摘录

通过零力矩点(ZMP)的预见控制，我们提出了一种新的双足机器人行走步态生成的方式。首先, 将双足机器人的动力学模型看作一辆移动的小车放在一个平台上, 给出了一个方便的表示方法来论述 ZMP。在回顾了传统的基于 ZMP 的步态生成方法之后, 我们将双足机器人行走问题转化为 ZMP 跟踪控制器的设计。通过使用未来信息的预见控制理论，可以实现这样一个控制器。结果表明, 使用预见控制器可以补偿简单模型与精确多体模型之间的 ZMP 误差。通过对螺旋台阶行走的仿真, 证明了该方法的有效性。

1. 绪言

双足仿生机器人的研究是目前机器人领域最激动人心的课题之一, 有许多正在进行的项目[1, 2, 3, 4, 21]。从控制和行走步态产生来看, 这些工作可以分为两种。第一种需要对机器人动力模型的精确建模, 包括质量、质心位置和每个环节的惯性, 以及行走前的状态。因此, 这主要依靠动力模型的准确性[1, 2, 15, 5]。我们将这一类称为 ZMP法, 因为它们经常使用零力矩点 (ZMP) 来进行步态生成和行走控制。

与第一种相反的，第二种方法只使用部分动力模型参数，例如总质心的位置, 总角动量等。由于控制器对系统结构知之甚少, 这种方法很大程度上依赖于反馈控制[6, 10, 7, 8]。我们可以称之为倒立摆法, 因为它们经常使用倒立摆模型。

本文基于第二种方法提出了一种行走控制和步态生成的方法，并在仿真和实验成功实现了动态双足行走。然而，因为我们通过改变脚步原本放置的位置来来产生稳定的步态，这种方法不能用来解决像在踏脚石上行走，需要将脚放到特定位置的这样的问题(Figure 1)。大多数倒立摆法都遇到了这个问题, 而基于 ZMP 的方法可以处理这种情况[15]。

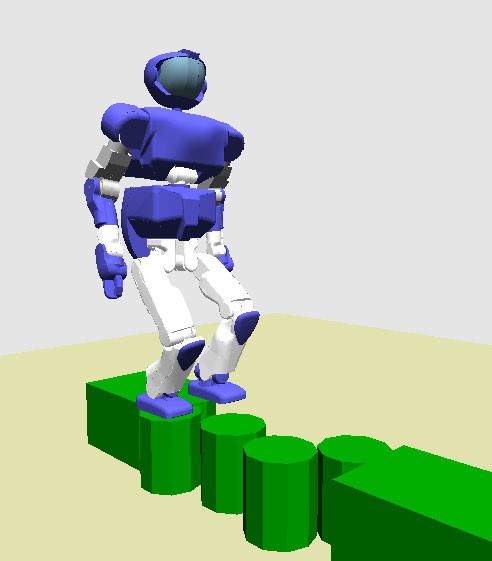


Figure 1: 在随机放置的踏脚石上行走

本文介绍了一种新的,允许将脚任意放置、混合了ZMP 法和倒立摆法的步态生成方法。同时, 通过使用预览控制器, 我们可以考虑到精确的多体系统动力学, 即使我们的方法是基于一个简单的倒立摆模型。

2 双足机器人的动力模型

2.1 3维线性倒立摆模型与零力矩点

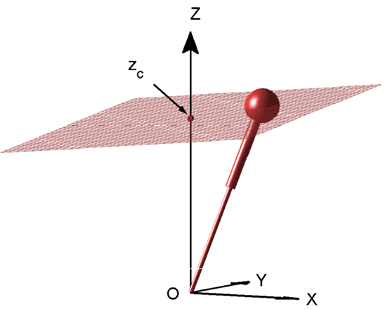
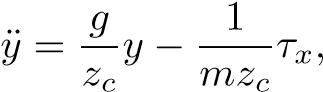


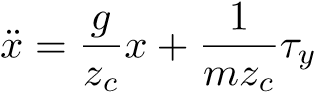
Figure 2: 约束下的倒立摆

当我们对倒立摆应用约束控制时, 机器人质心可以沿着任意定义的平面移动, 我们得到一个简单线性动力学模型称为三维线性倒立摆模型(3D-LIPM)[9, 11]。我们采用笛卡尔坐标，如图2所示，并将x轴方向定为行走的原点方向。约束面用给定的向量 (kx,ky,−1) 和与 z 轴的相交点zc表示

*z* = *kxx* + *kyy* + *zc.* (1)

如果约束平面是保持水平的(*kx* = *ky* = 0)，在约束控制下的动力模型给出如下

 (2)

*,* (3)

方程中 m 表示倒立摆的质量，*g*  是重力加速度，*τx,τy* 分别是围绕x 轴和y轴的扭矩。

即使在倾斜的约束平面下(Kx+ky != 0)，我们也可以通过添加输入扭矩的约束条件来获得同样的动力模型

*τxx* + *τyy* = 0*,* (4)

式(2) 和 式(3) 是线性方程组。控制这些动力学的唯一参数是zc ，即约束面与 z 轴的交点和平面的倾角不会影响水平运动。

对于具有水平约束的3维 LIPM(*kx* = *ky* = 0)，我们可以很容易地计算出广泛用于双足机器人研究的零力矩点(ZMP) [13]，

|  |  |  |  |
| --- | --- | --- | --- |
| *px* | = | *,* |  |
| *py* | = | *Τx*  *,*  *mg* | (5) |

(*px,py*)是ZMP在地面上的位置，将式(5)代入3维 LIPM的方程((2)和(3))中，可以得到

|  |  |
| --- | --- |
| *,* | (6) |
| *.* | (7) |

*zc*

2.2 ZMP方程和 车-平台模型

为了控制 ZMP, 它应该是系统的输出, 而当它出现在3维 LIPM 的输入中，因此, 我们重写方程(6) 及 (7) 将 ZMP 作为其输出, 得到

to have the ZMP as their outputs as

|  |  |
| --- | --- |
|  | (8) |
|  | (9) |

*g*

在本文的下面部分, 我们将把上述方程作为 ZMP 方程。

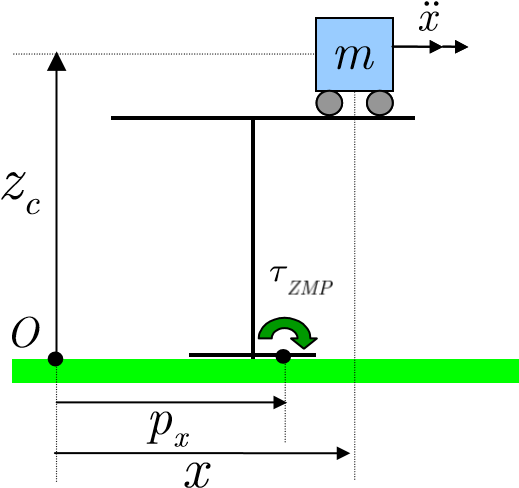


Figure 3: 车-平台模型

图3所示的模型直接对应于这些方程。这个模型描述了一个质量为m的小车在一个质量可以忽略的平台上运动(我们需要放置两辆小车在平台上来对应x，y的运动)

如图所示, 平台的脚太小, 不能让车停留在边缘。然而, 如果车加速以适当的速度, 可以保持平台直立一段时间。此时, ZMP 存在于平台的脚的内侧。由于此刻 ZMP 必定是零, 我们有

*τzmp* = *mg*(*x* − *px*) − *mxz*¨ *c* = 0(10)

我们可以用式9来验证，将得到同样的结果。

3 给定 ZMP 生成行走步态

3.1 将步态生成作为一个逆问题

当我们将机器人作为车-平台模型, 并把车的运动作为机器人质心 (CoM) 的轨迹时, 我们可以用 ZMP 方程(8) 及 (9)来方便地计算结果 ZMP。

另一方面, 步态的产生是这一逆问题。也就是说, 车的运动应该从给定的 ZMP 轨迹计算, 这是由所需的立足点和步进周期决定的。

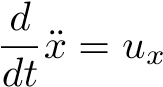
Takanishi 等人提出了用傅里叶变换求解这个问题 [14]。将快速傅里叶变换 (FFT) 应用于 ZMP 参考方程, ZMP 方程可以在频域中求解。然后, 逆 FFT 将结果的 CoM 轨迹返回到时域。

Kagami, Nishiwaki 等提出了一种在离散时域 [15] 中解决此问题的方法。结果表明, ZMP 方程可以离散为一个三项式表达式, 用 O (n) 算法可以在参考数据尺寸为 N时有效地求解。

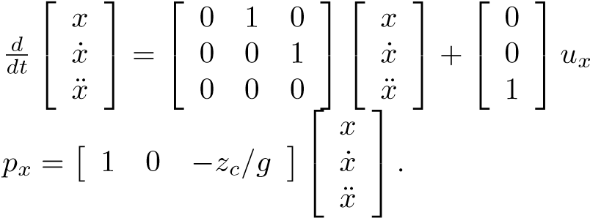
这两种方法都适用于在特定周期的参考 ZMP生成相应的 CoM 轨迹。为了产生长时间连续的步态，他们必须通过离线计算整个轨迹，或者必须将从参考 ZMP计算出来的轨迹分割成许多小段。

3.2 将ZMP 控制作为伺服控制问题

让我们定义一个新的变量 ux 作为 CoM 水平加速度在时间上的导数。

 (11)

将ux作为式(9)的输入，我们可以将 ZMP 方程转化为严格正则动力系统,

 (12)

对于式(8) 我们定义 uy, 获得相同形式的系统。

*x*

*u*

ZMP

reference

Servo

Controller

DynamicZMP

equation(12)

*p*

*ref*

*p*

*x*

+

−

*p*

ZMP

CoM

Figure 4: 步态生成作为 ZMP 跟踪控制

0

0.5

1

1.5

2

2.5

−0.05

0

0.05

0.1

0.15

0.2

0.25

0.3

x [m]

time [s]

zmp ref

x

Output

Input

x

ZMP

Figure 5: ZMP 和 CoM 的轨迹

使用动力方程(12)我们可以构建一个 ZMP 跟踪控制系统的步态发生器(图 4)。这个系统通过给定的参考zmp与实际zmp的反馈生成了CoM的轨迹。但是, 我们必须考虑这个问题的一个有趣的特点如下。图5说明了动态地向前迈进一步 ZMP 的和机器人的 CoM理想轨迹。机器人支持它的身体由后腿从0s 到 1.5s, 并且支持交换在1.5s 跟随前腿支持直到3.0s。因此, 参考 ZMP应该在1.5秒时变化，显然 CoM 必须在此之前移动。假设在图4中的控制器, 输出必须从未来的输入计算!

虽然这听起来很奇怪, 但我们不必违反因果律。事实上, 我们对在一条蜿蜒的道路上驾车的情况, 我们是很熟悉的, 在那里我们驾驶汽车, 看道路前方, 即看未来的参考。

一个利用未来信息的控制在1966年首次提出, 并被命名为 "预见控制"[16]。在 1969年, 早濑和市川提出了相同的概念, 解决了具有预览动作的线性二次 (LQ) 最优伺服控制器[17]。Tomizuka 和罗森塔尔在1979年开发了 LQ 最优预览控制器的数字版本 [18], Katayama等人在1985年完成了作为 MIMO 系统控制器的[19]。

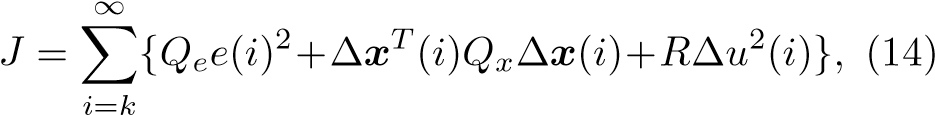
3.3 通过预见控制生成步态

让我们通过Katayama 等提出的的方法设计一个最佳的预览伺服控制器 [19]。

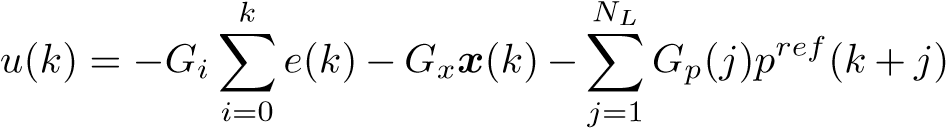
首先，我们定义一个采样周期T将式(12)的系统方程离散化。

|  |  |  |  |
| --- | --- | --- | --- |
| x(*k* + 1) = *A*x(*k*) + *Bu*(*k*)*,* | | |  |
| where | *p*(*k*) = *C*x(*k*)*,* | | (13) |
| x(*k*) | ≡ | [ *x*(*kT*) *x*˙(*kT*) *x*¨(*kT*) ]*T,* |  |
| *u*(*k*) | ≡ | *ux*(*kT*)*,* |  |
| *p*(*k*)  *A* | ≡  ≡ | *px*(*kT*)*,*   1 *T T*2*/*2   0 1 *T ,*   0 0 1  |  |
| *B* | ≡ | *,* |  |
| *C* | ≡ | [ 1 0 −*zc/g* ]*.* |  |

在给定参考ZMP函数pref(k)后，控制器的性能指标定义为



*e*(*i*) ≡ *p*(*i*)−*pref*(*i*) 为伺服控制误差，*Qe,R >* 0 并且*Qx* 是3×3对称非负正定矩阵。∆x(*k*) ≡x(*k*)−x(*k*−1) 是系统增量状态向量，以及∆*u*(*k*) ≡ *u*(*k*)−*u*(*k*−1) 是系统增量输入。

参考ZMP可以提前预见NL个采样时间，可以得到最小化性能指标(14)的最优控制器为(15)

其中 Gi、Gx 和 Gp (j) 是从权值*Qe,Qx,R*式(13)中系统参数计算所得的增益。

预见控制由三个部分组成, 即跟踪误差的积分、状态反馈以及使用将来参数的预先动作。

图6是预先动作的增益。我们可以发现控制器不需要太遥远的未来的数据，因为超过2秒后，预见增益值变化就非常小了

0

0.5

1

1.5

2

0

500

1000

1500

time [s]

preview gain

Figure 6: 预见控制增益 *Gp* (*T* = 5[ms], *zc* = 0*.*814[m], *Qe* = 1*.*0, *Qx* = 0, *R* = 1*.*0 × 10−6)

图7是1.6s 预见周期的行走模式生成示例。上图是沿 x 轴的运动, 下图为沿 y 轴的运动。我们可以看得到生成的CoM轨迹(虚线)比较平滑，并且得到的ZMP(粗线)精确的跟随参考ZMP(细线)。所生成的步态相当于向前走三步。参考ZMP设计为在单支撑阶段停留在支撑脚的中心, 并在双支撑阶段从旧支撑脚移动到新的支撑脚。为了在双腿支撑过程中获得平滑的 ZMP 轨迹, 我们使用了三次样条曲线。

图8是0.8s 预见周期的结果, 这对于 ZMP 跟踪来说是不够的。这样的控制下得到的ZMP(粗线) 跟随参考ZMP(细线)效果不是很好。

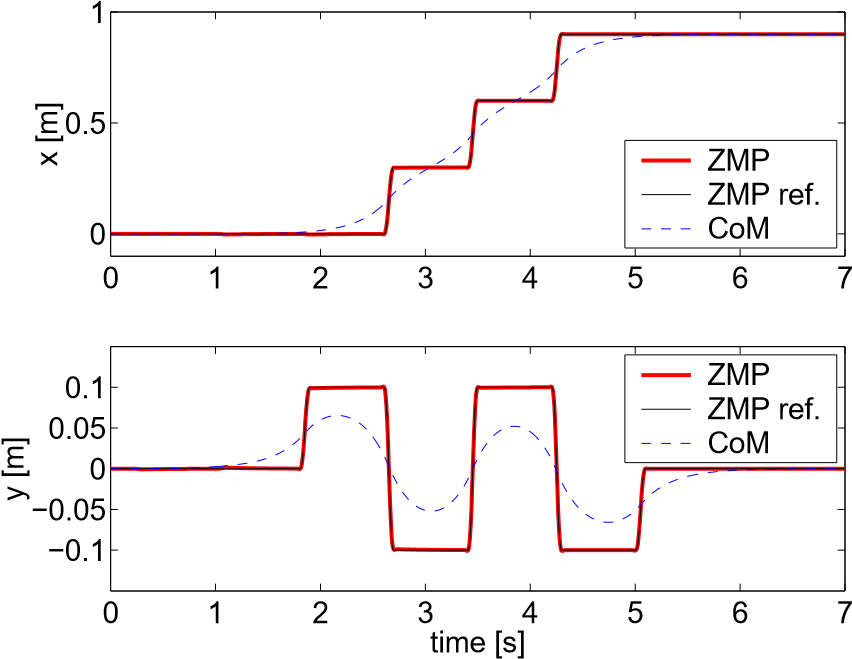
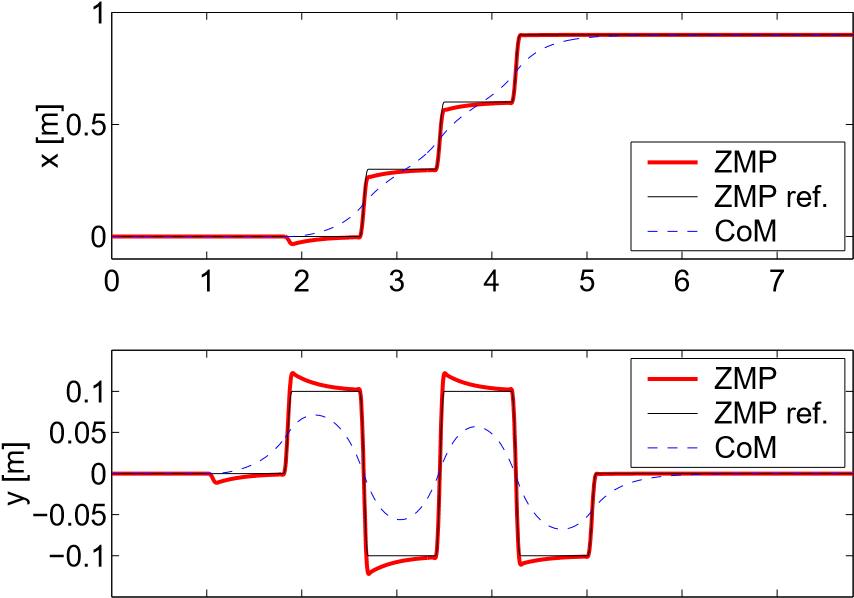


Figure 7: 通过预见控制的身体轨迹, 预见周期 *T* ∗ *NL* = 1*.*6(*s*)



0 1 2 3 4 5 6 7 time [s]

Figure 8: 使用更短的预见周期 *T* ∗ *NL* =0*.*8(*s*)