Bipedal Walking Robot Control

Nonlinear Control Project

Hao Su
Qiushi Fu
Overview

- Introduction
- PD control
- Virtual model control
- Robust adaptive virtual model control
- Central pattern generator
Introduction

- Model review

\[ A(q)\ddot{q} = b(q, \dot{q}, T, F) \]

Torso mass 5kg, length 0.8m;
Femur mass 2kg, length 0.5m;
Tibia mass 2kg, length 0.5m;

Ground contact parameters: \( k=10000, b=500 \);
Knee contact parameters: \( k=1000, b=100 \);
PD control

- **Control law**
  - *Event based, DSP & SSP*
  - *4 independent SISO, 2 knee, 1 hip, torso*
  - *Predefined trajectory for each phase to track*

- **Parameters**
  - *Different parameters in different phases*

<table>
<thead>
<tr>
<th></th>
<th>hip</th>
<th>knee</th>
<th>torso</th>
</tr>
</thead>
<tbody>
<tr>
<td>Both legs</td>
<td>$k_p = 60$, $k_\phi = 1$</td>
<td>$k_p = 40$, $k_\phi = 0.5$</td>
<td>$k_p = 40$, $k_\phi = 2$</td>
</tr>
<tr>
<td>Stance leg</td>
<td>$k_p = 70$, $k_\phi = 6$</td>
<td>$k_p = 30$, $k_\phi = 2$</td>
<td>$k_p = 40$, $k_\phi = 2$</td>
</tr>
<tr>
<td>Swing leg</td>
<td>$k_p = 70$, $k_\phi = 6$</td>
<td>$k_p = 10$, $k_\phi = 0.1$</td>
<td>$k_p = 40$, $k_\phi = 2$</td>
</tr>
</tbody>
</table>
PD control cont’d

- **Result**

  Normal walking

  ![Graph for Normal walking]

  Walking on upslope

  ![Graph for Walking on upslope]

  Walking with torso mass 8kg

  ![Graph for Walking with torso mass 8kg]
PD control cont’d

● Discussion
  ● Pros
    ● Simple
  ● Cons
    ● Basically rely on the linearized model
    ● Small working range
    ● Dedicate initial condition
    ● No adaptation to environment/mass change
Virtual Model Control (VMC)

- **Concept**
  - Task is specified in operational space by virtual components like spring, dampers
  - Virtual element produces force $F$, and joint torque needed to produce that virtual force can be computed with: $\tau = J^T F$
VMC cont’d

- Implementation

\[ F_x = b_x (\dot{x}_d - \dot{x}) \]
\[ F_y = k_y (y_d - y) + b_y (\dot{y}_d - \dot{y}) \]
\[ M_a = k_a (\alpha_d - \alpha) + b_a (\dot{\alpha}_d - \dot{\alpha}) \]
VMC cont’d

- Implementation

Swing leg strategy

\[ \tau_r = J^T F \]

\[
\begin{bmatrix}
0 \\
\tau_y \\
\tau_\beta
\end{bmatrix} = J^T_{3 \times 3} \begin{bmatrix}
f_x \\
f_y \\
f_\alpha
\end{bmatrix} \quad \begin{bmatrix}
\tau_y \\
\tau_\beta
\end{bmatrix} = M_{2 \times 2} \begin{bmatrix}
f_y \\
f_\alpha
\end{bmatrix}
\]

\[
\begin{bmatrix}
\bar{\tau}_l \\
\bar{\tau}_r
\end{bmatrix} = J^T_{6 \times 6} \begin{bmatrix}
\bar{F}_l \\
\bar{F}_r
\end{bmatrix} \quad \begin{bmatrix}
\tau_l^y \\
\tau_l^\beta \\
\tau_r^y \\
\tau_r^\beta
\end{bmatrix} = H_{4 \times 3} \begin{bmatrix}
f_x \\
f_y \\
f_\alpha
\end{bmatrix}
\]

Single Support Phase

Double Support Phase
VMC cont’d

- Result

**Vertical spring constant 5000, damping ratio 500**

1. **Normal walking**
2. **Walking upslope**
3. **Walking downslope**
VMC cont’d

- **Discussion**
  - **Pros**
    - Intuitive way of designing a controller
    - Robust to slope change
    - Does not need an accurate model of the environment
  
  - **Cons**
    - Need to make sure that the virtual forces can actually be generated by the robot’s motors
    - Finite-state machine for cycling through the different phases is a somewhat rigid mechanism
**VMC cont’d**

- More result

  Vertical spring constant 300, damping ratio 100
  Spring offset 0.35

Normal walking

Walking upslope

Walking with torso mass 8kg
Adaptive Virtual Model Control (AVMC)

- Sliding mode control
  - First drive system to stable manifold (reaching phase)
  - Then slide to equilibrium (sliding phase)

Define error as: \( \bar{x} = x - x_d \)

Define the error as: \( s(x, t) = \left( \frac{d}{dt} + \lambda \right)^{n-1} \bar{x} \)

Aim: \( \frac{1}{2} \frac{d}{dt} s^2 \leq -\eta |s| \)

\[
\ddot{x} = f + u \quad \Rightarrow \quad u = -\hat{f} + \ddot{x}_d - \lambda \dot{x} - k \text{sgn}(s)
\]

\[
\ddot{x} = f + \ddot{x}_d - \lambda \dot{x} - k \cdot \text{sat}\left( \frac{s}{\phi} \right)
\]
AVMC cont’d

- Adaptive control for second order system

Dynamic equation of 2nd order system

\[ a_1 \ddot{y} + a_2 f(\dot{y}, y) + a_3 g(y) + d(\dot{y}, y, t) = u \]

Define

\[ \dot{\tilde{y}} = y - y_d \]

\[ s = \dot{\tilde{y}} + \lambda \tilde{y} \quad s_\phi = s - \phi \text{sat}\left(\frac{s}{\phi}\right) \]

\[ Y = [\ddot{y}_r \ f \ g] \quad \hat{a} = [a_1 \ a_2 \ a_3] \]

Lyapunov Function

\[ V = \frac{a_1}{2} s_\phi^2 + \frac{1}{2} \tilde{a}^T \Gamma^{-1} \tilde{a} \]

Adaptive control law

\[ u = Y \hat{a} - ks \quad \dot{\hat{a}} = -\Gamma Y^T s_\phi \]
AVMC cont’d

- Implementation
  Y-direction dynamic model of massless leg

\[ m\ddot{y} + mg + d(p, \dot{p}, t) = F_y \]

Define

\[ \tilde{y} = y - y_d \]

\[ s = \dot{\tilde{y}} + \lambda \ddot{y} \quad s_\parallel = s - \phi \text{sat} \left( \frac{s}{\phi} \right) \]

\[ Y = \ddot{y}_r + g \quad \hat{\alpha} = m \]

\[ V = \frac{a_1}{2}s_\parallel^2 + \frac{1}{2}\tilde{a}^T \Gamma^{-1}\tilde{a} \]

Adaptive control law

\[ F_y = Y\hat{\alpha} - ks \quad \hat{\alpha} = -\Gamma Y^T s_\parallel \]
AVMC cont’d

- **Result**
  
  Normal walking

  ![Graph](image1.png)

  Torso mass 13 kg

  ![Graph](image2.png)

  Torso mass 8 kg on slope

  ![Graph](image3.png)
AVMC cont’d

- Discussion
  - Pros
    - Does not need an accurate model of the environment
    - Robust against disturbance and unmodeled dynamics
CPG

- Model

- Problem
  - Fewer mathematical tools than other methods
  - Not (yet) a clear design methodology, it is recommended to use learning/optimization algorithms
CPG cont’d

- result

Normal walking

failed1

failed2
References


Thanks

- Questions?