文章编号：2095-4980(2025)04-0403-07

弹跳机器人跳跃姿态模糊自适应PID控制仿真

皮　明，杨　涛

（西南科技大学 信息工程学院，四川 绵阳 621010）

摘　要：针对现有的弹跳机器人无法对自身的跳跃姿态角度进行精确控制的问题，提出一种基于飞轮反作用力调节的弹跳机器人跳跃时的姿态控制方法。在跳跃时，通过飞轮连续转动时产生的反作用力，可以实现机器人姿态角度的调节，每个飞轮调节器使用模糊自适应比例-积分-微分(PID)控制器进行控制。相较于传统的PID控制器，模糊自适应PID能够更快，并以更小的跟踪误差实现机器人姿态角度的调节。仿真结果表明，相较于传统的姿态控制方式，采用飞轮调节能够更快地实现弹跳机器人姿态角度的实时控制。特别在机器人连续跳跃时，飞轮能够更好地吸收间歇性的冲击动量，使机器人姿态角在更大的范围内保持稳定。在未来工作中，会考虑搭建弹跳机器人实物平台，进行室内外环境的越障测试。

关键词：弹跳机器人；姿态控制；模糊自适应PID；联合仿真

中图分类号：TP241　　**文献标志码：**A　　**DOI：**10.11805/TKYDA2023202

Simulation of fuzzy adaptive PID control for hopping posture in hopping robots

PI Ming，YANG Tao

(School of Information Engineering，Southwest University of Science and Technology，Mianyang Sichuan 621010，China)

Abstract：Aiming at the problem that existing jumping robots cannot precisely control their jumping posture angles, a posture control method for jumping robots based on flywheel reaction force regulation is proposed. During the jump, the reaction force generated by the continuous rotation of the flywheel can be utilized to adjust the robot's posture angle. Each flywheel regulator is controlled using a fuzzy adaptive Proportional-Integral-Derivative(PID) controller. Compared with traditional PID controllers, fuzzy adaptive PID can more quickly adjust the robot's posture angle with smaller tracking errors.Simulation results show that, compared with traditional posture control methods, using flywheel regulation can achieve real-time control of the jumping robot's posture angle more quickly. Especially during continuous jumping, the flywheel can better absorb intermittent impact momentum, allowing the robot's posture angle to remain stable over a larger range. In future work, a physical prototype of the jumping robot will be considered for construction to conduct obstacle-crossing tests in both indoor and outdoor environments.

Keywords：jumping robot；attitude control；fuzzy adaptive PID；co-simulation

跳跃运动有助于机器人在崎岖地形和复杂环境中进行快速移动[1-2]。然而，现有的弹跳机器人无法对自身的跳跃姿态进行精确控制，无法实现正确的着陆姿态从而无法避免撞击地面时造成的机体损坏[3-4]。为此，本文提出了一种基于飞轮反作用力调节的弹跳机器人跳跃时的姿态控制方法。

事实上，弹跳机器人因在非结构化环境中出色的运动性能受到研究人员的广泛研究。研究人员设计了一种弹跳机器人，通过压缩和释放身体上的弹簧来实现敏捷的跳跃运动[5-6]。然而，尽管机器人可以实现任意方向的跳跃运动，但是机器人着陆时的姿态无法进行调节。为了控制机器人在着陆时的姿态角，研究人员改进了一种弹跳机器人，通过调整起跳发射角度，可使机器人获得更平稳的着陆姿态[7]。受小型昆虫跳跃能力的启发，研究人员设计了一种可在水面跳跃的机器人，机器人的着陆姿态由机器人的发射角度和跳跃高度共同决定[8]。然而，

收稿日期：2023-07-22；**修回日期：**2023-12-30

基金项目：四川省重点研发计划基金资助项目(2024YFFK0039)；西南科技大学博士基金资助项目(21zx7142)

上述方法通常需要复杂的机械结构，同时也无法实现机器人姿态的精确实时控制。

在自然界中，白头叶猴在跳跃过程中，可通过尾巴的摆动来调节身体姿态，受此启发，研究人员设计了一种具有摆杆的弹跳机器人，通过摆杆的连续旋转可以精确调整自身的姿势角度[9]。当机器人的跳跃速度不快时，现有的控制方法可使弹跳机器人的姿态获得较好的稳定性。然而，当机器人快速跳跃时，机器人起跳腿产生的冲击动量会变得非常大，机器人姿态将无法再保持稳定[10-11]。

为了吸收起跳腿产生的冲击动量，从而使机器人姿态保持稳定，本文提出了一种基于飞轮反作用力调节的弹跳机器人跳跃时的姿态控制方法。

1 机器人动力学模型

弹跳机器人由3部分组成：主体、起跳腿、姿态调节器。起跳腿在机器人主体的两侧，作为起跳装置。姿态调节器由4个飞轮组成，分布在机器人主体上，前一个飞轮和后一个飞轮用于调节机器人沿x轴方向(横滚)的姿态角度变化，顶部飞轮用于调节沿z轴方向(偏航)的姿态角度变化，内侧飞轮用于调节沿y轴方向(俯仰)的姿态角度变化。4个飞轮具有相同的直径和质量。为了简化分析过程，只对沿着y轴的飞轮调节进行分析。

如图1所示，反作用飞轮w位于弹跳机器人尾部，其转轴L在机器人上固定且与机器人质心坐标系轴B_y平行，机器人的质心为B，质心坐标系为B_{xyz}。假设飞轮整体设计匀质、对称，质量为m_w，飞轮转轴L相对于机器人质心坐标系的距离为r_w，则机器人(包括飞轮)绕B_y轴转动惯量为$I=I'_y+m_w r_w^2$，其中，I'_y为机器人(不包括飞轮)绕B_y轴的惯量。

若外力矩为零，则上述系统遵循动量矩守恒定理。机器人(含飞轮)对质心轴B_y方向的动量矩满足：

$$I\ddot{\theta}_y+I_w(\ddot{\theta}_y+\ddot{\theta}_w)=0 \tag{1}$$

式中：$\ddot{\theta}_y$为机器人主体绕轴B_y的角加速度；$\ddot{\theta}_w$为飞轮绕轴L的角加速度；I_w为飞轮对机器人系统的转动惯量。

假设$-T_r$为机器人对飞轮的转矩，T_r为飞轮对机器人的转矩，可以将式(1)推导为：

$$I_w\ddot{\theta}_y+I_w\ddot{\theta}_w=-T_r \tag{2}$$

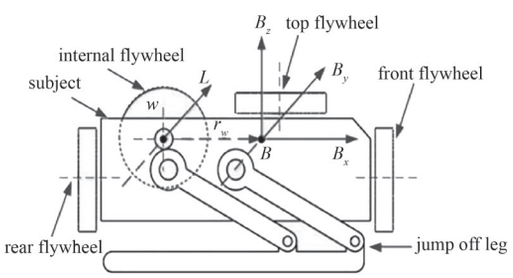

Fig.1 Dynamic model of robot
图1 机器人动力学模型

由式(1)、式(2)可得机器人动力学方程为：

$$\begin{cases} I\ddot{\theta}_y=T_r \\ I_w\ddot{\theta}_w=-T_r\left(1+\dfrac{I_w}{I}\right) \end{cases} \tag{3}$$

此外，有：

$$\begin{cases} \ddot{\theta}_y=\dfrac{T_r}{I} \\ \ddot{\theta}_w=-\dfrac{T_r}{I_w}\left(1+\dfrac{I_w}{I}\right) \end{cases} \tag{4}$$

当轴L穿过飞轮的中心时，飞轮的旋转运动不会改变机器人的质心。因此，I和I_w可以被视为常数，这表明$\ddot{\theta}_y$应该只由T_r决定，即θ_y应该只由T_r决定。同样，其他2个姿态角θ_x和θ_z也应该由它们相应的飞轮转矩决定。因此，可以通过控制相应的飞轮转矩来调节机器人的姿态角。姿态调节器通过飞轮的旋转来吸收间歇性冲击动量以保证连续跳跃过程中弹跳机器人姿态角的稳定。

2 姿态控制策略

飞轮控制系统的原理可以从动力学方程式(3)中概括出来。如图2所示，θ_d表示机器人姿态角的设定值。通过PID控制器，用θ_d和θ之间的差值e计算T_r。θ表示机器人的真实姿态角。作为对$-T_r$值的反作用，飞轮调节器将反向转矩T_r施加在机器人的主体上。然后，θ发生变化，并反馈到e。

在跳跃过程中，当机器人的姿态角始终保持在设定值附近时，机器人实现跳跃时的姿态稳定，通常假设θ_d的值为0°。

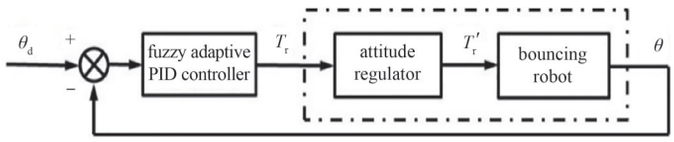

Fig.2 Principle diagram of posture angle adjustment
图2 姿态角度调节原理图

3　模糊自适应PID控制

模糊自适应PID控制算法是结合模糊控制和经典PID控制，通过模糊推理、反模糊化来实时调节PID控制参数k_P、k_I、k_D，用于弹跳机器人的姿态控制[12-13]。图3为模糊自适应PID控制系统方框图。

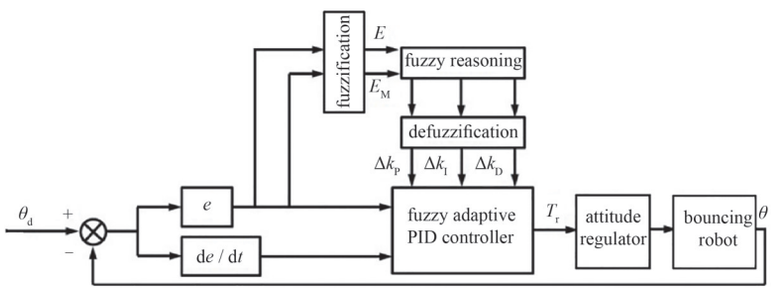

Fig.3 Block diagram of fuzzy adaptive PID control system
图3 模糊自适应PID控制系统方框图

在图3中，Δk_P、Δk_I、Δk_D分别为k_P、k_I、k_D的更新率。模糊控制器的输入变量为机器人姿态角误差e和误差变化率Δe，$e=\theta_d-\theta$，$\Delta e=\dfrac{de}{dt}$。e和Δe对应的模糊语言变量为E和E_M，经过模糊推理和模糊决策后得到模糊输出量R，再经过清晰化后得到精确的输出量T_r，也就是PID控制器的参数k_P、k_I、k_D。

模糊自适应PID控制参数自适应更新率为：

$$\begin{cases} k_P(t+1)=k_P(t)+\Delta k_P \\ k_I(t+1)=k_I(t)+\Delta k_I \\ k_D(t+1)=k_D(t)+\Delta k_D \end{cases} \tag{5}$$

式中：$k_P(t)$、$k_I(t)$、$k_D(t)$为当前时刻的控制参数；$k_P(t+1)$、$k_I(t+1)$、$k_D(t+1)$为更新后的控制参数。

控制器的输入e和Δe，以及控制器的输出Δk_P、Δk_I、Δk_D的模糊集定义如下：模糊语言变量的模糊集合均为{负大(NB)，负中(NM)，负小(NS)，零(ZO)，正小(PS)，正中(PM)，正大(PB)}。模糊论域划分：输入变量e、Δe和输出变量Δk_P、Δk_I、Δk_D的基础论域分别为{−60,60}、{−60,60}、{−30,30}、{−10,10}、{1.5,1.5}。将输入模糊语言变量E和E_M映射到{5,5}，将输出模糊语言变量k_P、k_I、k_D映射到{−3,3}。输入变量和输出变量的隶属函数均选择为三角形函数。模糊控制规则的建立采用经验归纳法，这里只给出k_P模糊规则的建立过程：

IF E=PB AND E_M=NB THEN R=ZO
IF E=PB AND E_M=NM THEN R=NS
IF E=PB AND E_M=NS THEN R=NS
IF E=PB AND E_M=ZO THEN R=NM
IF E=PB AND E_M=PS THEN R=NM
IF E=PB AND E_M=PM THEN R=NB
IF E=PB AND E_M=PB THEN R=NB

由此可以建立完整的模糊控制规则表。

4　控制仿真

利用Adams软件建立弹跳机器人动力学仿真模型，如图4所示。弹跳机器人由主体、起跳腿和姿态调节器

3 部分组成。本体(不含飞轮)设计参数为：长×宽×高=0.3 m×0.12 m×0.1 m，质量 m_1=5.4 kg；为实时测量本体与地面之间的夹角，在本体上设置虚拟角位移传感器坐标系 MARKER_41，角位移传感器函数为：AY(MARKER_41, MARKER_19)×180/pi，其中：MARKER_19 为地面标志坐标系。飞轮位于本体尾部，以本体为旋转关节，添加力矩驱动 VARIABLE_PID_torque，其产生的力矩值由模糊自适应 PID 控制算法反馈产生。飞轮设计参数为：半径 r=0.05 m，质量 m_2=1.2 kg。机器人起跳腿采用平面四连杆结构，分布于本体左右两侧，当起跳腿根部关节产生旋转运动时，起跳腿弹压地面，在反作用力下，使机器人本体飞起，实现弹跳运动，起跳腿弹跳驱动函数为：–STEP(time,0.0,0.0,0.05,90.0)×1d–STEP(time,0.15,0,0.2,–90.0)×1d。

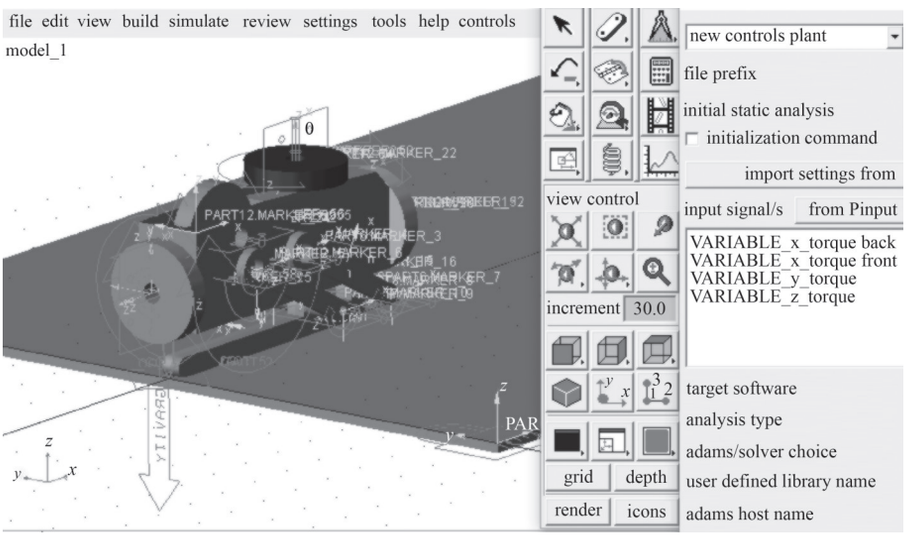

Fig.4 Co–simulation model of jumping robot
图 4 弹跳机器人联合仿真模型

利用 Matlab 软件的 Simulink 搭建弹跳机器人跳跃姿态控制系统，如图 5 所示。S 函数 adams–sub1 表示弹跳机器人的动力学模型，其输出为机器人的姿态角，输入为控制飞轮调节器的扭矩值，PID 控制器位于右侧。图 5 中 K–为增益。

在仿真中，当机器人的姿态角始终保持在设定值附近时，机器人实现跳跃时的姿态稳定。

在仿真开始时，腿部关节由 STEP 函数驱动产生机器人的跳跃运动。当一次跳跃结束时，机器人的起跳腿会先触及地面，由于地面反冲击力的作用，机器人姿态角会发生改变。与此同时，系统将姿态角度和角速度的值反馈给模糊自适应 PID 控制器。然后，如图 2 所示，控制器产生力矩来控制相应飞轮的转动以调节机器人的姿态角，从而抵消机器人受到地面反冲击力的影响。

在空中阶段，姿态角在短时间内调整到设定值。与此同时，机器人为下一次跳跃做准备。在第一次跳跃结束之后，机器人的起跳腿进行第 2 次跳跃。

仿真的时间段设置为 5 s。在仿真中，机器人起跳腿交替弹跳 6 次，每次弹跳耗时将近 0.7 s，这意味着左右两侧起跳腿各自弹跳 3 次。

5 讨论

与传统的姿态控制方法不同，弹跳机器人的姿态角由 4 组飞轮的转动进行调节。在每次起跳时，围绕 x 轴和 y 轴的姿态角由前飞轮、后飞轮和内飞轮调节，以保持机器人本体的底面与地面平行，这可以使机器人在起跳阶段和着陆阶段实现自身姿态的稳定。然后，在腾空阶段，围绕 z 轴的姿态角由顶部飞轮调节，以保持机器人偏航角的稳定，从而调整机器人向前的运动方向。

当弹跳机器人不采用姿态控制时，机器人在连续跳跃过程中的姿态角度变化如图 6 所示。在每次起跳后，每次跳跃都会使机器人的姿态角度进一步变大，始终无法恢复到设定值 0°。这表明弹跳机器人的弹跳姿态始终是不稳定的。事实上，在第 1 次跳跃仿真时，由于质心在推力的方向上，姿态角的变化很小；在第 2 次跳跃时(时间约为 0.7 s)，起跳腿推动地面产生的反作用力会使机器人的姿态变得不稳定，甚至使机器人在空中翻滚，每个姿态角变化更大。如图 6 所示，沿 x 轴方向的姿态角在不到 2 s 的时间内滚动了近 2 圈，沿 y 轴和 z 轴方向的姿态角度也发生了变化。

当弹跳机器人采用姿态控制时，机器人在连续跳跃过程中的姿态角度变化如图7所示。在起飞和腾空阶段，姿态角度的变化范围很小，最大变化角度仅为10°。姿态角度始终保持在设定值附近，使机器人在跳跃时姿态更加稳定。在3个姿态角中，沿z轴方向的姿态角度变化最大。由于其他2个姿态角的耦合影响，z轴的姿态角调整难度更大。z轴的姿态角反映了机器人的前进方向，其变化会导致运动方向的不确定性。在着陆阶段，由于受到地面冲击力的影响，z轴的姿态角变化也最大。

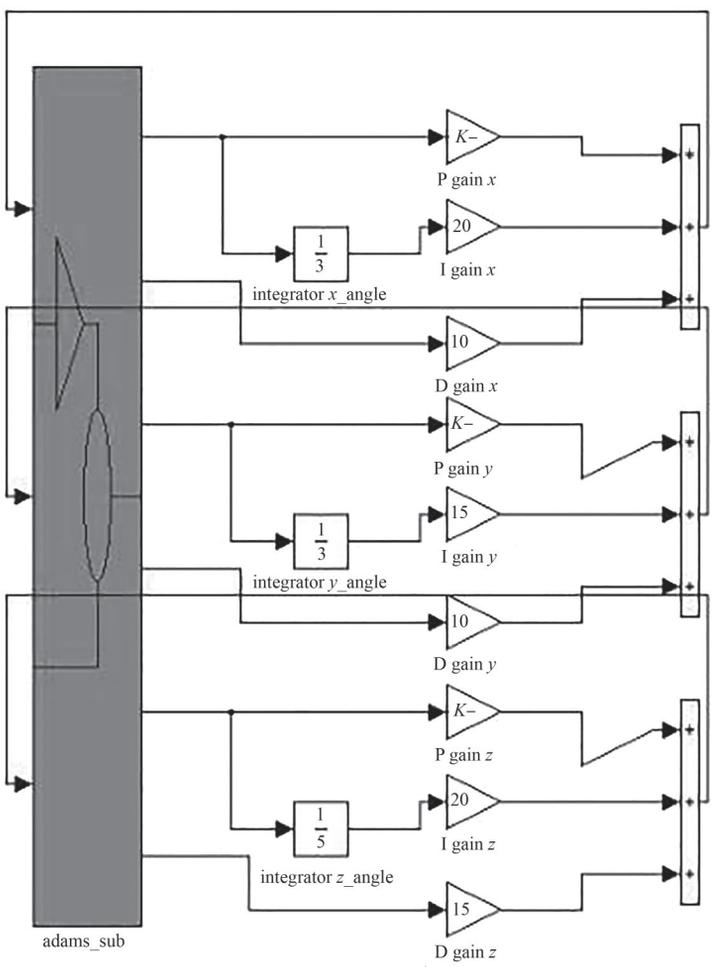

Fig.5 Simulation for hopping attitude control of jumping robot
图5 弹跳机器人跳跃姿态控制系统仿真

事实上，弹跳机器人姿态角的变化由起跳腿的推力和着陆时的姿态决定。起跳腿的推力是恒定的，这确保了每次弹跳力量相同。可以看出，外力、起跳腿的推力会影响机器人姿态角的变化。与此同时，飞轮的转动会吸收着陆时地面产生的冲击动量，使机器人不会被弹飞，使姿态保持稳定。但在一些特殊时刻，飞轮没有足够的时间来进行调节，此时机器人的姿态角就会发生剧烈的变化。如图7所示，当时间为1.4 s时，所有3个姿态角的变化都比其他任何时间剧烈。

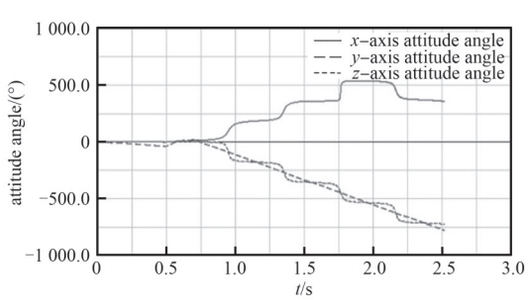

Fig.6 Change of posture angle of jumping robot without attitude control
图6 无姿态控制的弹跳机器人姿态角变化

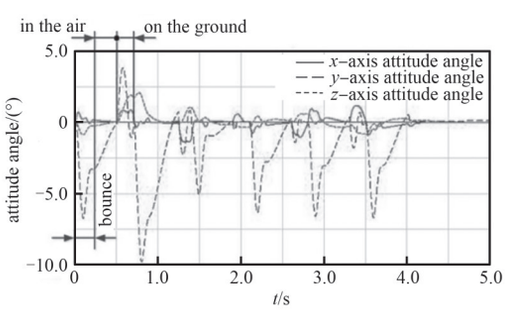

Fig.7 Change of posture angle of jumping robot with attitude control
图7 姿态控制的弹跳机器人姿态角变化

机器人跳跃过程中姿态控制力矩的变化如图 8 所示。由于机器人在起飞阶段的姿态角变化很大，飞轮的姿态控制力矩很大。然而，当机器人处于腾空阶段时，由于旋转的角速度恒定，飞轮的姿态控制力矩再次变小。姿态控制力矩沿 z 轴方向的最大范围接近 3 700 mm，也是 4 个飞轮姿态控制力矩中最大的。图 8 表明，在起跳阶段，机器人需要更大的姿态控制力矩来保持其姿态稳定；其次，在 y 轴方向上的机器人受到的外部冲击动量在姿态角度的干扰中占主导地位。因此，飞轮需要沿着 y 轴方向进行更精确的调节；最后，在每次跳跃都稳定的条件下，机器人姿态控制力矩的变化也呈现出规律性。

在图 9 中，弹跳机器人的中心位置在连续跳跃时发生了变化，机器人的中心位置在 4 s 内移动了约 6 m 的距离，每走一步消耗 0.7 s，移动的距离约为 1.1 m。因此，可以很容易地计算出弹跳机器人的平均速度为 1.57 m/s。沿着 x 轴方向的曲线有 6 个线段，每个线段都包括一段斜坡和一段水平直线。斜坡记录了机器人在 x 轴方向(向前移动方向)上的中心位置的移动，包括起跳阶段和腾空阶段。作为一条直线，斜率曲线可以被认为是一种匀速运动。在大约 0.2 s 的着陆阶段，飞轮通过吸收着陆时地面对机器人产生的冲击动量，调节机器人的姿态角，实现机器人姿态的再次稳定。

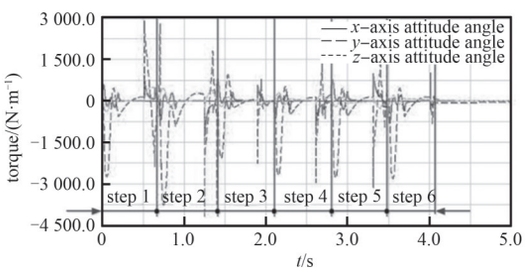

Fig.8 Change of torque of attitude control
图 8 姿态控制力矩的变化

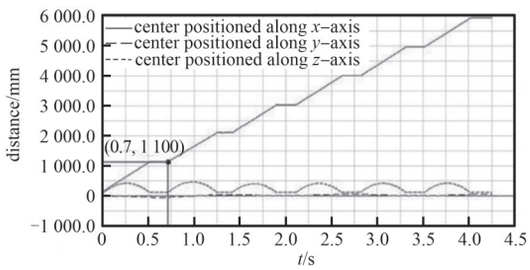

Fig.9 Change of the center position of jumping robot
图 9 弹跳机器人的中心位置的变化

如果飞轮在着陆阶段无法吸收地面接触力产生的冲击动量，那么下一次跳跃时机器人的姿态就可能会受到影响，变得不再稳定。当机器人进行跳跃时，沿垂直 z 轴方向的运动曲线就像一条抛物线。沿水平 y 轴方向的曲线表示的是前进方向的中心位置的偏移。中心位置在 y 轴方向上的偏移范围约为 5 mm，这意味着机器人在向前沿 x 轴运动时会在 y 轴方向上左右摆动。产生这种情况的主要原因是起跳腿接触地面时产生的推力线没有经过机器人的质心。当机器人的腿向外推进时，推力会产生一个横向力，使机器人沿水平 y 轴方向移动。当下一次跳跃发生时，产生的横向力的方向正好相反。

单次跳跃的运动过程如图 10 所示。每次跳跃都有相同的 3 个阶段：起飞阶段、腾空阶段和着陆阶段。在起飞阶段，起跳腿向后摆动并对地面施加推力，作为对推力的反应，机器人离开地面。与此同时，机器人在向前运动的同时会左右摆动。在腾空阶段，4 个飞轮将机器人的姿态角调节到设定值，以获得稳定的着陆姿态。这个过程大约 0.5 s。最后，机器人以稳定的着陆姿态降落在地面上。姿态调节器的飞轮在着陆阶段吸收冲击动量(这个过程大约 0.2 s)。

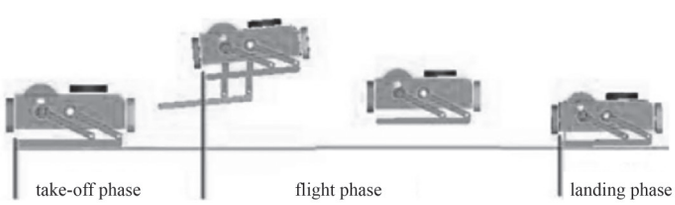

Fig.10 Jumping process of robot for one step
图 10 弹跳机器人一步跳跃过程

弹跳机器人跳跃的轨迹如图 11 所示。可以看出，机器人在 4 s 内跳跃了 6 次。机器人每条腿交替跳跃一次，实现了弹跳机器人向前的跳跃运动。由于起跳腿推力的作用，机器人的运动曲线会左右摇摆。在起飞阶段，当推力线不通过机器人的质心时，会在 y 轴方向产生一个横向力，使机器人左右摆动，这也使得弹跳机器人运动起来像人体在跑步时那样左右摆动。在第 1 次跳跃时会产生摆动，但在第 2 次跳跃时会产生方向相反的摆动。当跳跃次数为偶数时，机器人的质心总是可以回到向前运动的方向。

弗劳德数(F)是一个机器人运行速度的参考指数[14]，是一个无量纲数，其定义为：

$$F=\frac{V_{\text{avg}}}{\sqrt{Lg}} \tag{6}$$

式中：L 为机器人的腿长；V_{avg} 为机器人的平均前向移动速度。弹跳机器人的腿长 L=0.15 m，运行速度约为

5.4 km/h，即 F=1.22。与足腿式机器人相比，弹跳机器人在跳跃时通过飞轮转动保持身体稳定。弹跳机器人采取交替跳跃的方式，实现了机器人的高速连续跳跃。当每次跳跃之间的间隔越来越小时，机器人的运动就会更加平稳。

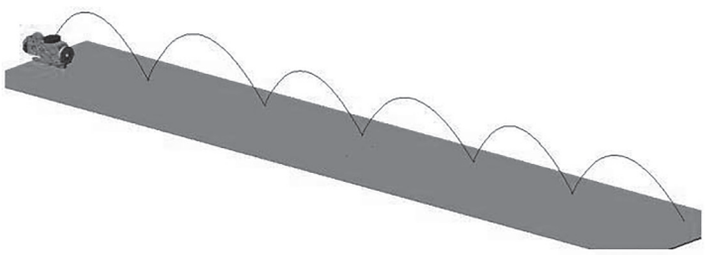

Fig.11 Trajectory of jumping movement of robot
图11 弹跳机器人跳跃的轨迹

6 结论

针对现有的弹跳机器人无法对自身的跳跃姿态角度进行精确控制的问题，本文提出了一种基于飞轮反作用力调节的弹跳机器人跳跃时的姿态控制方法。本文的贡献在于：首先，建立弹跳机器人的动力学模型，分析了机器人姿态角与飞轮转动之间的关系，概述了机器人姿态控制系统的基本原理；然后，采用模糊自适应PID控制器来控制飞轮的旋转从而调节机器人的姿态角；最后，搭建联合仿真平台，仿真结果表明，相较于传统的姿态控制方式，采用飞轮调节能够更快地实现弹跳机器人姿态角度的实时控制。特别在机器人连续跳跃时，飞轮能够更好地吸收间歇性的冲击动量，使机器人姿态角在更大的范围内保持稳定。在未来工作中，会考虑搭建弹跳机器人实物平台，进行室内外环境的越障测试。

参考文献：

[1] AHN D H, CHO B K. Online jumping motion generation via model predictive control[J]. IEEE Transactions on Industrial Electronics, 2022,69(5):4957–4965. DOI:10.1109/TIE.2021.3078396.

[2] SATO R,HIASA S,WANG Lei,et al. Vertical jumping by a legged robot with upper and lower leg bi–articular muscle–tendon complexes[J]. IEEE Robotics and Automation Letters, 2021,6(4):7572–7579. DOI:10.1109/LRA.2021.3099226.

[3] YIM J K,SINGH B R P,WANG E K,et al. Precision robotic leaping and landing using stance–phase balance[J]. IEEE Robotics and Automation Letters, 2020,5(2):3422–3429. DOI:10.1109/LRA.2020.2976597.

[4] NISHIKAWA S, ARAI Y, NIIYAMA R, et al. Coordinated use of structure–integrated bistable actuation modules for agile locomotion[J]. IEEE Robotics and Automation Letters, 2018,3(2):1018–1024. DOI:10.1109/LRA.2018.2794617.

[5] LI S G,RUS D. JelloCube:a continuously jumping robot with soft body[J]. IEEE–ASME Transactions on Mechatronics, 2019, 24(2): 447–458. DOI:10.1109/TMECH.2019.2899606.

[6] OSTRAICH B,RIEMER R. Simulation of a passive knee exoskeleton for vertical jump using optimal control[J]. IEEE Transactions on Neural Systems and Rehabilitation Engineering: a Publication of the IEEE Engineering in Medicine and Biology Society, 2020,28(12):2859–2868. DOI:10.1109/TNSRE.2020.3039923.

[7] MIYADAIRA A N,KOLM MADRID M,GARCIA J C,et al. Squat vertical jump of a 3DOF robot leg over an inclined plane: analysis with joint torque profile approximation[J]. IEEE Latin America Transactions, 2018,16(1):80–87. DOI:10.1109/TLA.2018.8291458.

[8] JIANG F,ZHAO J G,KOTA A K,et al. A miniature water surface jumping robot[J]. IEEE Robotics and Automation Letters, 2017, 2(3):1272–1279. DOI:10.1109/LRA.2017.2662738.

[9] JUNG G P,CASAREZ C S,LEE J,et al. JumpRoACH: a trajectory–adjustable integrated jumping–crawling robot[J]. IEEE–ASME Transactions on Mechatronics, 2019,24(3):947–958. DOI:10.1109/TMECH.2019.2907743.

[10] CHAE S H,BAEK S M,LEE J,et al. Agile and energy–efficient jumping–crawling robot through rapid transition of locomotion and enhanced jumping height adjustment[J]. IEEE–ASME Transactions on Mechatronics, 2022,27(6):5890–5901. DOI:10.1109/TMECH.2022.3190673.

（下转第422页）