

DOI: 10.13973/j.cnki.robot.2017.0355

自由落猫机器人时间最优下落轨迹研究

梁兴灿^{1,2}, 徐林森², 李露², 周波^{1,2}, 赵家轩^{1,2}

(1. 中国科学技术大学, 安徽 合肥 230026; 2. 中国科学院合肥物质科学研究院先进制造技术研究所, 江苏 常州 213164)

摘要: 机器人在未知复杂环境下作业时, 有从高处跌落的风险, 所以需要考虑其空中的姿态调整能力, 减轻由错误的着陆方式造成的伤害. 当猫从高空坠落时, 它总能通过自身调整安全落地. 受这一生物学现象启发, 通过研究猫能够安全落地的第一阶段, 姿态调整阶段, 首次提出以时间最优方式研究落猫机器人最优下落轨迹问题. 首先利用轴对称双刚体模型导出机器人的数学模型, 因为角速度不可积, 将落猫机器人的姿态控制问题转变成非完整系统的运动规划问题. 考虑到姿态调整所消耗的时间是决定调整结果较为重要的因素, 随后以虚拟力矩输入取代真实角速度输入, 建立时间优化函数, 并构造了求解该函数的方法. 然后以姿态调整耗时最短为控制目标, 通过粒子群优化算法寻找目标最优解. 最后运用所得最优解数据, 在虚拟物理环境下进行了虚拟样机实验, 并实现了落猫机器人的空中翻正动作. 结果表明, 该方法能有效地缩短自由下落机器人的姿态调整时间.

关键词: 自由落猫机器人; 姿态调整; 轨迹规划; 时间最优; 粒子群优化算法

中图分类号: TP242

文献标识码: A

文章编号: 1002-0446(2017)-03-0355-08

Research on Time-Optimal Falling Trajectory of a Free-Falling Cat Robot

LIANG Xingcan^{1,2}, XU Linsen², LI Lu², ZHOU Bo^{1,2}, ZHAO Jiakuan^{1,2}

(1. University of Science and Technology of China, Hefei 230026, China;

2. Institute of Advanced Manufacturing Technology, Hefei Institutes of Physical Science, Chinese Academy of Sciences, Changzhou 213164, China)

Abstract: A robot may be at the risk of falling from a high place when it works in an unknown complex environment, so the attitude control ability of the robot in the air should be considered to reduce the damage caused by wrong landing attitudes. When a cat drops from a high space, it can always right itself and land safely. Inspired by this biological phenomenon, the optimal falling trajectory of a free-falling cat robot is studied in a time-optimal manner for the first time, through investigating the first stage of safe landing of a cat, namely the attitude adjustment stage. Firstly, a mathematical model of the robot is formulated based on an axisymmetric dual rigid-body model. Owing to the non-integrable angular velocity, the attitude control problem of the falling cat robot is transformed into a nonholonomic motion planning problem. Considering that the time consumption of attitude adjustment is an important factor to determine the adjustment result, a time-optimal function is built with the virtual torque input instead of the real angular velocity input, and a method to solve this function is set up also. Then, a particle swarm optimization algorithm is proposed to obtain the solution of objective function with the least time consumption of attitude adjustment. Finally, using the optimal solution data, the virtual prototype experiment is carried out in the virtual physical environment, and the righting movement in the air for a falling cat robot is realized. The results show that the time consumption of attitude adjustment is effectively reduced for the free-falling robot by the proposed method.

Keywords: free-falling cat robot; attitude adjustment; trajectory planning; time-optimal; particle swarm optimization algorithm

1 引言 (Introduction)

当代机器人的研究与应用已从结构环境下的定点作业拓展到星际探测、地质勘探、灾难救援、反恐防暴等非结构环境下的自主作业领域. 在非结构环境中, 腿式机器人相对于轮式和履带式机器人更具优势, 但运动环境对腿式机器人仍有诸多限制^[1-3]. 为了进一步扩大腿式机器人应用场合, 如

外星球探测器着陆、深井救援机器人着陆以及高空救援机器人着陆等, 需要解决腿式机器人从空中无意跌落或被空投时的安全着陆问题. 因此, 研究机器人依靠自身的机械、控制系统实现任意姿态跌落的安全着陆问题至关重要, 可以为机器人代替人类进入森林火场、高山、高塔等危险环境工作提供保障^[4-5].

为了说明自由落猫机器人, 首先从落猫谈起. 众所周知, 当一只猫从高空坠落时, 它总能通过自身的调整, 安全地四脚着地. 在不违背角动量守恒的情况下, 猫是怎样在空中实现姿态的调整呢? 为揭示猫自动翻正的力学原理, 历史上很多科学家提出了各种假设^[6], 有四肢开合论、转尾巴论、绕双轴转动论、弯腰论等. 其中被广泛认可的是由 Kane 和 Scher 提出的弯腰论^[7], 他们将猫简化为一个以腰部为顶点的 2 个轴对称刚体组成的双刚体模型, 分别代表猫的前半身和后半身, 并且认为猫的脊柱仅可弯曲, 但前后体之间没有扭转, 通过力学分析和计算机模拟, 解释了猫的转体运动. 基于 Kane-Scher 模型, 很多学者研究了自由落猫的最优下落轨迹^[6,8-13]. Montgomery^[14-15] 通过几何微分学证明了 Kane-Scher 模型的正确性. 他认为在猫下落过程中, 当系统总角动量为 0 时, 落猫轨迹规划问题等价于非完整运动规划问题, 因此, 落猫问题成为一个典型的非完整运动问题. 落猫系统的非完整性是由不可积的角速度引起的^[16], 其特点是系统广义坐标的维数多于控制输入的维数, 非完整系统是一种特殊的非线性系统. 由于系统的自由度多于独立控制变量, 所以, 即使知道系统的动力学方程, 也没有唯一的方法处理非完整系统, 因此, 落猫控制问题变得非常困难.

尽管文 [6,8-13] 都能得到较好的下落轨迹, 但优化函数都是以能量最优的方式进行, 并未对时间最优进行探讨. 文 [10] 通过设计量子神经网络控制器, 把翻正时间缩短到 0.35 s, 但是控制器的设计过于复杂, 较难实现. 考虑到自由落体过程中, 下落高度固定, 提供给姿态调整的时间有限, 因此研究如何利用最少的时间来实现翻正是非常有意义的, 另外, 很少有人把对落猫的相关研究应用到机器人上. 本文旨在通过研究落猫的最优时间下落轨迹, 利用轨迹规划, 把落猫的这一特点应用到机器人上, 实现当机器人无意跌落或被空投时可以通过自身的控制系统和机械结构安全着陆.

本文采用 Kane-Scher 模型建立落猫机器人的动力学方程, 在系统角动量为 0 的情况下, 基于该模型将系统的姿态控制问题转化为非完整系统的运动规划问题. 在非完整运动规划中, 建立自由下落过程中姿态调整所耗时间的函数, 将该函数作为最优控制目标, 通过罚函数方法, 把求解具有约束方程的最优化问题转变成无约束优化问题, 最后通过粒子群优化 (PSO) 算法寻找最优解, 得到了落猫机器人在自由下落时从初始位姿到期望位姿的优化运

动轨迹, 结果表明, 采用本方法进行姿态调整的时间消耗少于以其他方式进行调整姿态的时间消耗.

2 落猫机器人的动力学模型 (Dynamical model of a falling cat robot)

在猫的下落过程中, 脊柱依次向各个方向弯曲, 猫的前半身做一轮弯腰运动, 整个身子就向相反方向转 180°, 达到翻正目的^[7]. 仿照自然界中落猫翻正的生物学原理, 本文以修正的 Kane-Scher 模型^[6] 作为原型进行研究, 如图 1 所示. 落猫机器人的简化模型是由 2 个轴对称的刚体组成, 分别代表机器人的前半身 B_1 和后半身 B_2 , 并以铰连接在腰部 O 点处. 为了更好地理解整个模型, 定义如下符号:

O	机器人前后体的连接处
O_c	整个机器人的质心
OX_1	从 O 指向头部的中心轴
OX_2	从尾部指向 O 点的中心轴
e_1, e_2, e_3	3 维坐标轴的标准正交基
χ_1, χ_2	前后体分别绕 OX_1 和 OX_2 转动角度
θ_1, θ_2	B_1 和 B_2 的弯曲角
ω	整个机器人模型绕水平轴转动的角度

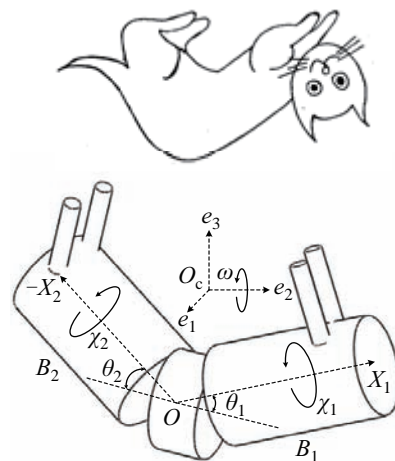


图 1 落猫机器人模型

Fig.1 A falling cat robot model

需指出, 图 1 中 B_1 和 B_2 上的 4 条腿仅仅是为了方便区分背部和腹部, 没有实际意义, 整个翻正过程中保持固定, 对实验结果没有任何影响; 另外, 中间的那部分代表前后体之间以铰连接在腰部.

由于模型的对称特性, 可以定义 $\chi_1 = \chi_2 = \chi$, $\theta_1 = \theta_2 = \theta$, 因此, 在以后的讨论中, 只需关心其中 3 个参数 ω 、 χ 、 θ 的变化. 从文 [6] 中, 可以得

到这 3 个变量的关系如下:

$$\dot{\omega} = f_1 \dot{\chi} + f_2 \dot{\theta} \quad (1)$$

其中

$$f_1 = \frac{\cos \theta \sin^2 \theta [\mu + (1 - \varepsilon) \cos^2 \chi]}{(1 - \sin^2 \theta \cos^2 \chi)[1 + (\mu - \varepsilon \cos^2 \chi) \sin^2 \theta]}$$

$$f_2 = \frac{\cos \chi \sin \chi \sin \theta (1 - \varepsilon + \mu \sin^2 \theta)}{(1 - \sin^2 \theta \cos^2 \chi)[1 + (\mu - \varepsilon \cos^2 \chi) \sin^2 \theta]}$$

这里, μ 和 ε 是与机器人质量和结构相关的参数. 定义状态变量 $\mathbf{x} = [\chi, \theta, \omega]^T \in \mathbb{R}^3$, 控制输入 $\mathbf{u} = [\dot{\chi}, \dot{\theta}]^T \in \mathbb{R}^2$, 需要指出 $u_1 = \dot{\chi}$, $u_2 = \dot{\theta}$, 此时, 式 (1) 可以重新写成如下形式:

$$\dot{\mathbf{x}} = \mathbf{A}(\mathbf{x})\mathbf{u} \quad (2)$$

$$\mathbf{A}(\mathbf{x}) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ f_1 & f_2 \end{bmatrix} \in \mathbb{R}^{3 \times 2}$$

非完整约束是指系统广义速度约束不能通过积分转变为系统广义位置约束, 受非完整约束支配的系统称之为非完整系统, 其特点是系统广义坐标的维数大于系统控制输入的维数^[17-19]. 注意方程 (2) 中系统控制输入维数 $m = 2$ 比状态变量维数 $n = 3$ 小, 落猫机器人自由下落时的转体运动可以用 2 个控制输入和 3 个状态变量的非线性控制系统描述, 且落猫系统的角速度不可积. 因此, 落猫系统是一个非完整系统, 其轨迹规划问题可以等价为非完整运动规划问题, 即给定初始位姿 \mathbf{x}_0 , 期望位姿 $\mathbf{x}_f \in \mathbb{R}^3$, 找一组控制输入 $\mathbf{u}(t) \in \mathbb{R}^2, t \in [0, T]$, 使系统 (2) 在时间 T 内从 \mathbf{x}_0 到达 \mathbf{x}_f , 具体证明参考文献 [14,16,19]. 从方程形式可以看出, 这是一个高度非线性和强耦合的非线性系统, 单独求解相当困难, 因此转变成目标函数的最优化求解.

3 建立时间目标优化函数 (Setting up objective optimal function based on time)

3.1 建立时间函数

不考虑空气阻力, 在仅受重力的情况下, 当机器人自由下落时, 空中姿态调整的时间有限, 必须要在落地之前完成翻正动作, 否则没有意义. 因此, 相对于能量最优化方法来讲, 缩短姿态调整的时间要比减少能量消耗更加重要. 所以, 本文建立机器人完成翻正所需时间的函数, 并以此作为优化的目标函数, 即:

$$Y(T) = T^2 \quad (3)$$

其中, T 表示姿态调整所耗时间.

由式 (2) 可知, 系统以角速度作为控制输入. 为了求出控制输入 \mathbf{u} 和完成翻正所需时间 T 的关系, 假设控制输入为力矩 \mathbf{M} , 我们说由于外加力矩 \mathbf{M} 的作用, 导致系统有了角速度 \mathbf{u} , 进而完成翻正动作, 整个过程消耗时间 T .

考虑如下方程:

$$\mathbf{M} = I\boldsymbol{\beta}$$

$$\boldsymbol{\rho} = \boldsymbol{\rho}_0 + \boldsymbol{\beta}t \quad (4)$$

$$\boldsymbol{\varphi} = \boldsymbol{\rho}_0 + \frac{1}{2}\boldsymbol{\beta}t^2$$

这里, \mathbf{M} 是常力矩输入, I 是系统转动惯量, $\boldsymbol{\beta}$ 是常角加速度, $\boldsymbol{\rho}_0$ 是初始角速度, $\boldsymbol{\rho}$ 是角速度, t 是力矩施加时间, $\boldsymbol{\varphi}$ 是转动角度.

在初始位姿 \mathbf{x}_0 和末端位姿 \mathbf{x}_f 已知, $\boldsymbol{\rho}_0 = 0$ 的前提下, 可知时间 T 是力矩 \mathbf{M} 的函数:

$$T = \text{sum}(\mathbf{t}) = \Gamma(\mathbf{M})$$

$\text{sum}(\mathbf{t})$ 表示 \mathbf{t} 的所有元素之和.

引入罚函数方法, 方程 (3) 可近似表示为

$$Y(\mathbf{M}, \tau) = \Theta(\mathbf{M}) + \tau \|\mathbf{x}(T) - \mathbf{x}_f\|^2$$

其中, τ 是罚因子, $\mathbf{x}(T) \in \mathbb{R}^3$ 为方程 (2) 由控制输入 \mathbf{u} 给定在 $t = T$ 时的解, 显然, $\mathbf{x}(T)$ 是 \mathbf{M} 的函数, 记 $\mathbf{f}(\mathbf{M}) = \mathbf{x}(T)$, 当 τ 已知时, 可以重写目标函数为

$$Y(\mathbf{M}) = \Theta(\mathbf{M}) + \tau \|\mathbf{f}(\mathbf{M}) - \mathbf{x}_f\|^2 \quad (5)$$

当 $\tau \rightarrow \infty$ 时, 通过方程 (5) 得到的优化解近似满足具有约束方程的式 (3). 求得 \mathbf{M} 之后, 可以进一步求得时间 T , 现在问题转化为寻找一组合适的参数 \mathbf{M} 使得目标函数值最小. 由于方程 (5) 中包含状态方程信息, 所以该方程也具有高度非线性且不一定存在连续的 1 阶、2 阶导数, 不宜采用传统的梯度下降法求解. 考虑到 PSO 算法是求解各种非线性优化问题的有效优化工具, 且该算法不需要任何梯度信息, 具有算法简单、易于实现、需要调节的参数较少等优点^[20-22], 因此本文采用 PSO 算法实现式 (5) 的最优化求解.

3.2 求解 $\mathbf{x}(T)$

需要指出, 在求解 $Y(\mathbf{M})$ 时, 难点在于求解 $\mathbf{x}(T)$. 这是因为必须知道初始状态 \mathbf{x}_0 和翻正时间 T , 才能求得 $\mathbf{x}(T)$, 而 T 是最优化求解过程中未知的量. 为了完整地描述本文提出的时间优化函数, 拟构造如下方法求解 $\mathbf{x}(T)$.

假设需要转动的角度为 φ . 把 φ 等分成 N 份, 则 $\boldsymbol{\varphi} = [\varphi_1, \varphi_2, \dots, \varphi_N]$, 其中, $\varphi_i = \varphi/N, i = 1, 2, \dots, N$. 记 $\boldsymbol{M} = [M_1, M_2, \dots, M_N]$, 其中 M_i 是转动 φ_i 时施加的力矩, 其他变量的标记类似. 由极限理论可知, 当 $N \rightarrow \infty$ 时, 转过 φ_i 时的角速度 ρ_i 、角加速度 β_i 和力矩 M_i 都可以看作常数, 满足方程 (4). 求解 $\boldsymbol{x}(T)$ 的具体步骤如下:

- Step1. 给 \boldsymbol{M} 赋值, 预置初始角速度 $\boldsymbol{\rho}_0 = 0$.
- Step2. 由 φ_1 、 $\boldsymbol{\rho}_0$ 、 $\beta_1 = M_1/I$, 求得 t_1 和 $\boldsymbol{\rho}_1$.
- Step3. 由 φ_2 、 $\boldsymbol{\rho}_1$ 、 $\beta_2 = M_2/I$, 求得 t_2 和 $\boldsymbol{\rho}_2$.
- Step4. 重复 Step3 $N-2$ 次, 求得力矩持续时间 $\boldsymbol{t} = [t_1, t_2, \dots, t_N]$, 对应点角速度 $\boldsymbol{\rho} = [\rho_1, \rho_2, \dots, \rho_N]$.
- Step5. 由 \boldsymbol{t} 求得时间节点 $\boldsymbol{T} = [T_1, T_2, \dots, T_N]$.
- Step6. 对 $[\boldsymbol{T}, \boldsymbol{\rho}]$ 进行曲线拟合, 得到 $[0, T_N]$ 上的控制输入 $u_j, j = 1, 2$.
- Step7. 取对应于 u_1 和 u_2 较大的时间点 T_N 赋值给 T .
- Step8. 运用求得的 \boldsymbol{u} 、 T 、初始状态 \boldsymbol{x}_0 , 通过 Matlab 中 ode45 函数便可求出 $\boldsymbol{x}(T)$.

4 粒子群优化算法 (Particle swarm optimization algorithm)

4.1 粒子群优化算法理论基础

PSO 算法是一种群体智能优化算法, 最早由 Kennedy 等提出^[20], PSO 算法源于对鸟类捕食行为的研究, 在 PSO 算法中, 每个待优化问题的解都是搜索空间中被抽象为没有质量和体积的一只鸟, 所有的粒子都对应一个适应值, 每个粒子还有一个速度决定它们飞翔的方向和距离.

PSO 算法首先在可行解空间中初始化一群粒子, 然后粒子们就追随当前的最优粒子在解空间中搜索. 粒子的运动特征用位置、速度和适应度值 3 项指标表示, 其中最重要的是适应度值, 由适应度函数计算得到, 其值的好坏代表粒子的优劣, 每个粒子都代表优化问题的一个潜在最优解. 粒子在解空间运动时, 每次通过追踪个体极值 (个体经历的最好位置) \boldsymbol{P}_i 和群体极值 (所有粒子经历的最好位置) \boldsymbol{P}_g 更新个体位置. 假设 D 维搜索空间中, 由 K 个粒子组成的种群, 第 i 个粒子的位置表示为 $\boldsymbol{X}_i = [X_{i1}, X_{i2}, \dots, X_{iD}]^T$, 速度 $\boldsymbol{V}_i = [V_{i1}, V_{i2}, \dots, V_{iD}]^T$, 个体极值 $\boldsymbol{P}_i = [P_{i1}, P_{i2}, \dots, P_{iD}]^T$, 群体极值 $\boldsymbol{P}_g = [P_{g1}, P_{g2}, \dots, P_{gD}]^T$. 在每次迭代运

算中, 粒子通过 \boldsymbol{P}_i 和 \boldsymbol{P}_g 来更新自己的速度和位置^[21], 即

$$\begin{aligned} \boldsymbol{V}_i &= \zeta \boldsymbol{V}_i + c_1 r_1 (\boldsymbol{P}_i - \boldsymbol{X}_i) + c_2 r_2 (\boldsymbol{P}_g - \boldsymbol{X}_i) \\ \boldsymbol{X}_i &= \boldsymbol{X}_i + \boldsymbol{V}_i \end{aligned} \quad (6)$$

其中, ζ 是惯性权重, c_1 和 c_2 为正的学习因子, r_1 和 r_2 为 0 到 1 之间均匀分布的随机数.

在 PSO 中, 惯性权重 ζ 是特别重要的参数. 为了便于加速收敛以及较易跳出局部极小值, 采用随机权重法^[22] 求解 ζ :

$$\begin{aligned} \boldsymbol{\omega} &= \boldsymbol{\omega}_{\min} + (\boldsymbol{\omega}_{\max} - \boldsymbol{\omega}_{\min}) \times r_3 \\ \zeta &= \boldsymbol{\omega} + \boldsymbol{\sigma} \times N(0, 1) \end{aligned} \quad (7)$$

其中 $N(0, 1)$ 表示标准正态分布的随机数, r_3 表示 0 到 1 之间的随机数, $\boldsymbol{\omega}$ 表示惯性权重参数.

4.2 最优化求解过程

通过 Matlab 软件编写程序, 运行, 便可得到最优时间 T , 具体运行步骤如下:

- Step 0:**
- ① 设置初始位姿和期望位姿 $\boldsymbol{x}_0, \boldsymbol{x}_t \in \mathbb{R}^3$.
 - ② 选择合适的 c_1, c_2, τ, K , 位置和速度向量维数 D , 最大迭代次数 C_1 , 随机初始化种群中各微粒的位置和速度.
 - ③ 通过式 (5) 计算各微粒适应值, 将对应位置存入 $\boldsymbol{P}_i (i = 1, 2, \dots, K)$, 将所有 \boldsymbol{P}_i 中最小的一个存入 \boldsymbol{P}_g .
- Step 1:**
- ① 用式 (7) 更新 ζ , 用式 (6) 更新粒子的速度和位置.
 - ② 对于每个微粒, 将其适应值与其经历过最好位置的适应值作比较. 如果较小, 则将其作为当前最好位置.
 - ③ 比较当前 \boldsymbol{P}_i 和 \boldsymbol{P}_g 的值, 选择较小的为新 \boldsymbol{P}_g .
 - ④ 满足最大迭代次数时, 停止搜索; 否则, 重复 Step 1.

5 仿真结果 (Simulation result)

机器人要实现从四脚朝天到安全着陆, 在自由下落的过程中, 其前、后体要能够分别以腰部为顶点做圆锥运动, 且整体往相反方向旋转 180° , 即当 χ 从 0 变化到 2π 时, ω 从 0 变到 π .

5.1 PSO 算法的参数设定

在仿真实验中, 取 $I = 0.0256$, $\mu = 3$, $\varepsilon = 0.01$ ^[7,9], 根据经验和尝试, PSO 算法中较好的参数组合如表 1 所示.

表 1 参数初始化设置
Tab.1 Initialization of parameters

参数	取值
采样时间	$\Delta T = 0.005 \text{ s}$
角度等分份数	$N = 360$
罚因子	$\tau = 3000$
学习因子	$c_1 = c_2 = 2$
惯性权重参数	$\omega_{\max} = 1.2, \omega_{\min} = 0.4, \sigma = 0.2$
种群数量	$K = 40$
维数	$D = 720$
最大迭代次数	$C_1 = 500$
初始位姿	$\mathbf{x}_0 = [0, \pi/6, 0]^T$
期望位姿	$\mathbf{x}_r = [2\pi, -\pi/6, \pi]^T$

5.2 Matlab 仿真结果分析

以下从 3 个示例来说明 Matlab 仿真结果, 每个示例仅仅是运行结果的不同, 各参数保持不变.

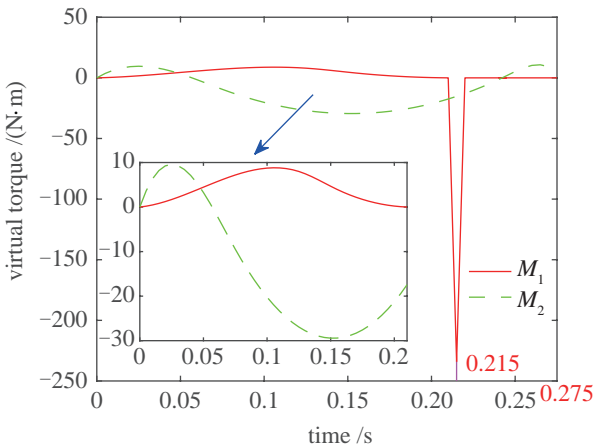


图 2 虚拟力矩输入 (示例 1)
Fig.2 The virtual torque inputs (case 1)

示例 1 经过 500 次迭代, 可得到近似最优解 $\Upsilon(\mathbf{M}) = 0.6944$, 虚拟力矩如图 2 所示, 实际控制输入和姿态角变化轨迹分别如图 3 和图 4 所示. 从图中可以得知, 当实际控制输入为 \mathbf{u} 时, 仅用 $T = 0.275 \text{ s}$ 就可实现机器人的翻正, 相对于文 [12] 提出的以能量最优实现翻正所耗时间 $T = 1 \text{ s}$ (如图 5、6 所示), 节约了近 3/4 的时间, 验证了本文提出方法的有效性. 仔细观察图 4 可以发现, 在 $T = 0.215 \text{ s}$ 时, 机器人就已翻正成功, 在以后的时间段 $[0.215, 0.275]$ 内, u_1 输入为 0, u_2 输入不为 0, 但输入的 u_2 丝毫不影响翻正效果, 仅仅改变了弯曲角. 为了验证这种现象是偶然还是必然, 进行了示例 2 实验, 结果如图 7~9 所示. 可以看出, 当 u_1 输入不为 0, u_2 输入为 0 时, 机器人的翻转角度随着 u_1 的输入而增加, 不受 u_2 为 0 的影响.

从示例 1 和示例 2 可以看出, u_2 仅仅改变脊柱弯曲角, 但不能引起翻转动作, u_1 是引起翻转动作的直接因素. 脊柱弯曲为落猫的翻转提供保障, 因为只有具有一定弯曲角, 落猫才能实现翻正 [7,9].

示例 3 结果表明, 可以得到更少的翻正时间 $T = 0.075 \text{ s}$, 如图 10~12 所示. 但是随之引起的问题就是需要输入很大的角速度才能达到预期效果,

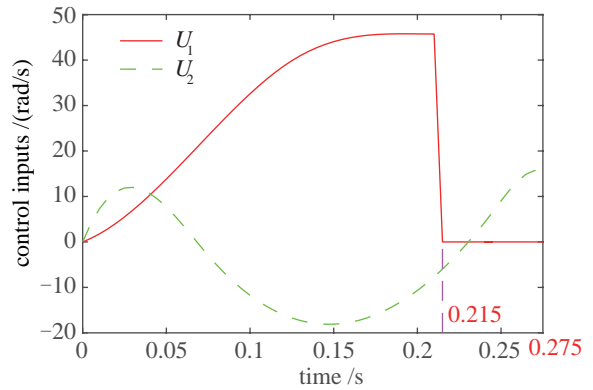


图 3 最优控制输入 (示例 1)
Fig.3 The optimal control inputs (case 1)

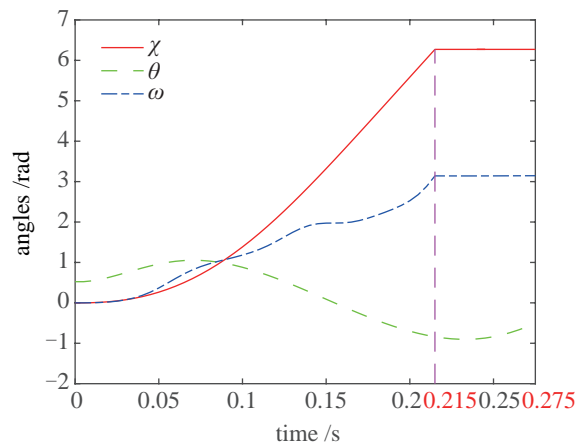


图 4 最优下落姿态角轨迹 (示例 1)
Fig.4 The optimal trajectories of attitude angles (case 1)

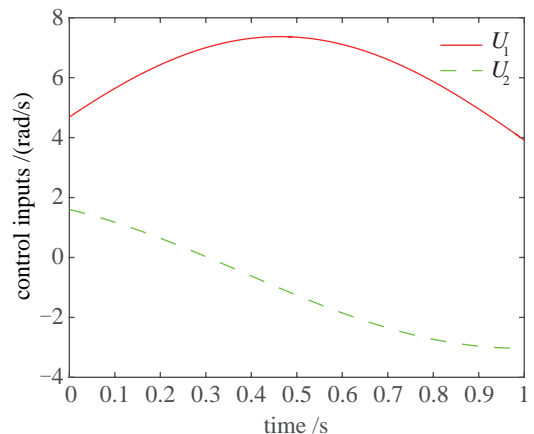


图 5 最优控制输入 (文 [12])
Fig.5 The optimal control inputs (Ref. [12])

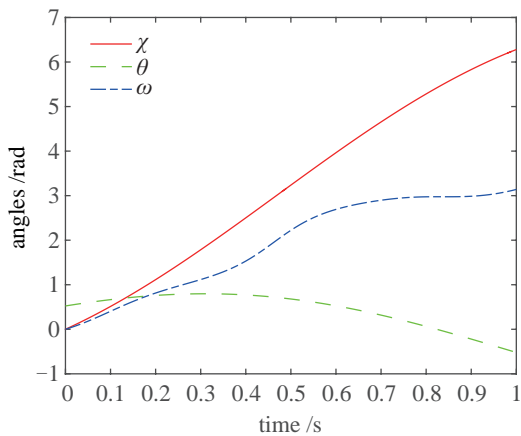


图6 最优下落姿态角轨迹 (文 [12])

Fig.6 The optimal trajectories of attitude angles (Ref. [12])

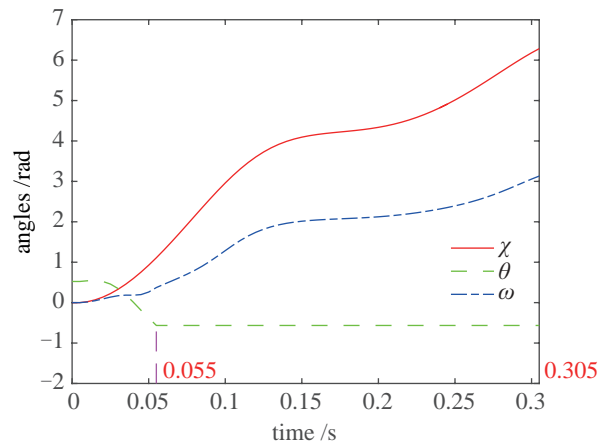


图9 最优下落姿态角轨迹 (示例2)

Fig.9 The optimal trajectories of attitude angles (case 2)

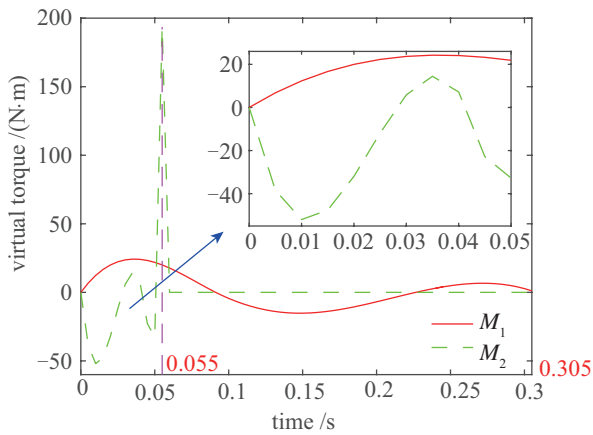


图7 虚拟力矩输入 (示例2)

Fig.7 The virtual torque inputs (case 2)

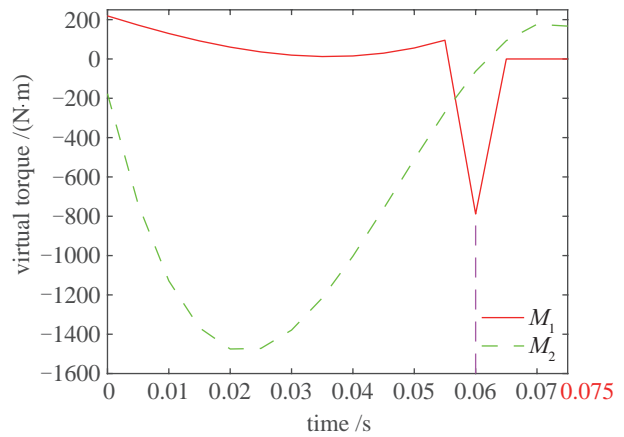


图10 虚拟力矩输入 (示例3)

Fig.10 The virtual torque inputs (case 3)

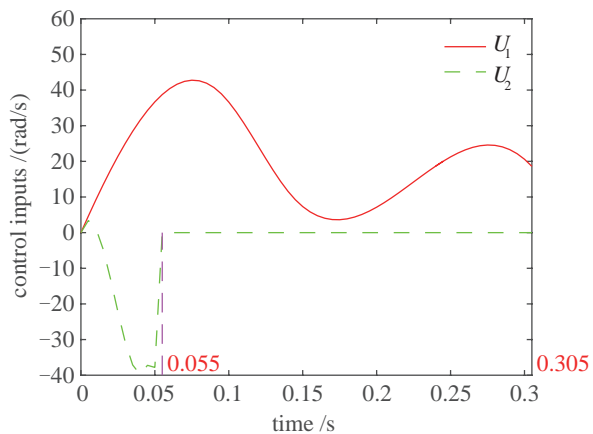


图8 最优控制输入 (示例2)

Fig.8 The optimal control inputs (case 2)

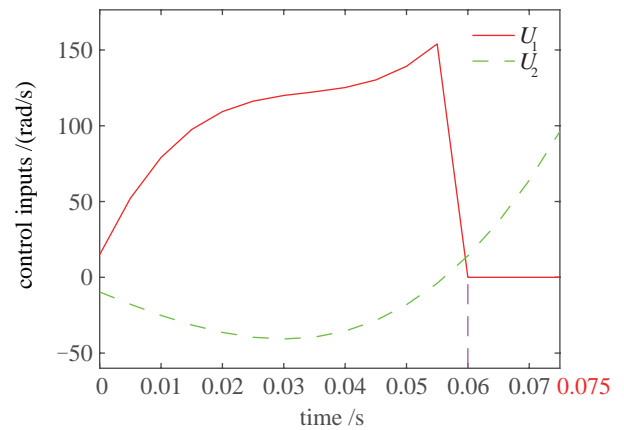


图11 最优控制输入 (示例3)

Fig.11 The optimal control inputs (case 3)

并且在 [0.055,0.06] 这段减速时间内,角加速度较大,因此对电机的性能要求很高,机器人需要承受的冲击力也较大。

从图2和图7可知,当翻转时间较长时,虚拟力矩除了在需要角速度迅速变为0这段时间较大之外,其余时间光滑渐变,大小可接受,一般电机都

能实现。从图10可知,当翻转时间较短时,不仅平时需要较大力矩,而且在需要角速度迅速变为0时,需要更大的力矩才能实现,一般电机很难满足要求。从角速度分析亦是如此。因此在设计自由落体机器人样机时,需要综合考虑翻正时间和电机性能,做到有所取舍。

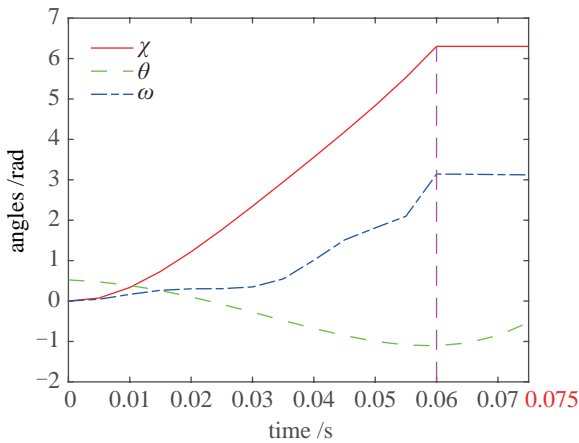


图 12 最优下落姿态角轨迹 (示例 3)

Fig.12 The optimal trajectories of attitude angles (case 3)

5.3 Adams 仿真结果分析

建立落猫机器人虚拟样机, 并在虚拟物理环境中实验来验证文中所提方法的有效性. 根据落猫空中姿态调整原理, 仿照 Kane-Scher 模型, 虚拟样机设计为前、后体对称分布且不可扭转、腰部可弯曲的物理模型, 如图 13 所示. 该样机使用电机分别驱动身体的转动以及腰部的弯曲, 同第 2 节所述相同, 四肢仅为了方便区分背部和腹部, 没有实际意义, 不影响实验结果. 在样机设计过程中, 首先用 SolidWorks 完成样机结构的设计和装配, 然后导入到 Adams 中进行下落仿真实验.

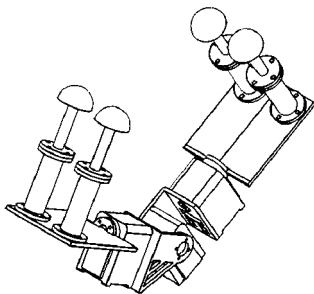
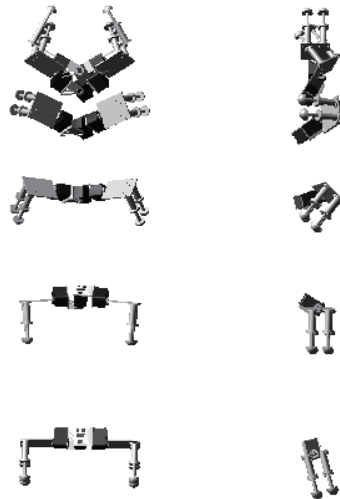


图 13 落猫机器人虚拟样机

Fig.13 The virtual prototype of a falling cat robot

受篇幅限制, 仅针对示例 1 的实验数据进行验证分析. 将示例 1 得到的各关节角速度数据作为虚拟样机中对应电机的角速度输出, 设置仿真步数为 500 步, 终止时间为 0.275 s, 进行仿真实验. 仿真结束后可得如图 14 所示的虚拟样机空中姿态调整过程, 从图中可知, 利用示例 1 的实验数据能够实现虚拟样机的空中翻正动作, 验证了本文所提方法的有效性. 为了便于比较, 将虚拟样机空中翻正角度和理论翻正角度的变化轨迹放在同一张图中, 如图 15 所示, 从图中可以看出, 虚拟样机空中翻转的角度变化虽然和理想情况有差别, 但是总体趋

势大致相同, 并且虚拟样机在 $T = 0.22$ s 时翻正成功, 和理论值 $T = 0.215$ s 很接近, 不过在样机翻正的 $[0.22, 0.275]$ 这段时间内有 8.3% 的超调量. 理论和实际情况存在误差的原因是样机参数和理论分析 (未考虑四肢、电机质量等) 中的参数不能完全匹配, 在进行真实样机试验时, 可通过 PID 控制、模糊控制等控制手段, 来减少由于样机本身原因造成的误差.



(a) 正面图

(b) 侧面图

图 14 虚拟样机姿态调整过程

Fig.14 The process of attitude adjustment of virtual prototype

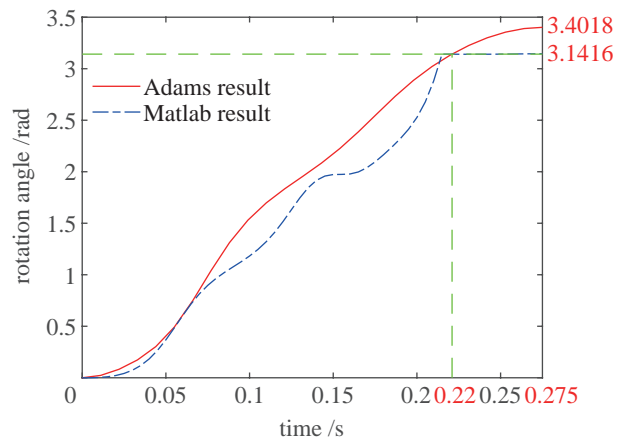


图 15 虚拟样机空中翻正角度和理论翻正角度对比

Fig.15 Comparison between the rotation angle of virtual prototype and the theoretical rotation angle

6 结论 (Conclusion)

猫从高处坠落能安全落地的能力一直被研究者们关注. 整个安全落地过程分为 2 个阶段, 姿态调整阶段和安全着陆阶段. 文中讨论了落猫机器人安全落地的第一阶段.

本文研究了落猫机器人以最短时间完成空中姿态调整时的最优下落轨迹. 针对角速度输入和翻

正时间的关系不易求出这一问题, 提出通过虚拟力矩输入方法来获得力矩和时间的关系, 进而建立时间优化函数, 并提出一种通过等分转动角度来求解 $\mathbf{x}(T)$ 的方法, 然后采用 PSO 算法求出目标函数的最优解, 最后建立了虚拟样机并实施了样机验证实验. 实验结果表明, 运用文中所提方法能够实现落猫机器人的空中翻正动作, 且以时间最优方式进行的姿态调整耗时要远少于以能量最优方式所得. 分析实验得到的结果, 也可以为以后样机的设计和控制电机的选取提供一些参考意见.

参考文献 (References)

- [1] 魏鲜明, 徐林森, 曹凯, 等. 足式水上行走机器人智能控制方法设计 [J]. 机器人, 2014, 36(1): 49-56.
Wei X M, Xu L S, Cao K, et al. Intelligent control method design of foot robot walking on water[J]. Robot, 2014, 36(1): 49-56.
- [2] 徐林森, 魏鲜明, 曹凯, 等. 仿生双足水上行走机器人优化设计及控制方法 [J]. 机械工程学报, 2014, 50(15): 12-18.
Xu L S, Wei X M, Cao K, et al. Optimization design and control method of biped robot running on water[J]. Journal of Mechanical Engineering, 2014, 50(15): 12-18.
- [3] Xu L S, Mei T, Wei X M, et al. A bio-inspired biped water running robot incorporating the Watt-I planar linkage mechanism[J]. Journal of Bionic Engineering, 2013, 10(4): 415-422.
- [4] Xu F Y, Wang X S, Jiang G P. Experimental study on the dynamic performance of a cable-robot detecting system[J]. Transactions of the Institute of Measurement and Control, 2016, 38(3): 338-347.
- [5] Xu F Y, Shen J J, Hu J L, et al. A rough concrete wall climbing robot based on grasping claws: Mechanical design, analysis and laboratory experiments[J]. International Journal of Advanced Robotic Systems, 2016, 13(5): 1-10.
- [6] Ge X S, Guo Z X. Nonholonomic motion planning for a free-falling cat using spline approximation[J]. Science China: Physics, Mechanics and Astronomy, 2012, 55(11): 2100-2105.
- [7] Kane T R, Scher M P. A dynamical explanation of the falling cat phenomenon[J]. International Journal of Solids and Structures, 1969, 5(7): 663-670.
- [8] Weng Z, Nishimura H. Final-state control of a two-link cat robot[J]. Advanced Robotics, 2002, 16(4): 325-343.
- [9] Ge X S, Zhao W J, Liu Y Z. Nonholonomic motion planning for a free-falling cat using quasi-Newton method[J]. Technische Mechanik, 2011, 31(1): 42-49.
- [10] Takahashi K. Remarks on motion control of nonholonomic system (falling cat) by using a quantum neural controller[C]//12th International Conference on Intelligent Systems Design and Applications. Piscataway, USA: IEEE, 2012: 961-966.
- [11] Nakano D, Maeda S, Ishii S. Control of a free-falling cat by policy-based reinforcement learning[C]//Artificial Neural Networks and Machine Learning. Berlin, Germany: Springer-Verlag, 2012: 116-123.
- [12] Liang X C, Xu L S, Li L. Research on trajectory planning of a robot inspired by free-falling cat based on modified quasi-Newton algorithm[C]//IEEE International Conference on Mechatronics and Automation. Piscataway, USA: IEEE, 2016: 552-557.
- [13] Liang X C, Xu L S, Li L, et al. Research on trajectory planning of a robot inspired by free-falling cat based on numerical approximation[C]//IEEE International Conference on Robotics and Biomimetics. Piscataway, USA: IEEE, 2016: 631-636.
- [14] Montgomery R. Isoholonomic problems and some applications [J]. Communications in Mathematical Physics, 1990, 128(3): 565-592.
- [15] Montgomery R. Gauge theory of the falling cat[J]. Fields Institute Communications, 1993(1): 193-218.
- [16] Fernandes C, Gurvits L, Li Z, et al. Near-optimal nonholonomic motion planning for a system of coupled rigid bodies[J]. IEEE Transactions on Automatic Control, 1994, 39(3): 450-463.
- [17] Neimark I, Fufaev N. Dynamics of nonholonomic systems[M]. Providence, USA: American Mathematical Society, 1972: 5-15.
- [18] Goldstein H. Classical mechanics[M]. 2nd ed. Mineola, USA: Dover Publications, 1980: 78-83.
- [19] Wen J T. Control of nonholonomic systems[M]//Levine W S. The Control Handbook. Boca Raton, USA: CRC Press, 1996: 1359-1368.
- [20] Poli R, Kennedy J, Blackwell T. Particle swarm optimization[J]. Swarm Intelligence, 2007, 1(1): 33-57.
- [21] Clerc M. Particle swarm optimization[M]. New York, USA: John Wiley & Sons, 2010: 3-58.
- [22] 龚纯, 王正林. 精通 MATLAB 最优化计算 [M]. 2 版. 北京: 电子工业出版社, 2012: 270-271.
Gong C, Wang Z L. Proficient in Matlab optimization calculation[M]. 2nd ed. Beijing: Publishing House of Electronics Industry, 2012: 270-271.

作者简介:

梁兴灿 (1989 -), 男, 硕士生. 研究领域: 机器人控制系统设计, 最优控制.

徐林森 (1981 -), 男, 博士, 研究员. 研究领域: 机构学与机械系统设计, 仿生机器人技术.

李露 (1981 -), 女, 博士, 副研究员. 研究领域: 机构设计, 仿生机器人和多体动力学.