

3次元2脚ロボットの旋回着地制御

Rotational-Touchdown Control of a 3D Biped Robot

○玄相昊*, 江村超*

○Sang-Ho Hyon*, Takashi Emura*

*東北大学

*Tohoku University

キーワード： 2脚ロボット (biped robot), 走行制御 (running control), 着地制御 (touchdown control), 空中姿勢制御 (aerial posture control), 入出力線形化 (input-output linearization)

連絡先： 〒980-8579 仙台市青葉区荒巻字青葉01 東北大学大学院工学研究科 機械電子工学専攻 江村研究室
玄相昊, Tel.: (022)217-6969, Fax.: (022)217-6967, E-mail: sangho@ieee.org

1. はじめに

3次元走歩行が可能な2脚ロボットの実現のために、本研究室で開発中の Fig. 1のような2脚ロボット¹⁾を対象とした制御則を考える。これまで様々な走行ロボットが開発されているが²⁾、本研究では、いかに少ないアクチュエータで動的かつ多様な運動を実現するのかを研究することで、脚ロボットの動的運動制御の本質を探ることを主眼にしている。本ロボットはそのためのテストベンチであり、同種のロボットは著者の知る限り、他には存在しない。制御の基本的な思想は、文献⁴⁾と同様、軌道安定化のための基本制御をベースとして施し、その結果現れる歩容を、様々なタスクに応じて整形していくというものである。本稿ではその第一ステップとして、軌道安定化の重要な部分を占める、着地制御則を提案する。

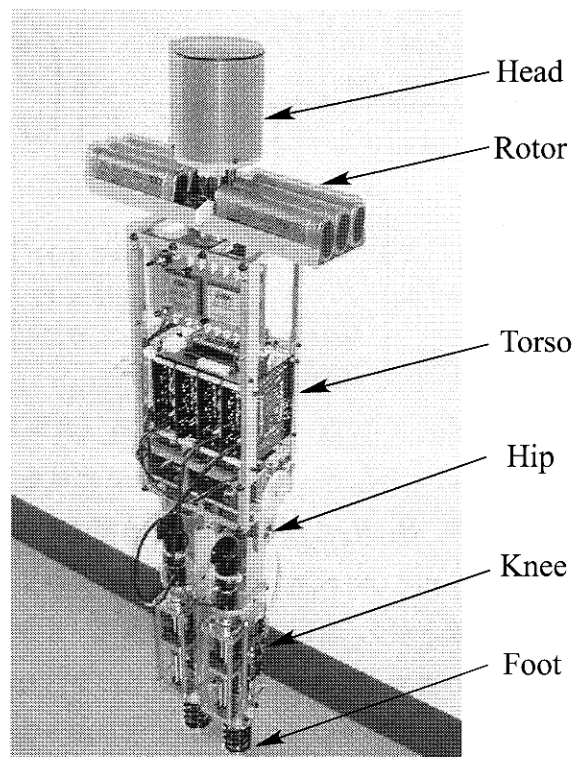


Fig. 1 3D biped prototype

2. モデル

Figure 2のような単純化したモデルを考える。胴体姿勢を表す e_1, e_2, e_3 はそれぞれ, Roll角, Pitch角, Yaw角を表している¹。これはFig. 1を単純化したモデルであるが, 著者等が既に走行制御則を導出した平面2脚ロボットのモデル³⁾にYaw軸まわりに回転するロータ(以下, Yaw軸補償器と呼ぶ)を追加することによって, 3次元に拡張したものとなっている。

2.1 運動方程式

本稿で取り扱うのは3次元走行のための空中姿勢制御なので, 浮遊期における運動方程式のみ示す。

運動方程式は, 重心位置 $x = (x_g, y_g, z_g)^T \in R^3$, 胴体姿勢 $e = (e_1, e_2, e_3)^T \in R^3$, および関節角 $p = (\psi_1, \psi_2, \psi_3)^T \in R^3$ からなる一般化座標を用いて

$$\left(\begin{array}{ccc|cc} \left[\begin{array}{ccc} M & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & M \end{array} \right] & & & 0(3 \times 6) & \\ \hline & & & \left[\begin{array}{cc} N_{11} & N_{12} \\ N_{21} & M_{22} \end{array} \right] & \end{array} \right) \begin{pmatrix} \ddot{x} \\ \dot{e} \\ \ddot{p} \end{pmatrix} \quad (1)$$

$$= \begin{bmatrix} 0 \\ -Mg \\ 0 \\ H_1(e, p, \dot{e}, \dot{p}) \\ H_2(e, p, \dot{e}, \dot{p}) \end{bmatrix} + \begin{bmatrix} 0(3 \times 1) \\ 0(3 \times 1) \\ \tau \end{bmatrix}$$

と表せる。ここで, $M = 2m + M_b$ (全質量), $N_{11}, N_{21} = N_{12}^T, N_{22} \in R^{3 \times 3}$ は e, p だけを含む慣性行列, $H_1, H_2 \in R^{3 \times 1}$ は e, p, \dot{e}, \dot{p} だけを含む非線形項, $\tau = (\tau_1, \tau_2, \tau_3)^T \in R^3$ は一般化力 (τ_1, τ_2 は股関節トルク, τ_3 はYaw軸補償器のトルク)をそれぞれ表す。重心 x の運動は取るに足らないので無視し, 以下では本式において水平線で区切られた下の部分, すなわち e, p に関するダイナミクスに着目する。

¹この局所座標だと直立姿勢が特異姿勢にならないので都合である。

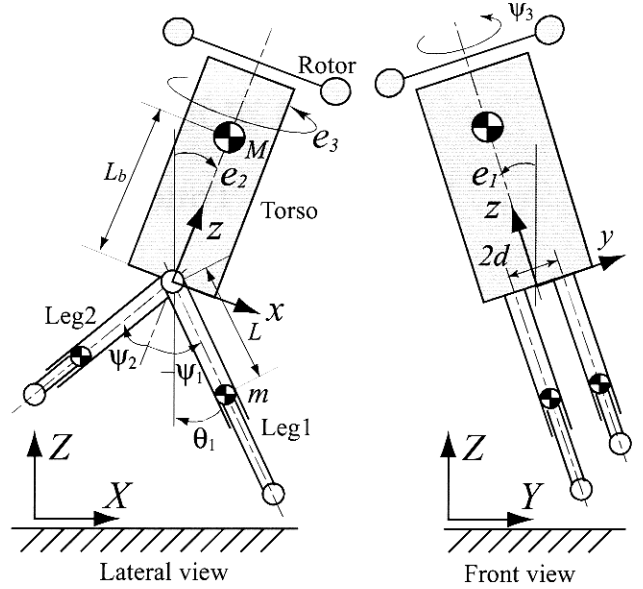


Fig. 2 Simplified model

3. 制御則

本ロボットを3次元空間内で走らせる際に問題となるのが横方向の安定性である。なぜなら本ロボットはRoll軸アクチュエータを持たないため, 横方向に対して不可制御かつ不安定だからである²。そこで, Yaw軸補償器を用いて転倒する方向にロボットを回頭させるというのが, 本稿における制御の基本的なアイデアである。

本節では, そのための水平姿勢制御, 仮想角運動量拘束制御, そして遊脚に関する部分線形化について述べる。

3.1 非干渉化

まず, 見通しを良くするために, 運動方程式を非干渉化する。

(2)式の注目する部分を改めて書き下すと,

$$N_{11}\ddot{e} + N_{12}\ddot{p} + H_1 = 0 \quad (2)$$

$$N_{21}\ddot{e} + N_{22}\ddot{p} + H_2 = \tau \quad (3)$$

²脚間距離 $2d$ を大きく取るか, または横に広い足平を取り付けば制御しなくても安定となることがわかっているが, そのようなモデルは興味の対象外である。

となる．ここではロボットの胴体姿勢を表す e に関して非干渉化する．(2) 式と (3) 式から \dot{p} を消去し， \ddot{e} に関して整理すると

$$\ddot{e} = \frac{N_{22}N_{12}^{-1}H_1 - H_2 + \tau}{(N_{21} - N_{22}N_{12}^{-1}N_{11})} =: v \quad (4)$$

なる非干渉化システムが得られる．ここで v は新しい入力である．

3.2 水平姿勢制御

転倒方向に回頭するという事は，すなわち胴体に固定された局所的なピッチ軸を水平に保つという制御である．これは，ローカル座標軸 y の単位ベクトルを絶対座標系 (X-Y-Z) で表したときの Z 成分が零になるということで定式化できる．すなわち，

$$\begin{aligned} & \left[E_e \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \right]_Z \quad (5) \\ & = \cos e_1 \sin e_2 \sin e_3 + \sin e_1 \cos e_3 = 0 \end{aligned}$$

ここで， E_e は座標変換行列である．

次に，これを「入出力線形化」あるいは「出力零化」の考え方をを用いて制御を行う．すなわち，

$$\alpha := \cos e_1 \sin e_2 \sin e_3 + \sin e_1 \cos e_3 \quad (6)$$

なる出力関数を定義し，これを零化する．出力零化には，次式を用いる．

$$\ddot{\alpha} + k_{ad}\dot{\alpha} + k_{ap}\alpha = 0 \quad (7)$$

ここで k_{ad}, k_{ap} は正の定数である．一方， $\ddot{\alpha}$ を展開すると

$$\ddot{\alpha} = a_1\ddot{e}_1 + a_2\ddot{e}_2 + a_3\ddot{e}_3 + a_4 \quad (8)$$

のように表すことができる．ここで， a_i ($i = 1, 2, 3, 4$) は e, \dot{e}, \ddot{e} を含む非線形項である．したがって，(7) 式を達成するために \ddot{e} が満足すべき条件は次式のようなになる．

$$(a_1 \ a_2 \ a_3) \begin{pmatrix} \ddot{e}_1 \\ \ddot{e}_2 \\ \ddot{e}_3 \end{pmatrix} = -(a_4 + k_{ad}\dot{\alpha} + k_{ap}\alpha) \quad (9)$$

3.3 仮想角運動量拘束制御と遊脚ダイナミクスの部分線形化

(2) 式はノンホロノミックな角運動量保存則を表している．しかし走行ロボットにおいては，ノンホロノミック性を利用して空中で複雑な姿勢制御を行うのは現実的ではないと考える．また，胴体姿勢 e を強引に任意の値に制御することも可動角の制限などから不可能である．そこで次のような目標ダイナミクスに従うような制御を施す．

$$\dot{\phi} = \dot{\phi}(0) \quad (10)$$

$$\ddot{\psi}_1 = u_1 \quad (11)$$

ここで， $\phi := \cos^{-1}(\cos e_1 \cos e_2)$ は，絶対鉛直軸から測った胴体の絶対姿勢角を表す．

(10) 式は，姿勢 ϕ に関して「仮想的な角運動量保存則」を設定するという事を意味している．本式は制御なしでは成り立たないので，モデル追従制御によってこれを達成する．なお，本式は両脚を対称に動かしたとき ($\dot{\psi}_1 + \dot{\psi}_2 = 0$)，「近似的に」成り立つことを補足しておく．したがって，これは走行運動においては直感的に最もらしい目標ダイナミクスと言えるだろう．

一方，(11) 式は遊脚の一つを新しい制御入力 u_1 によって自由に制御するための部分線形化である．これは次節で述べる遊脚の着地制御に用いられる．

まず，(10) 式に関しては，前節と同様に出力関数を定義してそれを零化する．今の場合の出力関数は

$$\beta := \dot{\phi} - \dot{\phi}(0) \quad (12)$$

である．次に，

$$\dot{\beta} + k_b\beta = 0 \quad (13)$$

(ここで k_b は正定数) によって出力零化を達成する．したがって，(12) 式と (13) 式を合わせると， \ddot{e}

が満足すべき条件は次式のようになる。

$$\ddot{\phi} = -k_b\beta \quad (14)$$

一方、 $\ddot{\beta}$ を展開すると

$$\ddot{\beta} = b_1\ddot{e}_1 + b_2\ddot{e}_2 + b_3\ddot{e}_3 + b_4 \quad (15)$$

のように表すことができる。ここで、 b_i ($i = 1, 2, 3, 4$) は e, p, \dot{e}, \dot{p} を含む非線形項である。したがって、次式を得る。

$$(b_1 \ b_2 \ b_3) \begin{pmatrix} \ddot{e}_1 \\ \ddot{e}_2 \\ \ddot{e}_3 \end{pmatrix} = -(b_4 + k_b\beta) \quad (16)$$

次に、(11)式は次のようになる。(3)式の第一行を抜き出すと、

$$\ddot{\psi}_1 = -(c_1\ddot{e}_1 + c_2\ddot{e}_2 + c_3\ddot{e}_3 + c_4) \quad (17)$$

となる。ここで、 c_i ($i = 1, 2, 3, 4$) は e, p, \dot{e}, \dot{p} を含む非線形項である。したがってこれを(11)式に代入して

$$(c_1 \ c_2 \ c_3) \begin{pmatrix} \ddot{e}_1 \\ \ddot{e}_2 \\ \ddot{e}_3 \end{pmatrix} = c_4 + u_1 \quad (18)$$

を得る。

3.4 遊脚の着地制御（有限整定）

ここでの制御目標は、ロボットの局所的な $x-z$ 平面から測った遊脚絶対角 θ_1 (Fig. 2) とその角速度 $\dot{\theta}_1$ を、定められた時刻 T_v である任意の目標値に整定することである。ただし、 $\theta_1 = \psi_1 + \phi$ 、 $\dot{\theta}_1 = \dot{\psi}_1 + \dot{\phi}$ である。

本ロボットは文献⁴⁾とは異なり、現在のところ腰関節にバネを持っていないので、最も簡単な有限整定制御を行う。(11)式は2階の線形微分方程式なので、2回の入力切替で ψ_1 および $\dot{\psi}_1$ を有限整定できる。まず、変数

$$\Phi := \begin{pmatrix} \phi \\ \dot{\phi} \end{pmatrix}, \quad \Theta := \begin{pmatrix} \theta_1 \\ \dot{\theta}_1 \end{pmatrix} \quad (19)$$

を定義しておく。

浮遊時間を二分した時間で系を離散化し、制御入力 u_1 を次式のように定義する。

$$u_1 = \begin{cases} \hat{u}_1, & \text{if } 0 \leq t < T_v/2 \\ \hat{u}_2, & \text{if } T_v/2 \leq t < T_v, \end{cases} \quad (20)$$

有限整定制御入力は

$$\begin{pmatrix} \hat{u}_1 \\ \hat{u}_2 \end{pmatrix} = B(T_v)^{-1} \{ \Psi(T_v) - A(T_v)\Psi(0) \} \quad (21)$$

となる。ここで、

$$A(T_v) = \begin{pmatrix} 1 & T_v \\ 0 & 1 \end{pmatrix}, \quad (22)$$

$$B(T_v) = \begin{pmatrix} \frac{3}{8}T_v^2 & \frac{1}{8}T_v^2 \\ \frac{1}{2}T_v & \frac{1}{2}T_v \end{pmatrix} \quad (23)$$

である。また、

$$\Psi(T_v) = \Theta_d - A(T_v)\Phi(0) \quad (24)$$

である。 Θ_d は任意の着地目標角である。

3.5 制御入力の計算

(9)式,(16)式,および(18)式から、

$$\begin{aligned} & \begin{pmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{pmatrix} \begin{pmatrix} \ddot{e}_1 \\ \ddot{e}_2 \\ \ddot{e}_3 \end{pmatrix} \\ &= \begin{pmatrix} -a_4 - k_{ad}\dot{\alpha} - k_{ap}\alpha \\ -b_4 - k_b\beta \\ -c_4 + u_1 \end{pmatrix} \quad (25) \end{aligned}$$

を得る。ここで u_1 は(20)式と(21)式から計算された値を代入する。本式より胴体姿勢角の目標角加速度 \ddot{e}_d を計算できる。すなわち、

$$\begin{aligned} e_d &= \begin{pmatrix} \ddot{e}_1 \\ \ddot{e}_2 \\ \ddot{e}_3 \end{pmatrix}_d \\ &= \begin{pmatrix} a_1 & a_2 & a_3 \\ 0 & 1 & 0 \\ c_1 & c_2 & c_3 \end{pmatrix}^{-1} \begin{pmatrix} -a_4 - k_{ad}\dot{\alpha} - k_{ap}\alpha \\ -b_4 - k_b\beta \\ -c_4 + u_1 \end{pmatrix} \quad (26) \end{aligned}$$

これを(4)式の \ddot{e} に代入すれば、実際にロボットに加える制御トルク τ が得られる。

4. シミュレーション

ロボットのパラメータは

$$M_b = 6.11, \quad L_b = 0.224, \quad d = 0.04,$$

$$\begin{aligned}
m &= 1.126, L = 0.1, \\
(I_{0x}, I_{0y}, I_{0z}) &= (6e^{-2}, 6e^{-2}, 2e^{-2}), \\
(I_{1x}, I_{1y}, I_{1z}) &= (4e^{-3}, 4e^{-3}, 1e^{-3}), \\
(I_{2x}, I_{2y}, I_{2z}) &= (3.5e^{-2}, 3.5e^{-2}, 7e^{-2})
\end{aligned}$$

とした。また、初期値（離陸時の各変数値）は

$$\begin{aligned}
(x_g(0), y_g(0), z_g(0)) &= (0, 0, 0.2) \\
(e_1(0), e_2(0), e_3(0)) &= (0, 0.01, 0) \\
(\psi_1(0), \psi_2(0), \psi_3(0)) &= (0.01, 0, 0) \\
(\dot{x}_g(0), \dot{y}_g(0), \dot{z}_g(0)) &= (0, 0, 1.5) \\
(\dot{e}_1(0), \dot{e}_2(0), \dot{e}_3(0)) &= (-1, 0, 0) \\
(\dot{\psi}_1(0), \dot{\psi}_2(0), \dot{\psi}_3(0)) &= (0, 0, 0)
\end{aligned}$$

とした。 $\dot{e}_1(0) = -1$ [rad/s] は、離陸時に胴体が Roll 軸の回転角速度を持っていることを意味している。整定時間は $T_v = 2z_g(0)/g = 0.3$ [s]（すなわち離陸時間）とし、遊脚の着地目標値は $\theta_{1d} = -1$ [rad], $\dot{\theta}_{1d} = 3$ [rad/s] とした。

Fig. 3 と Fig. 4 はシミュレーションの初期姿勢と最終姿勢を示している。Fig. 5–Fig. 10 は各変数の時間グラフである。横軸が時間軸（単位は秒）である。

Fig. 8 の上段のグラフは制御入力であるが、胴体を真横に回頭させるために比較的大きな Yaw 軸補償器のトルクを必要としている。これは直感と一致する。下段のグラフは重心周りの角運動量の 3 成分を示している。この中でロール軸周りの実際の角運動量を示す M_1 だけが非零となっている。

Fig. 9 における出力関数 α の収束の速さはゲインによって調整できる。 β については途中で外乱の影響をうけることなしに零に留まっている。また、Fig. 10 より、遊脚の角度と角速度が指定時刻で目標値に整定していることもわかる。特筆すべきは、零ダイナミクス（制御していない遊脚のダイナミクスに相当）が安定な挙動を示しているということである。

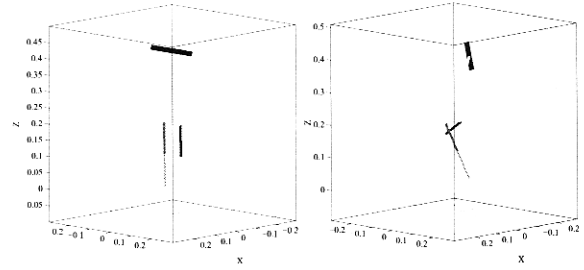


Fig. 3 Initial posture Fig. 4 Final posture

以上により、意図した通りの結果が得られ、提案した制御則の有効性を確認した。しかし、初期値と目標値によっては特異点に遭遇することも確認している。

5. まとめ

本稿では開発中の Yaw 軸補償器を有する 3 次元 2 脚ロボットの走行制御の実現のために、横方向への転倒を回避するための最も基本的な空中姿勢制御則を提案した。単純化したモデルを用いて制御則を導出し、ある初期値の範囲においては目標とする姿勢制御を達成できることをシミュレーションにおいて確認した。しかし、特異点が存在するので、その特定と対策を見いだすことが早急な課題として残されている。

参考文献

- 1) 江村, 黒田, 玄: Yaw 軸補償器を有する直動関節型 3 次元 2 脚ロボットの開発, 第 45 回自動制御連合講演会 (2002)
- 2) M.Raibert: *Legged Robots That Balance*, MIT Press (1985)
- 3) 玄, 江村: 2 脚ロボットの受動走行に向けて, 第 20 回日本ロボット学会学術講演会 (2002)
- 4) S.H.Hyon, T.Emura: Quasi-Periodic Gaits of Passive One-Legged Hopper, Proc. of IEEE IROS, 2625/2630 (2002)

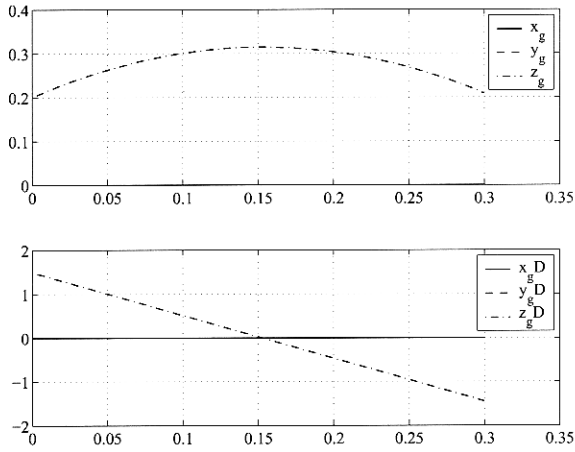


Fig. 5 COM positions and velocities

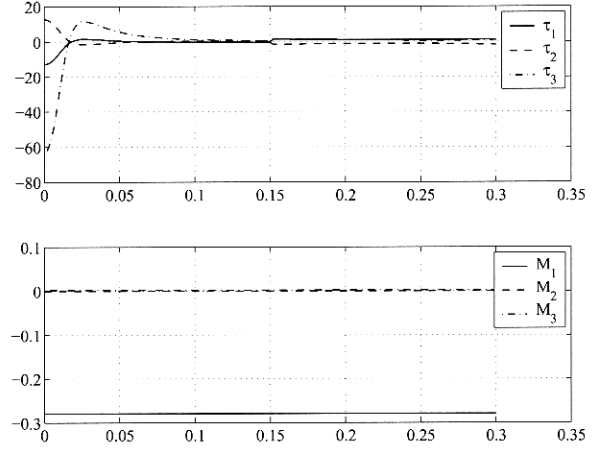


Fig. 8 Input torques and angular momentums

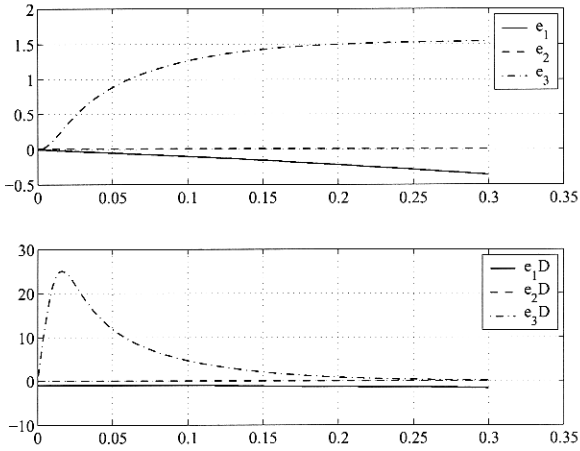


Fig. 6 Attitude and velocities

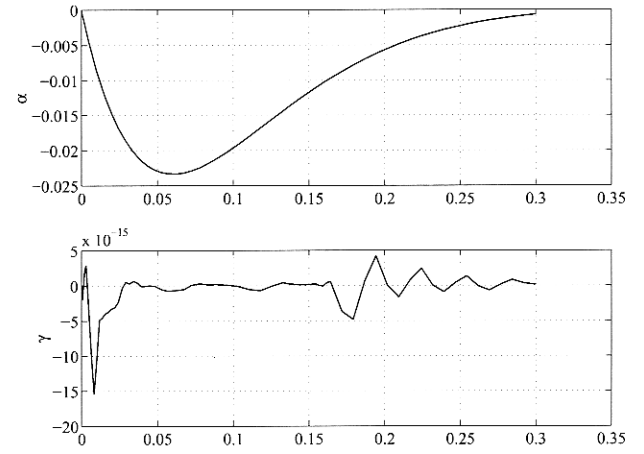


Fig. 9 Output functions

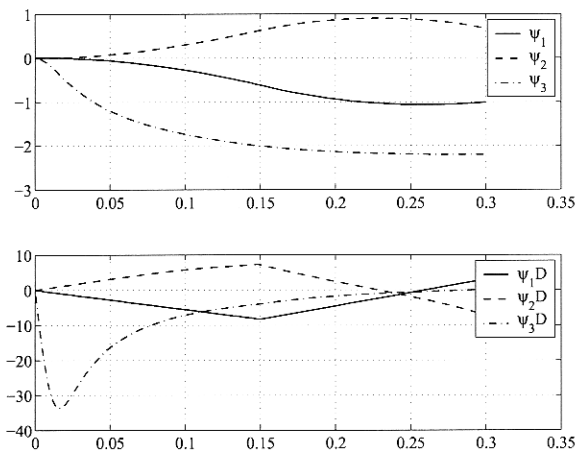


Fig. 7 Joint angles and velocities

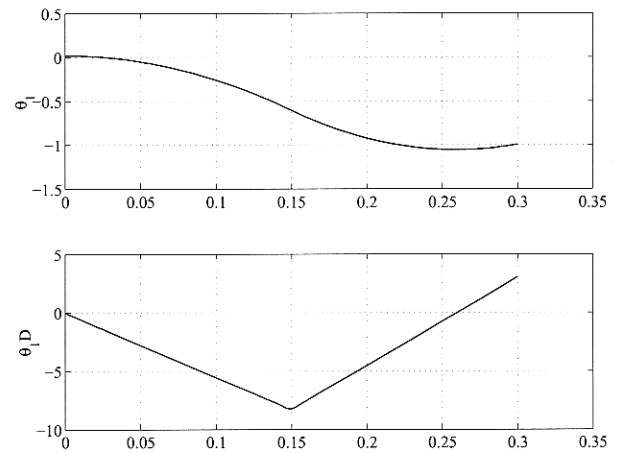


Fig. 10 Swing leg angle and velocity