 C/GMRES 优化更新时的奇异性,需要在代价函数中引入惩罚因子,则拉格朗日函数修改为

$$L(x(\tau), u(\tau)) = \frac{1}{2}[(x - x_f)^T Q(x - x_f) + u^T R u - \sum_{i=3}^6 k_i u_i] \quad (17)$$

为避免惩罚因子过大而对寻找最优解产生影响,故选择 k_i 为小的正常数。

2.3 最优化条件

通过 Karush-Kuhn-Tucker (KKT) 最优性条件导出求解动态 OCP 的最优性条件,利用广义哈密顿函数表示:

$$H(x, \boldsymbol{\lambda}, u) = L(x, u) + \boldsymbol{\lambda}^T f(x, u) + \boldsymbol{\mu}^T g(x, u) \quad (18)$$

其中, $\boldsymbol{\lambda}$ 表示共态向量。为了解决 TP-BVP 中,控制输入函数为未知量的情况,将预测范围 T 分为 N 个阶段,达到利用有限个参数来表示控制输入函数的目的。使用欧拉-拉格朗日方程变分法来离散化求解动态 OCP 最优化的必要条件。离散化如下:

$$0 = x_i + \nabla_{\boldsymbol{\lambda}} H(x_i, \boldsymbol{\lambda}_{i+1}, u_i, \boldsymbol{\mu}_i) \Delta \tau - x_{i+1} \quad (19)$$

$$0 = x_0(t) - x(t) \quad (20)$$

$$0 = \boldsymbol{\lambda}_{i+1} + \nabla_x H(x_i, \boldsymbol{\lambda}_{i+1}, u_i, \boldsymbol{\mu}_i) \Delta \tau - \boldsymbol{\lambda}_i \quad (21)$$

$$0 = \nabla_u H(x_i, \boldsymbol{\lambda}_{i+1}, u_i, \boldsymbol{\mu}_i) \quad (22)$$

$$0 = \boldsymbol{\lambda}_N - \nabla_x V(x_N(t)) \quad (23)$$

$$0 \geq g(x_i, u_i) \quad (24)$$

其中 $\Delta \tau = T/N$ 。分别用 $\{x_i(t)\}_{i=0}^N$ 、 $\{\boldsymbol{\lambda}_i(t)\}_{i=0}^N$ 和 $\{u_i(t)\}_{i=0}^N$ 表示在离散视界上的状态序列、共态序列以及控制输入序列。通过以上离散化的方法,成功地将 NMPC 表示为时间 t 时测量状态 $x(t)$ 的离散时间 TP-BVP。

重新定义一个由拉格朗日乘子 $\boldsymbol{\mu}$ 和输入向量组成的离散化的控制输入向量 $\boldsymbol{U}(t) \in \mathbb{R}^{m \times N}$:

$$\boldsymbol{U}(t) = [u_1^T(t) \quad \cdots \quad u_{N-1}^T(t)]^T \quad (25)$$

将 FB 函数作为互补条件,与最优化条件组成约束条件为

$$\boldsymbol{F}(\boldsymbol{U}, \boldsymbol{X}, \boldsymbol{x}, t) = \begin{bmatrix} \nabla_u H^T(x_0, \boldsymbol{\lambda}_0, u_0) \\ \phi_{SFB}(\boldsymbol{\mu}, -g(x_0, u_0)) \\ \vdots \\ \nabla_u H^T(x_{N-1}, \boldsymbol{\lambda}_{N-1}, u_{N-1}) \\ \phi_{SFB}(\boldsymbol{\mu}, -g(x_{N-1}, u_{N-1})) \end{bmatrix} = 0 \quad (26)$$

相比于将误差仅分布在 $\boldsymbol{F} = 0$ 中,多重打靶法倾向于将误差同时分布在状态向量和共态向量中时,预计将会减小整体误差,即算法的精度会提高,有利于非线性系统的稳定性及收敛性。使用多重打靶算法时,需要将状态向量 \boldsymbol{x} 、拉格朗日乘子 $\boldsymbol{\mu}$ 、共态向量 $\boldsymbol{\lambda}$ 以及控制输入向量 \boldsymbol{u} 视为独立变量。将连续状态向量和共态向量组合构建出一个新的向量 $\boldsymbol{X}(t) \in \mathbb{R}^{2nN}$:

$$\boldsymbol{X}(t) = [x_1^T(t) \quad \boldsymbol{\lambda}_1^T(t) \quad \cdots \quad x_N^T(t) \quad \boldsymbol{\lambda}_N^T(t)]^T \quad (27)$$

建立状态和共态向量的代数约束:

$$\boldsymbol{G}(\boldsymbol{U}, \boldsymbol{X}, \boldsymbol{x}, t) = \begin{bmatrix} x_1 - x_0 - f(x_0, u_0) \Delta \tau \\ \boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2 - (\nabla_x H)_1^T \Delta \tau \\ \vdots \\ x_N - x_{N-1} - f(x_{N-1}, u_{N-1}) \Delta \tau \\ \boldsymbol{\lambda}_{N-1} - \boldsymbol{\lambda}_N - (\nabla_x H)_{N-1}^T \Delta \tau \\ \boldsymbol{\lambda}_N - \nabla_x V(x_N) \end{bmatrix} = 0 \quad (28)$$

通过建立上述约束,可以保证系统总是相对之前获得的预测条件进行计算,并且其初始状态不会在整个时域 T 上进行积分。

2.4 优化求解

针对方程组式(26)和式(28),利用 t 为参数的连续法^[26],同时进行延拓。首先需对约束条件 \boldsymbol{F} 与控制输入向量 \boldsymbol{U} 进行拆分处理:

$$\boldsymbol{F} = [\bar{\boldsymbol{F}}^T \quad \hat{\boldsymbol{F}}^T]^T, \quad \boldsymbol{U} = [\bar{\boldsymbol{U}}^T \quad \hat{\boldsymbol{U}}^T]^T \quad (29)$$

同时,拆分时需要满足以下 3 个条件:

(1) 作为消去 $\dot{\boldsymbol{U}}$ 与 $\dot{\boldsymbol{X}}$ 的必要条件之一, $\bar{\boldsymbol{F}}$ 和 $\bar{\boldsymbol{U}}$ 的维数必须相同。

(2) 在分解时,约束 \boldsymbol{G} 中不包含 $\hat{\boldsymbol{U}}$ 中的任何元素,约束 $\hat{\boldsymbol{F}}$ 中不包含任何关于 \boldsymbol{X} 的元素。且在连续

状态方程中没有 \hat{U} 中的任何元素。在哈密顿量 H 中不包含与状态向量交叉的项,对应此式即为在状态方程中不存在虚拟输入 u 及拉格朗日乘子 μ 。

(3) 为了在某一时间步的输入向量 u 及拉格朗日乘子 μ 仅在其对应时间步的哈密顿量 H 中出现,要求当 $\hat{F}_{\hat{U}}^{-1}$ 与任意向量点乘时有解,且计算边界较小。

$$\bar{F}(\bar{U}, \hat{U}, X, x, t) = 0 \quad (30)$$

$$\hat{F}(\bar{U}, \hat{U}, x, t) = 0 \quad (31)$$

$$G(\bar{U}, X, x, t) = 0 \quad (32)$$

对上述的3个约束使用延拓法进行关于 t 的延拓,带入初始条件 $\bar{U}(0)$ 、 $\hat{U}(0)$ 及 $X(0)$ 进行积分操作求解,得到:

$$\bar{F}_{\bar{U}} \dot{\bar{U}} + \bar{F}_{\hat{U}} \dot{\hat{U}} + \bar{F}_X \dot{X} + \bar{F}_x \dot{x} + \bar{F}_t = -A_s \bar{F} \quad (33)$$

$$\hat{F}_{\bar{U}} \dot{\bar{U}} + \hat{F}_{\hat{U}} \dot{\hat{U}} + \hat{F}_x \dot{x} + \hat{F}_t = -A_s \hat{U} \quad (34)$$

$$G_{\bar{U}} \dot{\bar{U}} + G_X \dot{X} + G_x \dot{x} + G_t = -A_s G \quad (35)$$

其中 A_s 是正稳态矩阵。对式(33)~式(35)进行合并化简带入后,得到关于 $\dot{\bar{U}}$ 的线性方程:

$$\begin{aligned} & (\bar{F}_{\bar{U}} - \bar{F}_{\hat{U}} \hat{F}_{\hat{U}}^{-1} \hat{F}_{\bar{U}} - \bar{F}_X G_X^{-1} G_{\bar{U}}) \dot{\bar{U}} \\ & = -A_s \bar{F} - \bar{F}_x \dot{x} - \bar{F}_t - \bar{F}_{\hat{U}} \hat{F}_{\hat{U}}^{-1} (-A_s \hat{F} - \hat{F}_x \dot{x} - \hat{F}_t) \\ & \quad - \bar{F}_X G_X^{-1} (-A_s G - G_x \dot{x} - G_t) \end{aligned} \quad (36)$$

为了避免逆矩阵带来庞大计算量,提出一种消除式(36)中的 G_X^{-1} 矩阵方法。引入一个虚拟变量 $\rho_k \in \mathbb{R}^{2nN}$, 满足 $-G_X^{-1} G_U \bar{u}_k = \rho_k$, 对其进行移项处理后有:

$$G_X \rho_k + G_U \bar{u}_k = 0 \quad (37)$$

计算中,为了减少雅可比矩阵与线性方程相乘时带来的计算量,引入前向差分近似概念:

$$\begin{aligned} F_U W + F_x w + F_t \omega &= D_h F(U, x, t; W, w, \omega) \\ &\approx \frac{F(U + hW, x + hw, t + h\omega) - F(U, x, t)}{h} \end{aligned} \quad (38)$$

式中, h 表示前向差分步长,是一个小的正实数。利用前向差分近似处理式(37)有:

$$G_U \bar{u}_k + G_X \rho_k = \frac{G(U + h\bar{u}_k, X + h\rho_k, x + h\dot{x}, t + h)}{h}$$

$$\frac{G(U, X, x + h\dot{x}, t + h)}{h} = 0 \quad (39)$$

化简可得:

$$\begin{aligned} G(U + h\bar{u}_k, X + h\rho_k, x + h\dot{x}, t + h) \\ = G(U, X, x + h\dot{x}, t + h) \end{aligned} \quad (40)$$

在上述算法中已经提到,多重打靶法相比于原算法的改进之处,在于将误差同时分布到了状态向量和共态向量约束 G 中来减小整体误差。取误差向量表示为 $E = [e_1^T, \dots, e_{2N}^T] \in \mathbb{R}^{2nN}$, $x \in \mathbb{R}^n$, $t \in \mathbb{R}$, 对 X 满足:

$$G(\bar{U}, X, x, t) = E \quad (41)$$

定义一个映射 $B: \mathbb{R}^{mN} \times \mathbb{R}^{2nN} \times \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^{2nN}$, 当 U, x, t 均确定时, B 即为 G 的逆映射。由此,根据 G 的定义可得 $X = B(\bar{U}, E, x, t)$, 且满足:

$$x_{i+1} = x_i + f(x_i, u_i) \Delta\tau + e_{2i+1} \quad (42)$$

$$\begin{aligned} \lambda_{i+1} &= \lambda_{i+2} + (\nabla_x H)^T(x_{i+1}, \lambda_{i+2}, u_{i+1}, \mu_{i+1}) \Delta\tau \\ &\quad + e_{2i+2} \end{aligned} \quad (43)$$

$$\lambda_N = (\nabla_x V)^T(x_N) + e_{2N} \quad (44)$$

通过此方法分别求得 x_i, λ_i 后,即可通过递归计算求得 $B(\bar{U}, E, x, t)$, 结合式(40)可以推导出:

$$\begin{aligned} X + h\rho_k &= B(\bar{U} + h\bar{u}_k, G(\bar{U}, X, x + h\dot{x}, t + h), \\ &\quad x + h\dot{x}, t + h) \end{aligned} \quad (45)$$

对式(45)进行变换即可求得虚拟变量 ρ_k , 即可得到 $-G_X^{-1} G_U \bar{u}_k$ 的值为

$$\begin{aligned} -G_X^{-1} G_U \bar{u}_k &\approx \\ & \frac{B(\bar{U} + h\bar{u}_k, G(\bar{U}, X, x + h\dot{x}, t + h), x + h\dot{x}, t + h) - X}{h} \end{aligned} \quad (46)$$

根据条件式(3)得 $\hat{F}_{\hat{U}}^{-1} \hat{F}_{\bar{U}} \bar{u}_k$ 的计算边界较小, 则式(36)左侧可以前向差分近似为如下形式。

$$\begin{aligned} & (\bar{F}_{\bar{U}} - \bar{F}_{\hat{U}} \hat{F}_{\hat{U}}^{-1} \hat{F}_{\bar{U}} - \bar{F}_X G_X^{-1} G_{\bar{U}}) \bar{u}_k \\ & = \bar{F}_{\bar{U}} \bar{u}_k + \bar{F}_{\hat{U}} (-\hat{F}_{\hat{U}}^{-1} \hat{F}_{\bar{U}} \bar{u}_k) + \bar{F}_X (-G_X^{-1} G_{\bar{U}} \bar{u}_k) \\ & \approx \bar{F}(\bar{U} + h\bar{u}_k, \hat{U} - h\hat{F}_{\hat{U}}^{-1} \hat{F}_{\bar{U}} \bar{u}_k, B(\bar{U} + h\bar{u}_k, G(\bar{U}, X, x + h\dot{x}, t + h), x + h\dot{x}, t + h) / h \\ & \quad - \frac{\bar{F}(\bar{U}, \hat{U}, X, x + h\dot{x}, t + h)}{h} \end{aligned} \quad (47)$$

上式虽然看似比较复杂,其实可以仅仅使用 $\bar{\mathbf{F}}$ 和 \mathbf{B} 直接表示获得。

对于式(36)右侧难点,主要集中在对式 $\mathbf{G}_x^{-1}(-\mathbf{A}_s \mathbf{G} - \mathbf{G}_x \dot{\mathbf{x}} - \mathbf{G}_t)$ 的处理,提出一种新的方法。定义一个虚拟向量 $\boldsymbol{\omega} \in \mathbb{R}^{2nN}$, 满足 $\mathbf{G}_x^{-1}(-\mathbf{A}_s \mathbf{G} - \mathbf{G}_x \dot{\mathbf{x}} - \mathbf{G}_t) = \boldsymbol{\omega}$, 即:

$$\mathbf{G}_x \boldsymbol{\omega} + \mathbf{G}_x \dot{\mathbf{x}} + \mathbf{G}_t = -\mathbf{A}_s \mathbf{G} \quad (48)$$

对式(48)前向差分近似处理,得到:

$$\begin{aligned} & \frac{\mathbf{G}(\bar{\mathbf{U}}, \mathbf{X} + h\boldsymbol{\omega}, \mathbf{x} + h\dot{\mathbf{x}}, t + h) - \mathbf{G}(\bar{\mathbf{U}}, \mathbf{X}, \mathbf{x}, t)}{h} \\ & = -\mathbf{A}_s \mathbf{G}(\bar{\mathbf{U}}, \mathbf{X}, \mathbf{x}, t) \quad (49) \end{aligned}$$

化简后可得:

$$\begin{aligned} & -\mathbf{A}_s \bar{\mathbf{F}} + \bar{\mathbf{F}}_{\hat{\mathbf{U}}}(-\hat{\mathbf{F}}_{\hat{\mathbf{U}}}^{-1} \boldsymbol{\sigma}) + \bar{\mathbf{F}}_x(-\rho) + \bar{\mathbf{F}}_x(-\dot{\mathbf{x}}) + \bar{\mathbf{F}}_t(-1) \\ & \approx -\mathbf{A}_s \bar{\mathbf{F}} - \frac{\bar{\mathbf{F}}(\bar{\mathbf{U}}, \hat{\mathbf{U}} + h\hat{\mathbf{F}}_{\hat{\mathbf{U}}}^{-1} \boldsymbol{\sigma}, \mathbf{X} + h\boldsymbol{\omega}, \mathbf{x} + h\dot{\mathbf{x}}, t + h) - \bar{\mathbf{F}}(\bar{\mathbf{U}}, \hat{\mathbf{U}}, \mathbf{X}, \mathbf{x}, t)}{h} = \frac{1 - h\mathbf{A}_s}{h} \bar{\mathbf{F}}(\bar{\mathbf{U}}, \hat{\mathbf{U}}, \mathbf{X}, \mathbf{x}, t) \\ & - \frac{1}{h} \bar{\mathbf{F}}(\bar{\mathbf{U}}, \hat{\mathbf{U}} + \hat{\mathbf{F}}_{\hat{\mathbf{U}}}^{-1}(\hat{\mathbf{F}}(\bar{\mathbf{U}}, \hat{\mathbf{U}}, \mathbf{x} + h\dot{\mathbf{x}}, t + h) + (h\mathbf{A}_s - 1)\hat{\mathbf{F}}(\bar{\mathbf{U}}, \hat{\mathbf{U}}, \mathbf{x}, t)), \mathbf{B}(\bar{\mathbf{U}}, (1 - h\mathbf{A}_s)\mathbf{G}(\bar{\mathbf{U}}, \mathbf{X}, \mathbf{x}, t), \mathbf{x} \\ & + h\dot{\mathbf{x}}, t + h), \mathbf{x} + h\dot{\mathbf{x}}, t + h) \quad (53) \end{aligned}$$

综上,利用 GMRES 算法对式(36)计算,快速求解获得 $\hat{\mathbf{U}}$ 。根据映射 \mathbf{B} 及式(51),无需通过 GMRES 即可求得:

$\dot{\mathbf{X}} =$

$$\frac{\mathbf{B}(\bar{\mathbf{U}} + h\hat{\mathbf{U}}, (1 - h\mathbf{A}_s)\mathbf{G}(\bar{\mathbf{U}}, \mathbf{X}, \mathbf{x}, t), \mathbf{x} + h\dot{\mathbf{x}}, t + h) - \mathbf{X}}{h} \quad (54)$$

根据式(34)可以解得 $\hat{\mathbf{U}}$:

$$\hat{\mathbf{U}} = \hat{\mathbf{F}}_{\hat{\mathbf{U}}}^{-1}(-\mathbf{A}_s \hat{\mathbf{F}} - \hat{\mathbf{F}}_{\hat{\mathbf{U}}} \hat{\mathbf{U}} - \hat{\mathbf{F}}_x \dot{\mathbf{x}} - \hat{\mathbf{F}}_t) \quad (55)$$

在求得 $\hat{\mathbf{U}}$ 及 $\dot{\mathbf{X}}$ 之后,利用欧拉法(Euier)分别对其积分,更新控制输入 \mathbf{U} 及状态向量 \mathbf{X} 。

2.5 算法优化

多重打靶法的引入提高算法精度的同时,也带入了较为沉重的计算负担,主要集中在对每个时间步上非线性函数的雅可比矩阵的计算,比使用传统的 GMRES 算法计算效率大幅提高。然而,该算法

$$\begin{aligned} & \mathbf{G}(\bar{\mathbf{U}}, \mathbf{X} + h\boldsymbol{\omega}, \mathbf{x} + h\dot{\mathbf{x}}, t + h) \\ & = (1 - h\mathbf{A}_s)\mathbf{G}(\bar{\mathbf{U}}, \mathbf{X}, \mathbf{x}, t) \quad (50) \end{aligned}$$

根据上文所提的映射 \mathbf{B} ,可以解得 $\mathbf{X} + h\boldsymbol{\omega}$:

$$\begin{aligned} \mathbf{X} + h\boldsymbol{\omega} & = \mathbf{B}(\bar{\mathbf{U}}, (1 - h\mathbf{A}_s)\mathbf{G}(\bar{\mathbf{U}}, \mathbf{X}, \mathbf{x}, t), \\ & \mathbf{x} + h\dot{\mathbf{x}}, t + h) \quad (51) \end{aligned}$$

为了简化 $\mathbf{A}_s \hat{\mathbf{F}} + \hat{\mathbf{F}}_x \dot{\mathbf{x}} + \hat{\mathbf{F}}_t$, 需要引入虚拟向量计算,即 $\boldsymbol{\sigma} = \mathbf{A}_s \hat{\mathbf{F}} + \hat{\mathbf{F}}_x \dot{\mathbf{x}} + \hat{\mathbf{F}}_t \in \mathbb{R}^{2nN}$, 对其前向差分近似:

$$\boldsymbol{\sigma} = \frac{1}{h} \hat{\mathbf{F}}(\bar{\mathbf{U}}, \hat{\mathbf{U}}, \mathbf{x} + h\dot{\mathbf{x}}, t + h) + \frac{h\mathbf{A}_s - 1}{h} \hat{\mathbf{F}}(\bar{\mathbf{U}}, \hat{\mathbf{U}}, \mathbf{x}, t) \quad (52)$$

由条件式(3)可知 $\hat{\mathbf{F}}_{\hat{\mathbf{U}}}^{-1} \boldsymbol{\sigma}$ 求解时计算边界很小,则式(36)右侧可以化简为

在对 Hessenberg 矩阵进行正交分解时,依然涉及比较复杂繁琐的计算。因此,此算法在此处仍可进行优化。很多学者都提出了基于 GMRES 算法的改进算法^[27]。然而在一些情况无法避免地出现震荡现象。本文通过对 Krylov 子空间的第 2 个元素开始进行 Arnoldi 过程处理,将原本的 $n \times m$ 维 Hessenberg 矩阵 $\bar{\mathbf{H}}_{k,k}$, 改进为上三角矩阵 \mathbf{W}_k , 避免正交分解处理,就能减少计算量。

考虑非线性方程:

$$\mathbf{A}\mathbf{M} = \mathbf{b} \quad (56)$$

为提高该控制算法的稳定性^[28],可将子空间 $K_k(\mathbf{A}, \mathbf{r}_0) = \text{span}\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \dots, \mathbf{A}^{k-1}\mathbf{r}_0\}$ 的元素,即传统 GMRES 算法的基替换为如下形式:

$$\begin{aligned} \mathbf{U}_k & = (\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_{k-1}) \\ & = \left(\frac{\mathbf{r}_0}{\|\mathbf{r}_0\|_F}, \frac{\mathbf{r}_1}{\|\mathbf{r}_1\|_F}, \dots, \frac{\mathbf{r}_{k-1}}{\|\mathbf{r}_{k-1}\|_F} \right) \quad (57) \end{aligned}$$

其中初始参量为 $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{M}_0$, 其余参量满足 $\mathbf{r}_i = \mathbf{b} - \mathbf{A}\mathbf{M}_i \in K_k(\mathbf{A}, \mathbf{r}_0)$ 。将向量组 \mathbf{U}_k 进行 Arnoldi 处

理:

$$AU_k = V_k W_k \quad (58)$$

式中 $V_k = (v_1, v_2, \dots, v_k)^T$, 为 $AK_k(A, r_0)$ 的标准正交基, 令 $w_{i,j} = (Au_j, v_i)$ 为上三角矩阵, 取 $\|v_{j+1}\|_F = w_{i,j+1}$, 求得的 W_k 为上三角矩阵, 取 $M_k = M_0 + t_k$, 其中 $t_k \in U_k$, 即 $t_k = U_k y_k, y_k \in \mathbb{R}^{k \times 1}$ 。假设经过 k 次迭代后, 有

$$r_k = r_0 - AU_k y_k \quad (59)$$

通过调整 y_k 使得 r_k 与 $AU_k y_k$ 垂直时, $\|r_k\|_F$ 取得最小值, 即

$$V_k^T r_0 = V_k^T AU_k y_k \quad (60)$$

结合式(59)可得到:

$$y_k = W_k^{-1} V_k r_0 \quad (61)$$

求得近似解:

$$M_k = M_0 + U_k W_k^{-1} V_k r_0 \quad (62)$$

改进之后的 GMRES 算法减少了正交分解所带来的额外的计算需求, 提高了算法的运行效率与稳定性。

3 仿真结果

在仿真模拟中使用为个人电脑的 CPU 为英特尔 Core i7-9750H, 6 核, 时钟频率为 4.5 GHz。在 Matlab 2016a 环境下对圆形(固定曲率)与伯努利双纽线(变曲率)路径进行跟踪仿真模拟。

圆形路径定义如下:

$$x = \cos(\varpi) y = \sin(\varpi)$$

式中, ϖ 为路径参数, 确定了虚拟目标在世界坐标系下的位置, 初始状态和终端状态分别选取为

$$x_0 = [-1.1 \ 0 \ 0 \ 0 \ 0]^T$$

$$x_f = [-1 \ 0 \ 0 \ 0.05 \ 0.1]^T$$

初始输入为 $u = [0 \ 0]^T$, 仿真参数设置中, $T = 1 \text{ s}, N = 10$, 采样时间选择为 $\Delta\tau = 0.1$, 权重矩阵取:

$$Q_f = Q = \text{diag}\{15, 15, 1, 0.1, 0.1\}$$

$$R = \text{diag}\{0.1, 0.1\}$$

约束参数中设置 $v_{\min} = 0 \text{ m/s}, v_{\max} = 0.16 \text{ m/s}, \omega_{\max} = 0.8 \text{ rad/s}, u_{1\min} = -0.08 \text{ m/s}^2, u_{1\max} = 0.08 \text{ m/s}^2, u_{2\min} = -0.6 \text{ rad/s}^2, u_{2\max} = 0.6 \text{ rad/s}^2$ 。

伯努利双纽线路径定义为

$$x = \sin(2\varpi) \quad y = 2 \times \sin(\varpi)$$

初始状态与终端状态分别选取为

$$x_0 = [0.3 \ 0.03 \ 1 \ 0 \ 0]^T$$

$$x_f = [0.6 \ 0.6 \ 1 \ 0 \ 0]^T$$

初始输入为 $u = [0 \ 0]^T$, 仿真参数设置中, $T = 1 \text{ s}, N = 10$, 采样时间选择为 $\Delta\tau = 0.1$, 权重矩阵取:

$$Q_f = Q = \text{diag}\{25, 25, 0.01, 0.1, 0.1\}$$

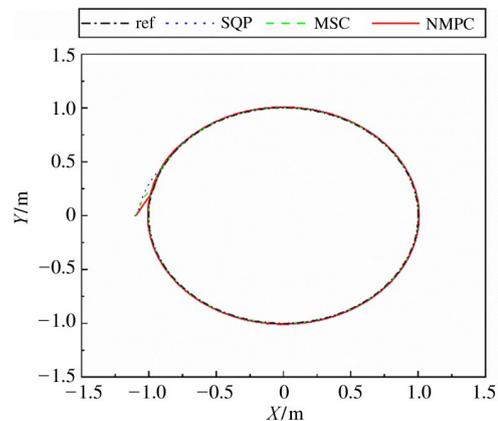
$$R = \text{diag}\{0.1, 0.1\}$$

约束参数中设置 $v_{\min} = 0 \text{ m/s}, v_{\max} = 1 \text{ m/s}, \omega_{\max} = 0.8 \text{ rad/s}, u_{1\min} = -0.2 \text{ m/s}^2, u_{1\max} = 0.3 \text{ m/s}^2, u_{2\min} = -0.4 \text{ rad/s}^2, u_{2\max} = 1 \text{ rad/s}^2$ 。

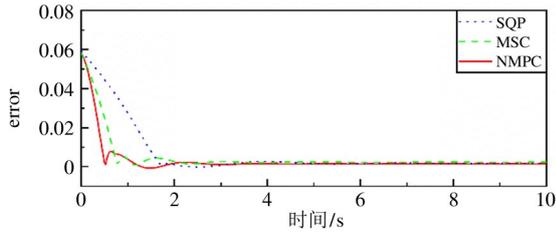
其他相关参数为 $\varepsilon = 0.01, h = 0.001, A_s = D_h = 1/h = 1000, m = 1.5, R = 0.048, I = 0.0348, C/GMRES$ 算法内部参数设置初始控制输入与乘子的误差容限 $rtol = 1.0e - 8$, GMRES 的迭代数 $k_{\max} = 10$ 。

仿真中引入另外 2 种算法与本文所提算法进行对比。第 1 种选择 Matlab 自带的 SQP 算法; 第 2 种为 MSC^[29] 算法, 通过障碍函数法对不等式进行处理; 第 3 种为本文提出的改进算法(NMPC)。其中 ref 表示虚拟目标的参考路径。跟踪误差表示为 $e = \sqrt{s^2 + y^2}$, 最优性误差 $\|F\|$, 定义为式(26)中 F 的 2-范数。

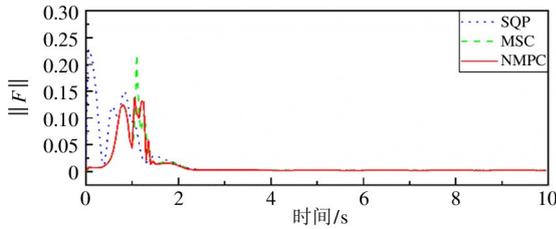
对圆形路径的跟踪仿真中, 设定运行时间为 10 s, 由图 3(b) 可知, 针对固定曲率的路径进行跟踪, 3 种算法都表现出良好的跟踪效果, 在稳定跟踪



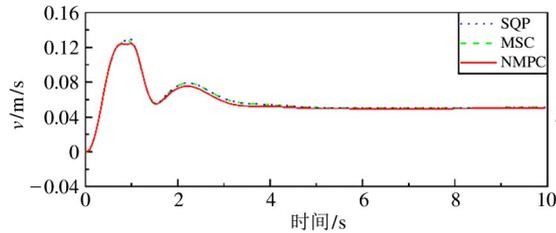
(a) 跟踪路径



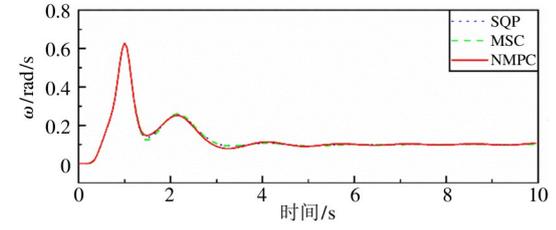
(b) 路径跟踪误差



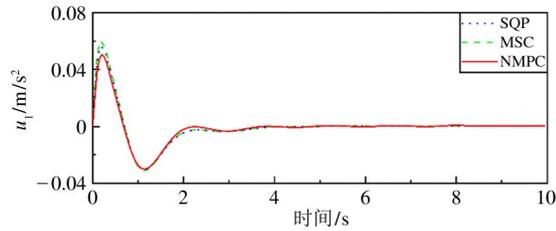
(c) 最优性误差



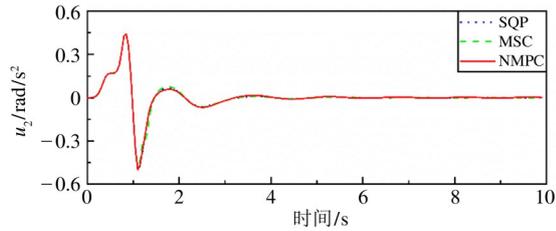
(d) NWMR 线速度



(e) NWMR 角速度



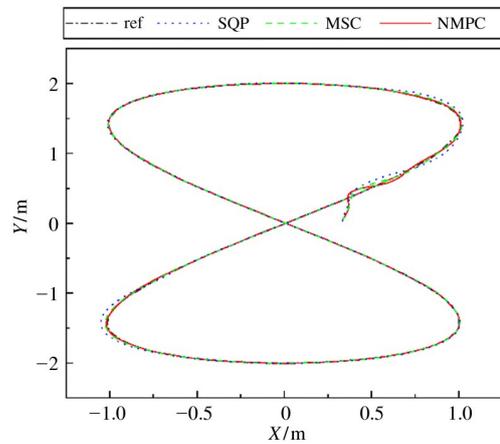
(f) 线加速度



(g) 角加速度

后,跟踪误差均能控制在 0.01 m 范围内。相比之下,本文所述算法的跟踪结果在 0.5 s 时已经跟踪至目标路径且趋于稳定,最优性误差峰值最低,比另外 2 种算法更为敏感。从图 3(d) ~ (g) 可知,跟踪速度与加速度在接近 3 s 时开始收敛,达到期望值,3 种算法表现相当。

对伯努利双纽线路径跟踪仿真中,设定仿真运行时间为 10 s,由图 4(a)和(b)可知,针对变曲率路径 3 种控制效果差异较为明显。SQP 算法在曲率变化较大的区域会出现较为明显的跟踪误差,且其最优性误差峰值较大,对于拐角区域的路径并不敏感。相比较而言 MSC 与本文的算法控制效果更佳,在拐角区域基本可以稳定跟踪。同时针对不同预测步长 N ,对 3 种算法的控制更新平均计算时长和平均误差进行了仿真,结果如表 1 和表 2 所示。在 $N = 5$ 的条件下,MSC 与 SQP 满足在采样周期中进行跟踪。然而,随着采样步长的增加,两种算法的计算速率开始降低,在 $N = 10$ 的条件下,平均计算时间分别达到了 0.132 s 和 0.2134 s,都超过了采样周期,代表着其无法在采样时间要求 0.1 s 内获得 NMPC 的最优控制输入,在实际工作时可能无法准确进行路径跟踪。相较而言,本文算法在计算速率上展现出了优势,即使在采样步长 $N = 50$ 的情况下,依旧可以仅消耗采样时长 10% 以内的时间有效求解 NMPC,同时满足运行的实时性要求,虽然相比较 MSC 其误差有所增加,但依旧满足精度要求。在针对更加复杂的非线性系统时,本文所述算法的提升会更加显著。



(a) 跟踪路径

图 3 圆形路径跟踪仿真结果

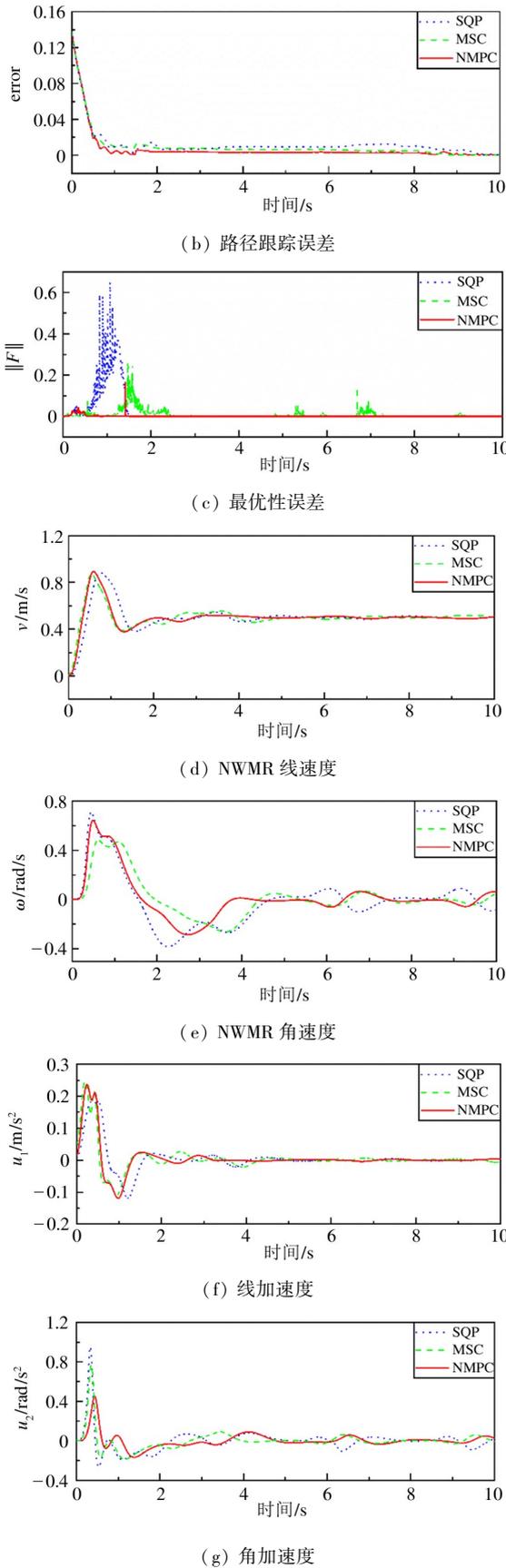


图4 伯努利双扭线路径跟踪仿真结果

表1 平均计算时长

N	SQP/s	MSC/s	NMPC/s
5	0.0523	0.098	0.0009
10	0.2134	0.132	0.0017
30	1.1930	0.271	0.0059
50	2.1698	0.481	0.0092

表2 平均跟踪误差

N	SQP/mm	MSC/mm	NMPC/mm
5	0.1063	0.0943	0.1078
10	0.0406	0.0122	0.0197
30	0.0201	0.0039	0.0086
50	0.0142	0.0012	0.0019

4 结论

针对非完整移动机器人路径跟踪问题,本文通过定义 NWMR 与虚拟目标的跟踪误差,建立 Bolza 形式的 NMPC 最优控制问题框架。引入半平滑函数,将不等式约束融入最优化必要条件之中,通过使用 GMRES 算法计算求解广义哈密顿函数。同时引入多重打靶法及改变残量求取方式以提高运算精度及运算速率。仿真结果表明,本文所述的控制算法可以在满足实时性的前提下,保证更高的计算效率及精度。

然而本文并未对实际环境中的外界因素,如摩擦圆极限条件下的打滑情况^[30]进行考虑,因此针对如何提高该算法的鲁棒性将是未来的研究方向。

参考文献

[1] Hossein M. Optimization-based nonlinear control laws with increased robustness for trajectory tracking of non-holonomic wheeled mobile robots[J]. *Transportation Research Part C: Emerging Technologies*, 2019, 101: 1-17

[2] 胥芳,丁信斌,占红武. 高精度移动目标位姿测量方法[J]. *高技术通讯*, 2019,29(2): 119-133

[3] 陈家颖,章伟杰,张强,等. 浅谈机器人在国内火电厂的应用研究[J]. *仪器仪表用户*, 2020,27(4):88-91

[4] 卢红生,刘斌,蒋峥,等. 基于 MMPC 的无人汽车轨迹跟踪控制[J]. *高技术通讯*, 2020,30(2):203-212

[5] Lin S, Goldenberg A A. Neural-network control of mobile

- manipulators [J]. *IEEE Transactions on Neural Networks*, 2001, 12(5) :1121-1133
- [6] 何文凯, 王江北, 陈萌, 等. 绳索牵引式并联机器人神经网络 PID 控制[J]. *高技术通讯*, 2018, 28(7) : 627-632
- [7] 刘峤, 吴怀宇, 陈洋, 等. 基于反步法与参考模型的四旋翼飞行器自适应滑动模态控制[J]. *高技术通讯*, 2018,28(7) :633-642
- [8] Fontes F A C C, Magni L. Min-max model predictive control of nonlinear systems using discontinuous feedbacks[J]. *IEEE Transactions on Control System Technology*, 2003, 48(10) : 1750-1755
- [9] Pei Y, Zhang K, Pan J, et al. Nonlinear model predictive tracking control of nonholonomic wheeled mobile robot using modified C/GMRES algorithm [C] // Proceedings of the 43rd Annual Conference of the IEEE Industrial Electronics Society (IECON), Beijing, China, 2017 : 6298-6303
- [10] Gu D B, Hu H S. Receding horizon tracking control of wheeled mobile robots [J]. *IEEE Transactions on Control Systems Technology*, 2006,14(4) :743-749
- [11] Guerreiro B J, Silvestre C, Cunha R, et al. Trajectory tracking nonlinear model predictive control for autonomous surface craft [J]. *IEEE Transactions on Control Systems Technology*, 2014,22(6) :2160-2175
- [12] Liu Y, Yu S, Guo Y, et al. Receding horizon control for path following problems of wheeled mobile robots [J]. *Control Theory and Applications*, 2017, 34(4) : 424-432
- [13] Büskens C, Maurer H. SQP-methods for solving optimal control problems with control and state constraints: adjoint variables, sensitivity analysis and real-time control [J]. *Journal of Computational and Applied Mathematics*, 2000, 120(1-2) : 85-108
- [14] Bock H G, Diehl M, Schlöder J P. Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations[J]. *Journal of Process Control*, 2002, 12(4) : 577-585
- [15] Findeisen R, Allgöwer F. Computational delay in nonlinear model predictive control [J]. *IFAC Proceedings Volumes*, 2004, 37(1) : 427-432
- [16] Bayer F, Hauser J. Trajectory optimization for vehicles in a constrained environment [C] // Proceedings of the 51st IEEE Conference on Decision and Control (CDC), Maui, USA, 2012 :5625-5630
- [17] Wang Q, Liu Y, Liu H, et al. Parallel numerical continuation of periodic responses of local nonlinear systems [J]. *Nonlinear Dynamics*, 2020, 100(3) :2005-2026
- [18] Li N, Zhang Y G, Wang C C. A convex combination LMS algorithm based on Krylov subspace transform [C] // Proceedings of the 5th International Joint Conference on Computational Sciences and Optimization, Harbin, China, 2012 :802-805
- [19] Tajeddin S, Vajedi M, Azad N L. A Newton/GMRES approach to predictive ecological adaptive cruise control of a plug-in hybrid electric vehicle in car-following scenarios [J]. *IFAC-Papers On-Line*, 2016, 49(21) :59-65
- [20] Ohtsuka T. A continuation/GMRES method for fast computation of nonlinear receding horizon control [J]. *Automatica*, 2004, 40(4) : 563-574
- [21] Domahidi A, Zraggen A U, Zeilinger N M, et al. Efficient interior point methods for multistage problems arising in receding horizon control [C] // Proceedings of the 51st IEEE Conference on Decision and Control (CDC), Maui, USA, 2012 : 668-674
- [22] Liao M D, Huang M, Kolmanovsky I. A regularized and smoothed fischer-burmeister method for quadratic programming with applications to model predictive control [J]. *IEEE Transactions on Automatic Control*, 2019, 64(7) :2937-2944
- [23] Schwickart T, Voos H, Darouach M. A fast model predictive speed controller for minimised charge consumption of electric vehicles [J]. *Asian Journal of Control*, 2016, 18(1) :133-149
- [24] Sajadi-Alamdari S A A, Voos H, Darouach M. Nonlinear model predictive control for ecological driver assistance systems in electric vehicles [J]. *Robotics and Autonomous Systems*, 2019, 112:291-303
- [25] Seguchi H, Ohtsuka T. Nonlinear receding horizon control of an underactuated hovercraft [J]. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, 2003, 13(34) : 381-398
- [26] Marutani J, Ohtsuka T. A real-time algorithm for nonlinear infinite horizon optimal control by time axis transformation method [J]. *International Journal of Robust and Nonlinear Control*, 2013, 23(17) : 1955-1971
- [27] Jiranek P, Rozložnik M, Gutknecht M H, et al. How to

- make simpler GMRES and GCR more stable[J]. *SIAM Journal on Matrix Analysis and Applications*, 2008, 30(4):1483-1499
- [28] 祖佳琪, 刘巧华. 简化块 GMRES 的稳定算法[J]. *应用数学与计算数学学报*, 2016, 30(1):51-59
- [29] Hauser J, Saccon A. A barrier function method for the optimization of trajectory functionals with constraints[C] // Proceedings of the 45th IEEE Conference on Decision and Control, San Diego, USA, 2006:864-869
- [30] Owaki Y, Yuno T, Kawabe T. Nonlinear model predictive control for path following of simple small electric vehicle using C/GMRES[J]. *IFAC-Papers OnLine*, 2018, 51(20): 253

Path tracking model predictive control of water wall moving robot using modified GMRES algorithm

Huang Wei, Zhan Hongwu, Xu Fang

(Key Laboratory of E&M (Zhejiang University of Technology), Ministry of Education
& Zhejiang Province, Hangzhou 310023)

Abstract

A fast nonlinear model predictive control (NMPC) algorithm is proposed for water wall climbing robot (WWCB) path tracking. The tracking error between WWCB and virtual target is defined, and the tracking problem is transformed into Bolza form of NMPC optimal control problem. The semi-smooth function is introduced to optimize the inequality, and the necessary conditions for optimization are constructed. Based on Pontriagin's minimum principle, the generalized Hamiltonian function is established, and continuous/generalized minimum residual algorithm (C/GMRES) is used to calculate the solution, as well as to improve the calculation accuracy, the multiple shooting method is introduced. At the same time, GMRES algorithm is optimized to improve the operating efficiency. The tracking simulation of Bernoulli double-wire and circular path is carried out to verify the effectiveness of the proposed tracking control algorithm.

Key words: water wall climbing robot (WWCB), nonlinear model predictive control (NMPC), semi smooth function, continuous/generalized minimum residual algorithm (C/GMRES), multiple shooting method, path tracking