DOI:10.11921/j.issn.2095-8382.20210412

机器人无迹微分动态规划算法研究

刘伟，马利强，马彪，陈雪辉，黄磊

（安徽建筑大学 机械与电气工程学院，安徽 合肥 230601）

摘 要：针对机器人非线性系统轨迹规划中微分动态规划算法由于动力学导数计算导致的实时性差与梯度下降慢问题，采用微分动态规划算法与无迹卡尔曼思想相结合的方式，以采样与差分方式代替动力学导数计算，建立无迹微分动态规划算法。将无迹微分动态规划与微分动态规划在非线性的倒立摆模型上进行模拟仿真对比。实验结果表明，系统参数相同时，无迹微分动态规划算法在保证良好的二阶收敛性和相同控制效果的前提下，既能减少迭代次数，又对成本压缩更敏感，且梯度下降更快，同时缩短算法整体的运行时间。

关键词：轨迹规划；无迹微分动态规划；仿真对比；二阶收敛；时间缩短

中图分类号：TP242.6 **文献标识码：**A **文章编号：**2095-8382(2021)04-071-06

Research on Unscented Kalman Dynamic Programming Algorithm of Robot

LIU Wei，MA Liqiang，MA Biao，CHEN Xuehui，HUANG Lei

（School of Mechanical and Electrical Engineering，Anhui Jianzhu University，Hefei 230601，China）

Abstract：The differential dynamic programming algorithm in the trajectory planning of the robot's nonlinear system due to the calculation of the dynamic derivative leads to poor real-time performance and slow gradient descent.Trackless differential dynamic programming algorithm adds Unscented Kalman thought to differential dynamic programming，using sampling and difference instead of dynamic derivative calculation.The traceless differential dynamic programming and differential dynamic programming are compared on the nonlinear inverted pendulum model.The experimental results show that when the system parameters are the same，the traceless differential dynamic programming algorithm can ensure good second-order convergence and control effect.The traceless differential dynamic programming algorithm can not only reduce the number of iterations and the overall running time of the algorithm，but also has a better sensitivity of cost compression and a faster gradient descent.

Key words：trajectory planning；trackless differential dynamic programming；simulation；second order convergence；reduce time

在机器人轨迹规划领域[1-2]，微分动态规划算法处理复杂高维非线性系统时有其独特优点[3-5]，但由于导数计算复杂，实时性差，在实际机器人控制中带来一定挑战。因此，研究一种无导数计算的微分动态规划算法，对于开发高性能的机器人控制系统的精准实时控制至关重要。

微分动态规划算法源于迭代线性二次调节控制（iLQR），而iLQR只使用动力学一阶导数计算[6]，虽然提高了运算速度，但是收敛速度不高[7]。例如iLQR与预测模型（MPC）应用于仿人机器人，由于实时性不强，算法常在收敛前终止[8]。由于高阶非线性系统结构复杂，任务需求大，需要快速计算，因而采用微分动态规划算法（DDP）[9]，对于光滑离散的系统具有良好的二阶收敛。但微分动态规划算法需计算反向二阶动力学导数[10]，而二阶导数计算比较复杂且较大的占用运算空间，算法运行耗时

收稿日期：2020-11-02

基金项目：安徽建筑大学引进人才及博士启动基金项目（2018QD16）；安徽省高校协同创新项目（GXXT-2019-036）。

作者简介：刘伟（1982-），男，副研究员，博士，主要从事机器人及智能装备、智能制造。

较长。因此对微分动态规划优化方向是如何处理海森矩阵。本文将微分动态规划算法与以无迹变换(UT)为基础的无迹卡尔曼[11](UKF)相结合为基础,以采样与差分方式代替动力学导数计算的思想,建立无迹微分动态规划算法。将无迹微分动态规划算法与微分动态规划通过非线性系统倒立摆仿真对比,验证了无迹微分动态规划的诸多优点,无需计算动力学导数。不仅保证了二阶收敛的特性与微分动态规划算法收敛轨迹相同,而且实时性优良,迭代次数减少,对于运行成本压缩更敏感,梯度搜索更快。该研究对机器人轨迹规划实时控制具有重要意义。

1 倒立摆模型

为了验证算法的可行性,本文采用文献[12–15]中的倒立摆模型。倒立摆是典型的多变量、高阶次、非线性、强耦合、自然不稳定的系统[16]。倒立摆系统的稳定控制是控制理论中的典型问题,在倒立摆控制的过程中能有效反映控制理论中许多关键的问题,例如非线性问题、鲁棒性问题、随动问题、跟踪问题等,因此,倒立摆系统常被用来检验新的控制算法的正确性及其在实际应用中的有效性[17]。

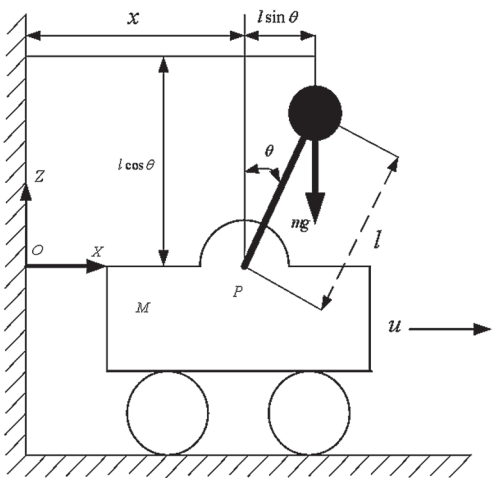


图 1 倒立摆模型

如图 1 的倒立摆小车的位移 x,摆杆与重力方向的夹角 θ,小车质量 M,摆球质量 m,摆杆转轴心到球心长度 l,输入控制量 U,摆球绕质心转动惯量 I。忽略阻力、摩擦与干扰,用拉格朗日(Lagrange)方程建立倒立摆模型的动力学方程[18]。

$$\begin{bmatrix} M+m & ml\cos\theta \\ ml\cos\theta & I+ml^2 \end{bmatrix}\begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix}+\begin{bmatrix} 0 & -ml\sin\theta \\ 0 & 0 \end{bmatrix}\begin{bmatrix} \dot{x} \\ \dot{\theta} \end{bmatrix}=\begin{bmatrix} U \\ mgl\sin\theta \end{bmatrix} \tag{1}$$

其中相关参数如表 1。

表 1 无迹微分动态规划伪代码

无迹微分动态规划
procedure UDP(x,u,ε) repeat K,l, δ V ← backward pass using (9)–(17) x,u, δ J ← FORWARDPASS($x,u,K,l,\delta V$) until $\lvert\delta J\rvert\leqslant\varepsilon$ return x,u end procedure

由(1)式求出关于$\ddot{\theta}$与$\ddot{x}$微分方程

$$\begin{cases} \ddot{\theta}=\dfrac{(M+m)mgl\sin\theta-ml\cos\theta\left(u+mgl\dot{\theta}\sin\theta\right)}{(M+m)\left(I+ml^2\right)-m^2l^2\cos^2\theta} \\ \ddot{x}=\dfrac{\left(I+ml^2\right)\left(u+mgl\dot{\theta}\sin\theta\right)-m^2l^2\cos^2\theta\sin\theta}{(M+m)\left(I+ml^2\right)-m^2l^2\cos^2\theta} \end{cases} \tag{2}$$

选择状态变量,由于动力学方程为非线性,使用时间步长为 h 的四阶龙格库塔法(Runge–Kutta)对动力学方程离散求解[19]。

2 无迹微分动态规划算法

系统的时间动力学方程通常采用:

$$x_{k+1}=f(x_k,u_k) \tag{3}$$

式中 $x\in R^n$ 是系统状态,$u\in R^m$ 是控制输入,定义一个反映控制过程中误差或损失大小的目标函数为

$$J(x_N,u_N)=\phi_f(x_N)+\sum_{k=1}^{N-1}L(x_k,u_k) \tag{4}$$

其中 J 表示的总成本是运行成本,φf(XN)表示最终成本的总和。等式后的第二项为中间过程代价和。我们的目的是寻找一组 $U^*=\{u_0,u_2,u_3,\cdots,u_{N-1}\}$ 使目标函数 J 最小。

对于非线性系统的目标函数,我们可以看做由二次与非二次形式组成

$$J=\frac{1}{2}x_N^TW_Nx_N+w_Nx_N+\sum_{k=1}^{N-1}\frac{1}{2}(x_k^TW_kx_k+R_ku_k)+w_kx_k+r_ku_k \tag{5}$$

其中 $W\in S_{++}^n$ 和 $R\in S_{++}^m$ 分别是正定和正定成本加权矩阵 $\omega\in R^n$ 和 $r\in R^m$ 成本权重向量。

V^* 表示最优运输总成本，Q^* 表示迭代过程的成本，这两个量将在时间步长 k 和 N 之间累计。如果遵循最优控制策略，使用 Bellman 的最优性原理，这个函数用递归关系表示：

$$J=\min Q_k^* \equiv \min(\sum_{k=1}^{N-1}\frac{1}{2}x_k^T W_k x_k + w_k x_k + \frac{1}{2}u_k^T R_k u_k + r_k u_k + V_{k+1}^*(f(x,u))) \tag{6}$$

当认为这是一个迭代更新过程时，这就是经典的微分动态规划 DDP 算法。由于复杂动力学系统为非线性的，当 $K=N$ 时，Vk^* 二次方程，但当 k 为其他值时为非二次形式。因此，在初始轨迹附近，V^* 要进行局部近似：

$$V_k(x+\delta x)\approx V_k(x)+\frac{1}{2}\delta x^T V_{xx|k}\delta x + V_{x|k}^T\delta x \tag{7}$$

其中 $V_{xx|k}$ 和 $V_{x|k}$ 分别是 V_k 的海森矩阵和梯度在 x 处取值。总时间成本 Q^* 近似为：

$$Q_k(x+\delta x,u+\delta u)\approx Q_k(x,u) + \frac{1}{2}\begin{bmatrix}\delta x\\ \delta u\end{bmatrix}^T\begin{bmatrix}Q_{xx} & Q_{ux}\\ Q_{xu} & Q_{uu}\end{bmatrix}\begin{bmatrix}\delta x\\ \delta u\end{bmatrix}+\begin{bmatrix}Q_x\\ Q_u\end{bmatrix}\begin{bmatrix}\delta x\\ \delta u\end{bmatrix} \tag{8}$$

其中 Q_{xx}、Q_{xu}、Q_{ux}、Q_{uu} 为 Q^* 的海森矩阵块，Q_x 与 Q_u 为梯度向量。若直接按上式（8）计算海森矩阵和梯度，即为经典算法微分动态规划。经典迭代算法 iLQR 算法，海森矩阵中出现的二阶导数计算量复杂，常常被忽略，导致系统收敛变差。

无迹微分动态规划为保持（8）式中的海森矩阵和梯度数据信息，采用扩展卡尔曼思想以采样差分的方式计算导数，具体处理过程如下（9）到（16）。

使用下面的 Cholesky 因子分解生成一组 2（$n+m$）个样本点

$$H=cholesky\left(\begin{bmatrix}V_{xx|k+1} & 0\\ 0 & R_k\end{bmatrix}^{-1}\right) \tag{9}$$

将样本点看成列向量 $H=[H_1,H_2,\cdots,H_{m+n}]$ 插入下列矩阵

$$\begin{cases}\begin{bmatrix}x_i^s\\ u_i^s\end{bmatrix}=\begin{bmatrix}x_{k+1}\\ u_k\end{bmatrix}+\sqrt{m+n+\tau}H_i\\ \begin{bmatrix}x_{i+n+m}^s\\ u_{i+n+m}^s\end{bmatrix}=\begin{bmatrix}x_{k+1}\\ u_k\end{bmatrix}-\sqrt{m+n+\tau}H_i\end{cases} \tag{10}$$

$$\lambda=\begin{cases}\frac{\tau}{m+n+\tau}+1-\varsigma^2+\beta & i=0\\ \frac{1}{2(m+n+\tau)} & i=1,\cdots,2(m+n)\end{cases} \tag{11}$$

$$\tau=\varsigma^2(m+n+\kappa)-(m+n) \tag{12}$$

其中 λ 为缩放因子，ς 通常取值范围为 10^{-4} 到 1，但常取极小的值，κ 大于等于 0，通常取 0。高斯分布时 β 取 2，非高斯情况取其他值。此时，其中 $V_{xx|k}$ 和 R_k 中的数据信息被添加到状态与控制相关的向量上，会通过动力学方程 $x_i^{s-}=f^-(x_i^s,u_i^s)$ 传递。样本点在反向传递的过程中，海森矩阵表示为

$$\begin{bmatrix}Q_{xx} & Q_{ux}\\ Q_{xu} & Q_{uu}\end{bmatrix}=\left(\frac{1}{2\lambda^2}\sum_{i=1}^{2^{m+n}}\left(\begin{bmatrix}x_i^{s-}\\ u_i^s\end{bmatrix}-\begin{bmatrix}x_k\\ u_k\end{bmatrix}\right)\left(\begin{bmatrix}x_i^{s-}\\ u_i^s\end{bmatrix}-\begin{bmatrix}x_k\\ u_k\end{bmatrix}\right)^T\right)^{-1}+\begin{bmatrix}W_k & 0\\ 0 & 0\end{bmatrix} \tag{13}$$

梯度向量 Q_x 与 Q_u 通过将 $V_{x|k}$ 投影到用于计算海森值的样本点上，可以计算出这一点。

$$V_i^s=V_{x|k+1}^T x_i^s \tag{14}$$

求解（m+n）（m+n）的线性方程组

$$\begin{bmatrix}x_1^{s-}-x_{n+m+1}^{s-} & \cdots & x_{n+m}^{s-}-x_{2(n+m)}^{s-}\\ u_1^s-u_{n+m+1}^s & \cdots & u_{n+m}^s-u_{2(n+m)}^s\end{bmatrix}\begin{bmatrix}f_x^T V_{x|k+1}\\ f_u^T V_{x|k+1}\end{bmatrix}=\begin{bmatrix}V_1^s-V_{n+m+1}^s\\ \vdots\\ V_{n+m}^s-V_{2(n+m)}^s\end{bmatrix} \tag{15}$$

由方程解出 $f_x^T V_{x|k+1}$ 与 $f_u^T V_{x|k+1}$ 的过程等价于计算中心差分的近似处理，将解出结果带入下式：

$$\begin{cases}Q_x=f_x^T V_{x|k+1}+W_k x+\omega_k\\ Q_u=f_u^T V_{x|k+1}+R_k x+r_k\end{cases} \tag{16}$$

关于 δu 的最小化方程得到控制轨迹的如下修正：

$$\delta u_k=-Q_{uu}^{-1}(Q_{xu}\delta x+Q_u)=-K_k\delta x-\alpha l_k \tag{17}$$

它由常数项 lk 和一个线性反馈项 $K\delta x$ 组成，将这些项代回各式得到式中的 Hk 和 gk：

$$\begin{cases}V_{xx|k}=W_k+K_k^T Q_{uu}K_k-K_k^T Q_{xu}-Q_{xu}^T K_k\\ V_{x|k}=w_k+W_k x+Q_x+(K_k^T Q_{uu}-Q_{xu}^T)l_k-K_k^T Q_u\end{cases} \tag{18}$$

当成本迭代运算到 k 时刻时，成本预期变化为：

$$\delta V_k = -l_k^T Q_{uu} l_k - Q_u^T l_k \tag{19}$$

从这里开始，执行向前递推过程，直至求出 u_1，结束前向递推。将新的状态控制 U 作为反馈修正带入动力学方程，可计算出新的状态 X。然后不断重复交替上述过程，直到满足终止条件[20]，则收敛到局部最优。

3 实验设计与结果分析

3.1 实验设计

用 MATLAB 书写倒立摆动力学模型与微分动态规划及无迹微分动态规划代码，将两种算法设置相同的参数，分别与倒立摆模型结合。由于动力学方程为非线性，使用步长 h 为 0.2 的四阶 Runge-Kutta 法离散。我们的目标是使钟摆从它的向下平衡摆动与重力方向夹角为 π，则目标状态为 $X_N=\begin{bmatrix} x & \pi & \dot{x} & \dot{\theta} \end{bmatrix}$，起始状态为 $X=[0\ 0\ 0\ 0]$，g 取 9.81 $m\cdot s^{-2}$，目标函数设置 $W_N=980I_{4x4}$，$w=0.5I_{4x4}$。根据仪器装置，仿真采用相关数据如表 2，实验过程如流程图 2。

表 2 相关参数

M/kg	m/kg	R	λ	N	l/m
15	1	0.01	1	50	1

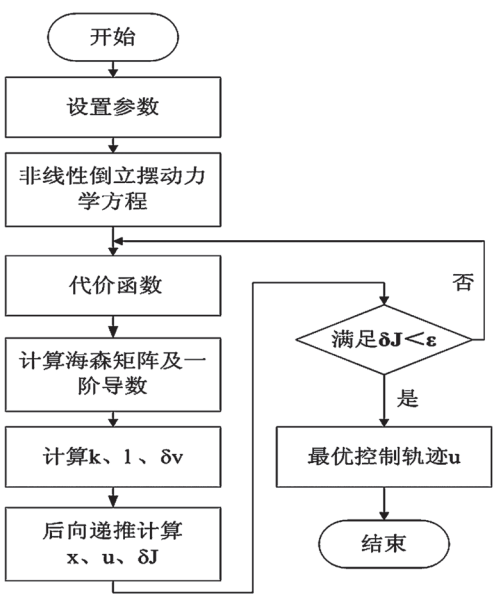


图 2 实验过程图

3.2 实验结果分析

实验采用的图表由程序运行直接生成。

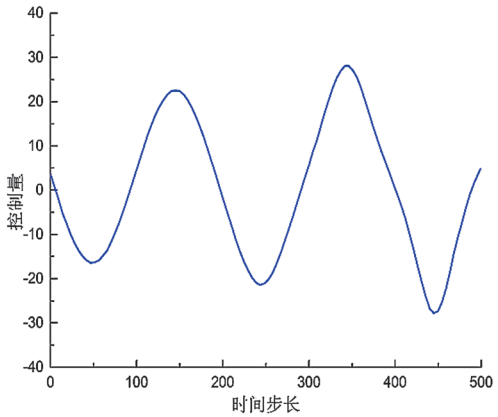


图 3 微分动态规划控制量变化

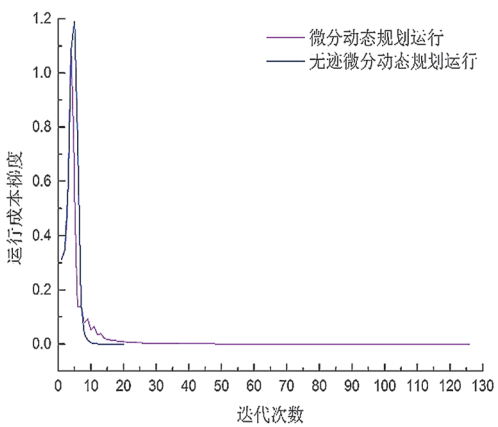
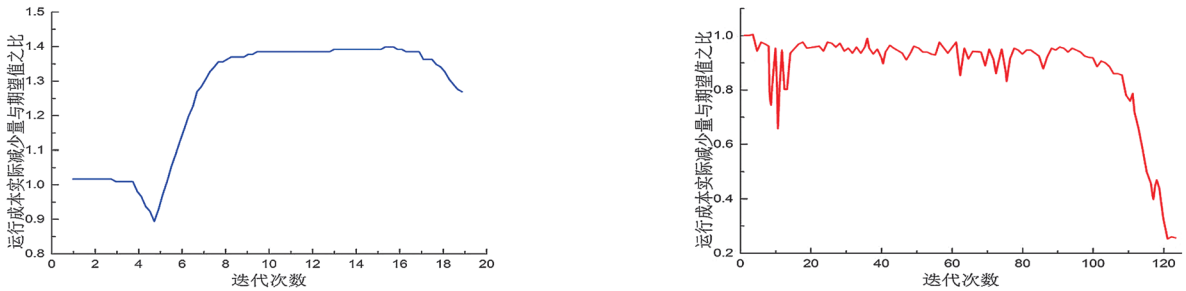


图 4 两种算法梯度下降对比图

两种算法的控制量得出如图 3 相同的结果，控制量在时间步长 [0,500] 上不断变化，控制量 u 始终处于 [-30,30] 范围内，从波峰到波谷不断震荡。在相同的时间步长上，无迹微分动态规划与微分动态规划控制量大小相同，因此，两种算法计算出的控制量 U 的收敛轨迹相同，两者达到轨迹相同的控制效果。

图 5 中微分动态规划算法的实际运行成本与期望值之比在迭代步长上取值在 0.2 到 1 之间；图 5(a) 中无迹微分动态规划算法的实际运行成本与期望值之比均值大于 1，所以无迹微分动态规划对成本更敏感，更易于向最优方向搜索控制量 U。

图 6 中每次迭代的运行成本 V 分析，条件与参数相同时，起点总成本相同，与最后收敛到相同成本时，无迹微分动态规划前 5 次迭代中成本迅速下降，5 到 20 迭代过程成本变化量较少，稳定



（a）无迹微分动态规划　　（b）微分动态规划

图 5　实际减少与期望值比率

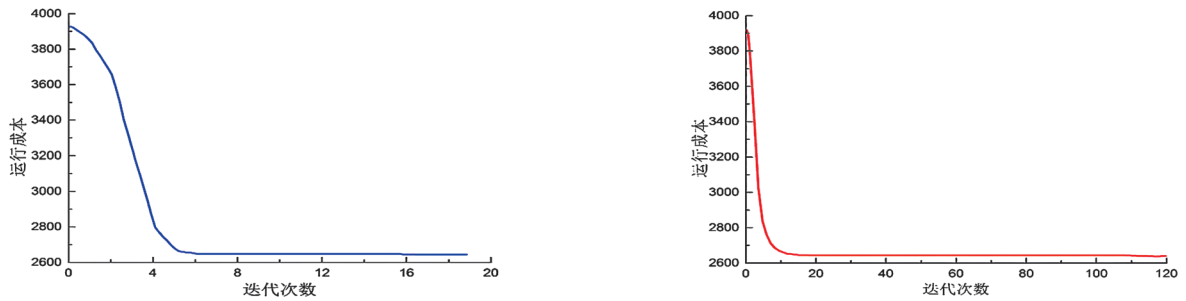


图 6　运行成本变化

2 640。而微分动态规划前 10 迭代成本迅速下降，10 到 120 之间迭代过程，成本减少量不大，与无迹微分动态规划收敛的成本相同 2 640。因此无迹微分动态规划对成本压缩更快，对成本更敏感。

图 4 微分动态规划算法的运行成本梯度从一个小于 1.1 大于 1 的值在迭代次数 [0，20] 的区间内迅速递减，之后趋于平稳。而无迹微分动态规划在迭代步长 [0，10] 上先递增后递减且最大梯度值大于微分动态规划。最优值与梯度方向密切相关，无迹微分动态规划先升后减的过程更易搜索局部最优，因此，无迹微分动态规划能以较小的迭代次数收敛到与微分动态规划相同的最优控制过程。

在算法中写入记录迭代次数，算法运行最终价值及记计时等相关命令。算法仿真结束后，从 MATLAB 命令窗口得到表 3 数据：

表 3　微分动态规划结果相关数据

无迹微分动态规划
SUCCESS：cost change < tolFun
iterations：20
final cost：2644.956
final grad：2.367922e–07
time / iter：413 ms
total time：8.25 seconds，of which
derivs：0.1%
back pass：59.1%
fwd pass：5.2%
other：35.6%（graphics etc.）
========== end==========

微分动态规划和无迹微分动态规划算法运行结束，总成本与梯度都分别收敛于 2644.956，2.5e–07，无迹动态微分规划迭代次数 20，每次迭代时间 413 ms，而微分动态规划以每次 1 650 ms 的速度，迭代 126 次。无迹微分动态规划不仅迭代次数少，而且总的运行时间 8.25 秒远小于微分动态规划总耗时 207.86 秒。

针对非线性系统轨迹优化中，微分动态规划算法每次迭代过程的状态动作值函数导数计算比较复杂，算法迭代次数多，且运行时间慢，不利于实时控制等问题，运用改进的微分动态规划进行对比仿真分析，仿真结果验证了无迹微分动态规划算法弥补了微分动态规划的缺陷，如表 4 所示。

表 4　无迹微分动态规划结果相关数据

微分动态规划
SUCCESS：cost change < tolFun
iterations：126
final cost：2 644.956
final grad：2.907 287e–07
time / iter：1 650 ms
total time：207.86 seconds，of which
derivs：88.0%
back pass：2.2%
fwd pass：1.3%
other：8.5%（graphics etc.）
========== end==========

无迹微分动态规划保持着微分动态规划算法的优点，如各种条件相同时，在相同的时间步长上

两种算法收敛的控制轨迹相同，且保证了二阶收敛的特性。

无迹微分动态规划每次迭代搜索无需直接计算状态动作值函数中海森矩阵。在对比实验中，无迹微分动态规划相比于微分动态规划，迭代搜索次数少，梯度下降快，对于成本压缩表现敏感且运行总时间更少。

4 结论

（1）本文针对微分动态规划算法轨迹规划因导数计算引起实时性不高与梯度下降慢问题，结合无迹卡尔曼思想，用采样差分的方式替代导数计算，建立无迹微分动态规划算法。

（2）用 Lagrange 方程建立非线性倒立摆模型，基于控制变量法的思想，将两种算法设置相同参数，在此模型上进行实验对比。

（3）实验结果表明，无迹微分动态规划算法不但能用于非线性系统和保持二阶收敛的特性，而且还减少了算法整体运行的时间。最后，由于该算法蕴含无迹卡尔曼思想，因此在传播样本价值的基础上，减少迭代次数，对于运行成本压缩更敏感，梯度搜索更快。

参考文献：

[1] Luis C E，Vukosavljev M，Schoellig A P.Online trajectory generation with distributed model predictive control for multi-robot motion planning[J].IEEE Robotics and Automation Letters，2020，5（2）：604-611.

[2] 李东洁，邱江艳，尤波．一种机器人轨迹规划的优化算法 [J]. 电机与控制学报，2009，13（1）：123-127.

[3] 孙茂相．最优控制计算中的微分动态规划（DDP）方法 [J]. 信息与控制，1985，14（2）：40-43.

[4] 郭行，符文星，付斌，等．复杂动态环境下无人飞行器动态避障近似最优轨迹规划 [J]. 宇航学报，2019，40（2）：182-190.

[5] Brockett R.The early days of geometric nonlinear control[J]. Automatica，2014，50（9）：2203-2224.

[6] Faisal M，Jamil M，Awais Q，et al.Iterative linear quadratic regulator （ILQR） controller for trolley position control of quanser 3DOF crane[J].Indian Journal of Science and Technology，2015，8（16）：1-7.

[7] Ching S，Kabamba P T，Meerkov S M.Simultaneous design of controllers and instrumentation： ILQR/ILQG[J].IEEE Transactions on Automatic Control，2010，55（1）：217-221.

[8] Liang C，Li Y，Luo J W.A novel method to detect functional microRNA regulatory modules by bicliques merging[J]. IEEE/ACM Transactions on Computational Biology and Bioinformatics，2016，13（3）：549-556.

[9] Tassa Y，Erez T，Todorov E.Synthesis and stabilization of complex behaviors through online trajectory optimization[J].2012 IEEE/RSJ International Conference on Intelligent Robots and Systems，2012：4906-4913.

[10] Sun W，Pan Y P，Lim J，et al.Min-max differential dynamic programming： continuous and discrete time formulations[J].Journal of Guidance，Control，and Dynamics，2018，41（12）：2568-2580.

[11] Chang L B，Hu B Q，Li A，et al.Transformed unscented Kalman filter[J].IEEE Transactions on Automatic Control，2013，58（1）：252-257.

[12] Kajita S ，Tan K .Study of Dynamic Walk Control of a Biped Robot on Rugged Terrain - Derivation and Application of the Linear Inverted Pendulum Mode -[J].Transactions of the Society of Instrument and Control Engineers，1991，27（2）：177-184.

[13] Chang L，Piao S H，Leng X K，et al.Inverted pendulum model for turn-planning for biped robot[J].Physical Communication，2020，42：101168.

[14] 徐明，周滨，胡国良．直线一级倒立摆系统控制仿真及实验研究 [J]. 机床与液压，2018，46（6）：90-95.

[15] 唐火红，丁婧，严启凡．液压驱动双足机器人步态规划及动力学仿真 [J]. 机械设计与制造，2020（4）：248-252，257.

[16] Cao X，Li N F，Zhang H X.Robust controller design for inverted pendulum system[J].Advanced Materials Research，2013，631/632： 1342-1347.

[17] Gîlcă G，Ionescu M，Bîzdoacă N G，et al.The control of an inverted pendulum system based on simple and adaptive fuzzy regulators[J].IOP Conference Series： Materials Science and Engineering，2018，444： 042019.

[18] Fan J，Wang X H，Fei M R.Research of parallel-type double inverted pendulum model based on Lagrange equation and LQR controller[C]// Life System Modeling and Intelligent Computing，2010，6328：147-156.

[19] Grzelczyk D，Sta ń czyk B，Awrejcewicz J.Prototype，control system architecture and controlling of the hexapod legs with nonlinear stick-slip vibrations[J].Mechatronics，2016，37：63-78.

[20] Andrei N.Another conjugate gradient algorithm with guaranteed descent and conjugacy conditions for large-scale unconstrained optimization[J].Journal of Optimization Theory and Applications，2013，159（1）：159-182.