強化学習を用いてダンパ係数を指定した4脚ロボットのコンプライアンス制御

成 清 辰 生^{*1} 松 本 大 河^{*1} Barkan Ugurlu^{*2} 川 西 通 裕^{*1}

Compliant Locomotion Control for a Quadruped Robot with Damper Coefficients Assigned by Reinforcement Learning

Tatsuo Narikiyo^{*1}, Taiga Matsumoto^{*1}, Barkan Ugurlu^{*2} and Michihiro Kawanishi^{*1}

Since quadruped robots have been considered to be effective on rough terrain, they have been studied in various fields such as rescue and planetary exploration. In the legged locomotion literature, researchers mostly focus on mechatronics hardware and actuation systems of quadruped robots. Nevertheless, such systems may be questionable from the stability point of view, in particular, for dynamic walking and running. Therefore, to develop high locomotion capabilities to adjust to different type of terrains, we propose a new type of compliance control based on reinforcement learning. In this method, compliance control parameters of each joint can be adjusted by external forces acting on the robot feet. Due to this adaptive mechanism, dynamically balanced jumping motion and trotting quadruped locomotion on the rough terrain can be realized. In order to demonstrate the efficiency of the proposed method, jumping and trotting experiments were conducted on our quadruped robot. As a result, we obtained periodic, continuous and repetitive jumping and trotting cycles in which dynamic balance was ensured.

Key Words: Quadruped Robots, Compliant Control, Reinforcement Learning, Actor Critic

1. は じ め に

不整地での走行を目的として、脚型ロボットに対してコンプライアンス制御が用いられてきた。コンプライアンス制御は受動コンプライアンス (passive compliance) と能動コンプライアンス (active compliance) に大別できる。受動コンプライアンスの例としては、Hutter らの StarETH [1], 木村らの鉄犬 II [2] があげられる。これに対し、能動的コンプライアンスでは、Hyon [3] や Ott [4] らの研究がある。これらの研究では、あらかじめ位置制御の PD ゲインを低くすることにより、各関節に仮想的なバネ-ダンパ系の弾性を持たせている。また、位置制御とは分離した力制御を行い、地面から受ける反力から生じる力の誤差を、そのままトルクに変換しロボットへのフィードバック制御を行っている。しかし、この手法では、脚が地面に接していない遊脚時においても、各関節に弾性を持たせていることにより位置制御に誤差が生じてしまう。これに対し Ugurlu らは、力制御と位置制御とを分離しない関節空間のコンプライアンス制御を提案した [5]。提案された手法では、ロボットが着地の際

に衝撃を受けた場合、各関節の仮想的なバネ-ダンパ系により、衝撃による力の誤差を緩和する。一方、ロボットの脚が遊脚時にあり、地面からの反力を受けない場合、位置制御による追従制御を行う。これにより、遊脚時の目標軌道に対する追従性が保持される。また外力が加わった際は、その力の大きさに応じた補正が行われる。この手法は大型の4脚ロボット (HyQ: Istituto Italiano di Tecnologia) と小型4脚ロボット (RoboCat-1: 豊田工業大学) に実装され、その有効性が示された [6]。関節空間でのコンプライアンス制御の関連研究として、Buchli らは不整地での4脚ロボット歩行実験を実現している [7]。Boaventura らも同じ制御手法を油圧駆動の4脚ロボットに用いて、屋外走行運動を実現している [8]。

しかし、このようなコンプライアンス性能の調整には、歩行時の安定性を目視で確認し特定の環境で調整する必要があるが、調整の規範となる確立した調整法は存在しない。そこで本研究では、ダンパ係数の決定に強化学習 [9] を導入する。強化学習を用いてより安定した歩行となるようにダンパ係数を学習する手法を提案し、小型4脚ロボット (RoboCat-1) に適用する。

2. 4脚ロボット (RoboCat-1)

動的で安定な脚移動を研究するため、試作した小型4脚ロボット (RoboCat-1) を用いた。Fig. 1 に実機を示し、またロボットの主な特徴を Table 1 にまとめる。ロボットは各脚におい

原稿受付 2016 年 6 月 23 日

^{*1}豊田工業大学

^{*2}Ozyegin University

^{*1}Toyota Technological Institute

^{*2}Ozyegin University

■ 本論文は有用性 (要素分野) で評価されました。

Fig. 1 Quadruped robot prototype RoboCat-1

Table 1 RoboCat-1 Specifications

Size:	37×25×33 [cm] (L×W×H)
Total Weight:	8.0 [kg]
link0(Hip-Knee):	5.5 [cm]
link1(Hip-Knee):	10 [cm]
link2(Knee-Foot):	13.8 [cm]
Sensors:	2 single axis gyro; 4×1 force sensor
Total DoF:	3 in every leg; 4×3 = 12 DoF

Table 2 AC Servo Motor Specifications

Item/model	FHA-8C-50	FHA-8C-100
Reduction ratio	50	100
Max. momentary torque [N.m]	3.3	4.8
Max. rotational speed [r/min]	120	60
Torque constant [N · m/A]	6.7	14
Moment of inertia [N · m/A]	0.0074	0.029
Detector resolution [p/rev]	400,000	800,000
Mass [kg]	0.40	
Encoder type	Incremental encoder	

Table 3 ForceSensor USL06-H5-500N Specifications

Size:	20×20×5 [mm] (L×W×H)
rating capacity Fx:	± 250 [N]
rating capacity Fy:	± 250 [N]
rating capacity Fz:	± 500 [N]

て股関節 2 軸と膝関節 1 軸を持ち、計 12 関節で構成されている。股関節により y 軸 (Pitch 軸) 周りと x 軸 (Roll 軸) 周りの回転が行えるため、ロボットは平面上を移動できる。各関節には、Harmonic Drive Systems 社の AC サーボアクチュエータ FHA-8C シリーズを用いている。股関節と膝関節には、それぞれ FHA-8C-100, FHA-8C-50 を使用している。これらにより、関節駆動と関節角計測を行う。アクチュエータの仕様を Table 2 に示す。動力部がこのアクチュエータのみのため、ロボットはバネなどの物理的なコンプライアンス要素を一切持たない。ロボットのリンクは、硬化アルミニウム製であり、軽くて十分な耐久性を持つ。

さらにセンサ部としては、胴体部に 1 軸のジャイロセンサを二つ、また床反力を測定するために各足裏に 3 軸力覚センサを取り付けている。ジャイロセンサは、datatec 社の 1 軸角度センサ Y/RP を使用し、胴体の姿勢角度 (Pitch 軸, Roll 軸周り

Fig. 2 (a) Sketch to define the body length (l_{RL}, l_{FH}), (b) Sketch to define the leg lengths (l_0, l_1, l_2), the leg coordinate system ($x-y-z$) and joint angles (q_0, q_1, q_2)

Table 4 Model Parameter

Parameter	Name	Values
l_0	length of hip-knee link (link0)	0.055[m]
l_1	length of hip-knee link (link1)	0.1[m]
l_2	length of knee-foot link (link2)	0.158[m]
m_0	mass of hip-knee link	0.45[kg]
m_1	mass of hip-knee link	0.41[kg]
m_2	mass of knee-foot link	0.12[kg]
l_{FH}	length of front-hind	0.37[m]
l_{RL}	length of right-left	0.25[m]

の角度) を測定する。床反力センサには、テック技販の小型 3 軸力覚センサ (500 [N]) USL06-H5-500N-C を用いた。また、Fig. 2 に 4 脚ロボットの座標モデルを示す。(a) は胴体のサイズ、(b) は $x-z$ 平面、 $z-y$ における 1 脚モデル (3 リンク) を示す。Fig. 2(b) に示すように、鉛直方向を z 軸方向、水平方向を x 軸方向と y 軸方向とする。股関節基準点にとり、その点から、リンク 0, リンク 1, リンク 2 と呼ぶ。股関節と膝関節の角度をそれぞれ q_0, q_1, q_2 とする。なお角度は時計回りを正にとる。また、各リンクの長さを l_0, l_1, l_2 とし、モデルの物理パラメータを Table 4 にまとめる。

実験システムの全体の構成を Fig. 3 に示す。実験を行うにあたり、MATLAB のコンポーネントの一つである Real-Time Workshop を用いたリアルタイム制御系を構成した。Real-Time Workshop を用いたシステム構成では、実際に制御を行う xPC Target と実験対象である 4 脚ロボットのほかに、xPC Target

Fig. 3 Computation frame of RoboCat-1

と通信を行う Main PC を用意する。Main PC は xPC Target で実行するプログラムの実行ファイルをコンパイルし、LAN 経由で実行ファイルを送信する役割を持つ。また、実験後にデータを収集する際に、xPC Target からデータを受け取る役割を持つ。一方 xPC Target は実験装置と信号をやり取りするために、様々なインタフェースを備えている。D/A 部分では、ロボットの各関節に取り付けられたモータを駆動するための電圧を出力する部分となっている。出力先に、Harmonic Drive Systems 社の AC サーボドライバである HA-800A シリーズに接続し、トルク制御モードによってモータを制御している。

3. 目標軌道設計

4 脚ロボットの目標軌道を設計する際に、運動学および逆運動学計算が必要となる。ここでは、Ugurlu らによって開発された設計法を簡単に紹介する。詳細は文献 [5] [6] を参照のこと。1 脚における順運動学計算は、各リンクの長さ (l_0, l_1, l_2) と関節角度 (q_0, q_1, q_2) から、Fig. 4 に示す足先の位置 $\mathbf{r}_{hf} = [x_f \ y_f \ z_f]^T$ を求める問題となる。

股関節の座標を基準点 $\text{joint}_0: [x_0 \ y_0 \ z_0] = [0 \ 0 \ 0]^T$, $\text{joint}_1: [x_1 \ y_1 \ z_1] = [-l_0 \ \sin q_0 \ \cos q_0]^T$ とすると、膝関節の座標 joint_2 は、 $[x_2 \ y_2 \ z_2] = [-l_1 \sin q_1 \ 0 \ -l_1 \cos q_1]^T$ と書ける。これより、足先の座標は以下で表される。

$$\begin{bmatrix} x_f \\ y_f \\ z_f \end{bmatrix} = \begin{bmatrix} -l_0 + l_1 \sin q_1 + l_2 \sin(q_1 + q_2) \\ (l_1 \cos q_1 + l_2 \cos(q_1 + q_2)) \sin q_0 \\ -(l_1 \cos q_1 + l_2 \cos(q_1 + q_2)) \cos q_0 \end{bmatrix} \quad (1)$$

1 脚における逆運動学計算は、足先の位置 $\mathbf{r}_{hf} = [x_f \ y_f \ z_f]^T$ から、関節角度 (q_0, q_1, q_2) を求める計算を行う。計算を簡単にするために、以下に定義する補助変数を導入する。

$$\begin{cases} x_{cons} = -x_f, y_{cons} = y_f, z_{cons} = -z_f \\ A = l_1 + l_2 \cos q_2 \\ B = l_2 \sin q_2 \end{cases} \quad (2)$$

股関節ロール軸 q_0 については、 y_f, z_f から求める。

$$q_0 = \text{atan2}(-y_f, z_f) \quad (3)$$

次に、式 (2) の補足変数を用いて各関節角度を求めていく。まず、式 (1) より、

$$x_{cons}^2 + y_{cons}^2 + z_{cons}^2 = l_1^2 + l_2^2 + 2l_1 l_2 \cos q_2$$

と表される。これより、膝関節 q_2 は、

Fig. 4 One leg model for Kinematics.

Fig. 5 Coordinate frames on RoboCat-1. Legs are numbered from 1 to 4

$$\begin{aligned} \cos q_2 &= \frac{x_{cons}^2 + y_{cons}^2 + z_{cons}^2 - l_1^2 + l_2^2}{2l_1 l_2} \\ \sin q_2 &= \sqrt{1 - \cos^2 q_2} \\ q_2 &= \text{atan2}(\sin q_2, \cos q_2) \end{aligned} \quad (4)$$

となる。また股関節 q_1 は、

$$\begin{aligned} \sin q_1 &= \frac{A x_{cons} - B z_{cons}}{A^2 + B^2} \\ \cos q_1 &= \sqrt{1 - \sin^2 q_1} \\ q_1 &= \text{atan2}(\sin q_1, \cos q_1) \end{aligned} \quad (5)$$

と計算できる。

以上より、与えられた足先の位置 $\mathbf{r}_{hf} = [x_f \ y_f \ z_f]^T$ から、各関節角度 (q_0, q_1, q_2) を求めることができる。

一方、各関節の目標角度を求めるためには、各脚において股関節から足先への位置ベクトルが必要である。例えば、2 番めの脚 (右前脚) におけるこの位置ベクトルは Fig. 5 中のベクトル $\mathbf{r}_{hf} = [x_f \ y_f \ z_f]^T$ で示される。Fig. 5 に示す位置ベクトルを用いて、ベクトル $\mathbf{r}_{hf}$ は以下の式で求められる。

$$\mathbf{r}_{hf} = \mathbf{R}_t^T (\mathbf{r}_f - \mathbf{r}_c) - \mathbf{r}_{ch} \quad (6)$$

$$\mathbf{R}_t = \mathbf{R}_\theta \mathbf{R}_\phi = \begin{bmatrix} \cos \theta & \sin \theta \sin \phi & \sin \theta \cos \phi \\ 0 & \cos \phi & -\sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \quad (7)$$

ここで, $\mathbf{r}_c = [x_{com} \ y_{com} \ z_{com}]^T$ はワールド座標系からみた重心位置 (CoM) のベクトル, R_t は胴体での姿勢角を示す回転行列であり, Pitch 軸 (θ) と Roll 軸 (ϕ) の二つの軸まわりの回転として定義する. $\mathbf{r}_{ch} = [x_{ch} \ y_{ch} \ z_{ch}]^T$ は重心 (CoM) から股関節への位置ベクトルである. 得られた $\mathbf{r}_{hf} = [x_f \ y_f \ z_f]^T$ を用いて, 1 脚 (3 リンク) における逆運動学計算を行うことで, 各関節角度が解析的に求められる. 以上の処理は, Fig. 6 中の 'Vectorial Leg Length Calculation' と 'Single Leg Inverse Kinematics' に相当する.

このように, 各関節の目標角度生成には定義された 12 の入力が必要である. 重心位置 ($x_{com}; y_{com}; z_{com}$), 各足先の位置 ($x_f; y_f; z_f$) $\times 4$ と胴体の姿勢角 (θ, ϕ, φ) の計 18 である. それに対し, 実機全体の自由度は 12 である. これはシステムの問題のように思えるが, トロット歩行の場合, 対角線上の脚を同時に動かすため, 実際には 12 のうち六つの入力が必要としないのである. 例えば, トロット歩行の軌道において支持脚相を考える. 4 脚のうち, 2 脚は地面に接地しているためそれら脚の足先位置を指定することは特に必要でない. 以上のことから, 2 脚分の足先位置である六つの入力は必要なくなり, 12

Fig. 6 Input part of control scheme. j stands for leg number ($j = 1, 2, 3, 4$); whereas i is joint number for the j^{th} leg ($i = 1$: Hip, $i = 2$: Knee)

自由度に対し 12 入力で計算可能となる.

4. コンプライアンス制御

制御系の全体構成を Fig. 7, コンプライアンス制御部を Fig. 8 に示す. 関節での位置制御手法である PD 制御に加え, ロボットの着地時に受ける衝撃力など環境から受ける外力を緩和するためのコンプライアンス制御を用いている. コンプライアンス制御を行う目的は, 環境から受ける外力の影響を小さくするためのフィードバックを生成することである. 具体的には, 各関節にバネ-ダンパ系の特性を持たせ, 目標軌道から得られる目標の力と, 足先に取り付けられた力覚センサの測定値との誤差を小さくするような各関節の角度を求める. 力の誤差を計算するため, まずは各脚の目標値 ($F_{refj} = [F_{xrefj} \ F_{zrefj}]^T$, $j = 1, \dots, 4$) を求める. このため, Fig. 9 に示す幾何情報を利用する. なお, 添え字の数字は各脚の番号と一致するものとする. Fig. 9(a) は, ロボットを右から見た図である. 各脚は地面に対し点接地となり, 床面に対する脚の接地角度は各関節角度から計算できる. 同様に, 左側の脚においても同様である. また Fig. 9(b) は, 重心の床面投影点から各足先へのベクトルを示した図である. これらの図より, 目標力の各成分と, 重心の床面投影点から各足先へのベクトルを求めることができる.

次に力覚センサの測定値と関節角度から実際の力の各成分 F_{xj} , F_{zj} を求める. Fig. 9 より, 力覚センサから得られる力 F_{xj} , F_{zj} と目標値から, 力の誤差 $\mathbf{F}_{errj}$ とその誤差に相当するトルク $\boldsymbol{\tau}_{cj}$ は以下のように求められる.

$$\mathbf{F}_{errj} = \begin{bmatrix} F_{xerrj} \\ F_{zerrj} \end{bmatrix} = \begin{bmatrix} F_{xrefj} - F_{xj} \\ F_{zrefj} - F_{zj} \end{bmatrix} \quad (8)$$

$$\boldsymbol{\tau}_{cj} = \begin{bmatrix} \tau_{c1j} \\ \tau_{c2j} \end{bmatrix} = J_j^T \begin{bmatrix} F_{xerrj} \\ F_{zerrj} \end{bmatrix} \quad (9)$$

式 (9) にて, J_j^T は 1 脚 (2 リンク) におけるヤコビ行列である. 本来, 各関節が持つ固有の静止摩擦や粘性摩擦により, 式 (9) に誤差が生じ, 十分な精度を示さなくなる. しかし, 本研

Fig. 7 General motion control frame. j denotes leg number ($j = 1, 2, 3, 4$). i refers joint number for the j^{th} leg ($i = 1$: Hip, $i = 2$: Knee)

Fig. 8 Active compliance control scheme. Output of this scheme (q_{c1}, q_{c2}) is feedback to respective PD servo controllers as illustrated in Fig. 7

Fig. 9 a) Resolution of GRF(Ground reaction force) on right side. b) Position vectors from CoM to each foot

究では摩擦補償を行うことで、各関節が持つ固有の摩擦を打ち消すことができる。そして、コンプライアンス制御の出力値である角度 $[q_{c1j}, q_{c2j}]^T$ は以下のように算出する。

$$\begin{bmatrix} q_{c1j} \\ q_{c2j} \end{bmatrix} = \begin{bmatrix} \frac{1}{k_{1j}+s} & 0 \\ 0 & \frac{1}{k_{2j}+s} \end{bmatrix} \begin{bmatrix} \tau_{c1j} \\ \tau_{c2j} \end{bmatrix} \quad (10)$$

式 (10) にて、 (k_{1j}, b_{1j}) と (k_{2j}, b_{2j}) は j 番の股関節と膝関節に対するバネ係数とダンパ係数である。また、 s はラプラス演算子である。

コンプライアンス制御の主な目的は、各脚に力の誤差を補償するために必要な、補助的な関節運動を生成することである。言い換えれば、この制御手法は間接的な力のフィードバックである。得られた角度 $[q_{c1j}, q_{c2j}]^T$ を PD 制御の入力として与えることで、コンプライアンス制御をロボットに実装することができる。

5. Actor-Critic 法の実装

5.1 Actor-Critic 法

本研究で用いた Actor-Critic 法の概略を記す [10]。確率的政策 π は、次式 (11) で与えられる。

$$\pi(a_t, s_t, w_i) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(a_t - \mu)^2}{2\sigma^2}\right) \quad (11)$$

$$\mu = w_1 s_t, \sigma = \frac{1}{1 + \exp(-w_2)} \quad (12)$$

$\pi(a_t, s_t, W_t)$ はパラメータ $w_i \in W$ により行動の選択確率が変化する。適正度は、この政策パラメータ w_i で確率的政策

$\pi(a_t, s_t, W_t)$ を偏微分することで得られる。ガウス関数を用いる利点として、log を取ったときに単純な形となり微分が行いやすい。また、ガウス関数の場合は、 μ と σ の二つのパラメータの内部変数として政策パラメータ w_i を式 (11), (12) で定義する。状態 s_t を得たときの行動 a_t は、正規分布を用いた擬似乱数 p で求めることになる。

$$a_t = \mu + \sigma p \quad (13)$$

$$e_i(t) = \frac{\partial}{\partial w_i} \ln \pi(a_t, s_t, W_t) \quad (14)$$

次に、適正度 $e_i(t)$ を用いて、適切な w_i の学習を行う。学習には、過去の適正度を用いることで現時点の行動だけでなく近い過去にとった行動も反映させる。適正度の履歴 $D_i(t)$ は、以下のように表す。

$$D_i(t) = e_i(t) + \beta D_i(t-1) \quad (15)$$

β は、過去の情報に対する重みであり、適正度の履歴を更新していくパラメータである。政策パラメータの更新は次の手順で行われる。

政策関数の評価には、価値関数によって求められた TD 誤差によって行われる。

$$\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t) \quad (16)$$

この TD 誤差の値が正の場合、 a_t を選択する将来の傾向を強めるように確率分布を更新し、逆に値が負の場合には、その傾向を弱めるように更新を行う。更新量は、適正度の履歴も用いて以下のように与えられる。

$$dw_i(t) = \delta D_i(t-1) \quad (17)$$

よって、更新式は次式で与えられる。

$$w_i(t+1) = w_i(t) + \alpha dw_i(t) \quad (18)$$

5.2 状態・行動・報酬

コンプライアンス制御は、Fig. 8 に示すように力覚センサから得られた情報と目標値となる力の差をヤコビ行列を用いることで各関節に必要なトルクへと変換する。その後、バネとダンパのパラメータを用いてトルク値を補正角度へ変換する。このときの、ダンパのパラメータを強化学習によって決定する。つまり、Actor によってダンパ係数 b_2 を確率的政策により連続値行動として選択する。この際に重要となる報酬の決定法について示す。

まず状態は、胴体に搭載したジャイロセンサによって Roll 軸、Pitch 軸の角度を用いて計算する。胴体中心の垂直方向の軸と z 軸のなす角を求め、角速度へと計算する。さらに角速度は、 $1 [\text{m/s}^2]$ ごとに離散化され価値関数の更新に用いられる。すなわち状態は、胴体が回転したときの角速度を用いており、回転方向は考慮しない。このようにする理由として、胴体がどのような速度で変位しているかが振動へ影響していると考えられるからである。 θ と ϕ の回転をジャイロセンサで取得し、胴体の回転を求める。以下に、回転の様子を図で示す。次に行動について

Fig. 10 State model scheme

Fig. 11 Compliance model scheme

説明する。行動は、ダンパ係数に対応させた。Fig. 11 に示すように、股、膝関節にバネとダンパが仮想的に実装されていると考えることができる。この膝関節のダンパ係数を行動とした。ダンパを学習として選択することで、発生した揺れの抑制を環境に合わせて行うように動作する。

最後に報酬であるが、胴体の重心周りの角運動量を求める。胴体の揺れは、各脚の曲げ伸ばしによって生じる。まず始めに胴体の重心周りの角運動量の時間微分は、オイラーの運動方程式を用いて以下の様に表される。

$$\dot{L}_x = I_{xx}\dot{\omega}_x - (I_{yy} - I_{zz})\omega_y\omega_z \quad (19)$$

$$\dot{L}_y = I_{yy}\dot{\omega}_y - (I_{zz} - I_{xx})\omega_z\omega_x \quad (20)$$

$$\dot{L}_z = I_{zz}\dot{\omega}_z - (I_{xx} - I_{yy})\omega_x\omega_y \quad (21)$$

式 (19)～(21) にて、 I_{xx}, I_{yy}, I_{zz} は胴体の重心周りの慣性モーメント、 $\omega_x, \omega_y, \omega_z$ は重心周りの角速度を示し、また $\dot{L}_x, \dot{L}_y, \dot{L}_z$ は胴体の重心周りの角運動量の時間微分である。式 (19)～(21) の値を求めるためには、Pitch 軸 (θ)、Roll 軸 (ϕ) 周りの角速度と角加速度を得る必要がある。そのためにまず胴体の回転行列 R_t を、Roll 軸の回転行列 R_ϕ と Pitch 軸の回転行列 R_θ との積によって求める。

$$R_t = R_\theta R_\phi = \begin{bmatrix} \cos \theta & \sin \theta \sin \phi & \sin \theta \cos \phi \\ 0 & \cos \phi & -\sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \quad (22)$$

次に、角速度ベクトルを求める。角速度ベクトルの成分は、歪称行列 (skew-symmetric matrix) $\omega^\dagger$ を用いて表される。

$$\begin{aligned} \omega^\dagger &= \dot{R}_t R_t^T = \begin{bmatrix} 0 & \dot{\phi} \sin \theta & \dot{\theta} \\ -\dot{\phi} \sin \theta & 0 & -\dot{\phi} \cos \theta \\ -\dot{\theta} & \dot{\phi} \cos \theta & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \end{aligned} \quad (23)$$

式 (23) を用いることで、角速度ベクトルは以下のように得られる。さらに、角速度ベクトルを微分することにより、角加速度ベクトルも得ることができる。

$$\begin{aligned} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} &= \begin{bmatrix} \dot{\phi} \cos \theta \\ \dot{\theta} \\ -\dot{\phi} \sin \theta \end{bmatrix} \\ \begin{bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix} &= \begin{bmatrix} \ddot{\phi} \cos \theta - \dot{\theta} \dot{\phi} \sin \theta \\ \ddot{\theta} \\ -\ddot{\phi} \sin \theta - \dot{\theta} \dot{\phi} \cos \theta \end{bmatrix} \end{aligned} \quad (24)$$

式 (24) を式 (19), (20) に代入して、重心の Roll 軸と Pitch 軸周りの角運動量の微分 ($\dot{L}_x, \dot{L}_y$) が求められる。ここで、 $\dot{L}_z$ はロボットが構造的に Yaw 軸周りの運動を生成できないため考慮しない。

$$\begin{aligned} \dot{L}_x &= I_{xx}(\ddot{\phi} \cos \theta - \dot{\theta} \dot{\phi} \sin \theta) \\ &\quad + (I_{yy} - I_{zz})(\dot{\theta} \dot{\phi} \sin \theta) \end{aligned} \quad (25)$$

$$\dot{L}_y = I_{yy}\ddot{\theta} + \frac{1}{2}(I_{zz} - I_{xx})(\dot{\phi}^2 \sin 2\theta) \quad (26)$$

式 (25), (26) より、与えられたロボットの姿勢角 (θ, ϕ) とそれらの微分値から胴体の重心周りの角運動量の微分が得られた。その後、積分することで角運動量を得ている。さらに、学習ステップでの角運動量の増加分を報酬とするため以下のように計算する。

$$\text{reward} = -(Lr_t - Lr_{t-1}) - (Lp_t - Lp_{t-1}) \quad (27)$$

Lr, Lp は、Roll 軸、Pitch 軸周りの角運動量である。

学習のサイクルとしては、始めに状態 s_t として胴体中心周りの角速度を観測し、状態 s_t において確率的政策によってダンパ b_2 を決定する。このダンパ b_2 で行動後、状態 s_{t+1} と報酬 r_{t+1} を環境より得る。報酬 r_t は、胴体を一つの剛体としたときの Roll 軸、Pitch 軸の回転に対する角運動量の時間当たりの増加量とし、この値が小さくなるように学習している。次に、Critic の確率的政策を TD 誤差および適正度の履歴によって更新していく。この一連の流れを繰り返すことで環境に適したダンパ係数 b_2 を選択していく。

これらを以下にまとめる。

- (1) 状態は、胴体の回転角速度とする。回転方向は考慮しない。 θ と ϕ の回転をジャイロセンサで取得し、胴体の回転を求める。
- (2) 行動は膝関節のダンパ係数とする。ダンパを学習として選択することで、発生した揺れの抑制を環境に合わせて行うように動作する。
- (3) 報酬 (Reward) を式 (27) で定義する。

6. 実験結果

6.1 コンプライアンス係数設定と角運動量制御

コンプライアンス係数はバネ係数とダンパ係数であり、それぞれ4脚の股関節と膝関節の制御部と胴体のピッチ角およびロール角の角運動量制御部に組み込まれている。したがって全部で20個のコンプライアンス係数が存在し、歩容状態によりそれぞれが独自の役割を果たしている。しかし、これらの係数の効果を各係数ごとに検討することは不可能に近く、実験条件の設定も難しい。そこで、各脚のバネ係数とダンパ係数はすべて等しく選び、コンプライアンス係数を固定し様々なパターンで係数設定値を探索した結果、バネ係数の値はダンパ係数よりかなり大きくとらないと安定な走行・跳躍ができないことが分かった。これは、バネ係数を小さくすることは脚の剛性を低下させることになり、脚の接地時の反力を低下させる反面、胴体部の揺動を大きくするためと考えられる。このため、初期係数の設定では、胴体部の振動、角速度を抑制することを目標として、バネ係数をある程度大きく設定し、そのバネ係数の下でダンパ係数を調整する実験を繰り返し行い、Table 5 に示す値を最良の初期係数とした。

k_1 と b_1 は股関節のバネ係数およびダンパ係数、 k_2 と b_2 は膝関節のバネ係数およびダンパ係数である。一方、 k_p 、 b_p は胴体の Pitch 軸のバネ係数とダンパ係数、 k_r 、 b_r は胴体の Roll 軸のバネ係数とダンパ係数であり、以下の角運動量制御で定義される。式 (25)、(26) より与えられたロボットの姿勢角 (θ , ϕ) とそれらの微分値から胴体の重心周りの角運動量の微分が得られた。胴体の姿勢角の目標値 (θ_{ref} , ϕ_{ref}) を式 (25)、(26) に代入し、重心周りのトルクの目標値が得られる。

$$\begin{aligned} \dot{L}_{xref} = & I_{xx}(\ddot{\phi}_{ref} \cos \theta_{ref} - \dot{\theta}_{ref} \dot{\phi}_{ref} \sin \theta_{ref}) \\ & + (I_{yy} - I_{zz})(\dot{\theta}_{ref} \dot{\phi}_{ref} \sin \theta_{ref}) \end{aligned} \quad (28)$$

$$\begin{aligned} \dot{L}_{yref} = & I_{yy} \ddot{\theta}_{ref} \\ & + \frac{1}{2}(I_{zz} - I_{xx})(\dot{\phi}_{ref}^2 \sin 2\theta_{ref}) \end{aligned} \quad (29)$$

そして、ジャイロセンサを用いて測定された実際の角度より、実際の重心周りのトルク値が算出できる。目標値と区別するために添え字を *mes* とする

$$\begin{aligned} \dot{L}_{xmes} = & I_{xx}(\ddot{\phi}_{mes} \cos \theta_{mes} - \dot{\theta}_{mes} \dot{\phi}_{mes} \sin \theta_{mes}) \\ & + (I_{yy} - I_{zz})(\dot{\theta}_{mes} \dot{\phi}_{mes} \sin \theta_{mes}) \end{aligned} \quad (30)$$

Table 5 Compliance coefficients

Parameter	Name	Values
k_1	spring coefficient of hip joint	60
b_1	damper coefficient of hip joint	6
k_2	spring coefficient of knee joint	72
b_2	damper coefficient of knee joint	12
k_p	spring coefficient of pitch angle	15
b_p	damper coefficient of pitch angle	2.5
k_r	spring coefficient of roll angle	15
b_r	damper coefficient of roll angle	2.5

$$\dot{L}_{ymes} = I_{yy} \ddot{\theta}_{mes} + \frac{1}{2}(I_{zz} - I_{xx})(\dot{\phi}_{mes}^2 \sin 2\theta_{mes}) \quad (31)$$

目標値から実際の重心周りのトルク値を引くことにより、Roll, Pitch 軸に関する重心周りのトルク誤差 $\dot{L}_{xerr}$ と $\dot{L}_{yerr}$ が得られる。

$$\dot{L}_{xerr} = \dot{L}_{xref} - \dot{L}_{xmes} \quad (32)$$

$$\dot{L}_{zerr} = \dot{L}_{zref} - \dot{L}_{zmes} \quad (33)$$

トルク誤差 $\dot{L}_{xerr}$ と $\dot{L}_{yerr}$ に相当するバネ-ダンパ系のシステムを考える。トルク誤差を補正するための補正姿勢角度 θ_c , ϕ_c を以下のように定義する [5]。

$$\theta_c = \frac{\dot{L}_{yerr}}{k_p + sb_p}; \quad \phi_c = \frac{\dot{L}_{xerr}}{k_r + sb_r} \quad (34)$$

以上の計算により、胴体の回転運動の補正角度 θ_c , ϕ_c が得られた。これらを Fig. 7 に示すように、目標軌道へフィードバックすることで角運動量制御を実装することができる。Yaw 角 ψ は目標軌道が常に 0 であり、Yaw 角運動は起きない前提で設計しているため、今回の実験では実装していないが、Yaw 角運動の補償も同様に行うことができる。この場合は、Yaw 軸周りのトルク誤差 $\dot{L}_{zerr}$ を計算し

$$\psi_c = \frac{\dot{L}_{zerr}}{k_y + sb_y} \quad (35)$$

をフィードバックする。

6.2 コンプライアンス係数の学習

実験環境は、砂利道を想定して石を並べた不整地と木片を敷き詰めた不整地を用いる。実験時の歩容は、トロット歩容である。トロットは走行運動であるが、研究目的である振動抑制は、遊脚接地時についての対応なので進行方向への動作をせず、z 軸方向への脚上げのみとしている。ここで、トロットの足踏み運動は、トロット歩行と同様に軸足を切り替えながら動作を繰り返すものとし、足踏み回数を軸足の切替回数とし、1 実験で 40 回とした。このときの、2 脚支持相と 4 脚支持相の時間をそれぞれ $t_s = 0.2$ [s]、 $t_d = 0.1$ [s] とした。重心の x 軸方向軌道は 0 [m] から移動させず、z 軸方向軌道は定数値 0.2 [m] とし、遊脚時の脚上げの高さを 0.03 [m] と設定した。設定後は、制作した不整地上に RoboCat-1 を降ろしトロット運動させる。Fig. 12 と Fig. 13 に、二つの不整地を想定した環境を示す。

学習は Table 6 に示す Actor Critic のパラメータを用いて行った。設定すべきコンプライアンス係数は 20 個であるが、こ

Fig. 12 Environment of gravel

Fig. 13 Environment of wood waste

Table 6 Parameters of Actor Critic

Parameter	Name	Values
α	step size parameter of Actor	0.1
α_{TD}	step size parameter of TD(0)	0.2
β	step size parameter of eligibility trace	0.9
γ	discount rate	0.9

Fig. 14 Weight of damper coefficient of knee joint w_1 severely learned by actor-critic

Fig. 15 Damper coefficient of knee joint severely learned by actor-critic

これらの係数を同時に学習することは不可能であるため、各脚のコンプライアンス係数は同じとして、膝関節ダンパ係数とバネ係数を学習する。Actor Critic の学習周期は 100 [ms] として実装した。

Fig. 14 に、膝関節のダンパ係数の政策パラメータ w_1 のみを学習した結果を示す。このとき、ダンパ係数以外のコンプライアンス係数は Table 5 の値を用いた。22 秒から制御・学習を開始して、約 10 秒後には、0.84~0.86 の間に収束し、学習が完了したことが確認できる。コンプライアンス係数は、式 (11)、(12) より政策パラメータと状態（胴体部の角速度）の積で与えられる。このときのダンパ係数の値を Fig. 15 に示す。この学習結果より提案手法では、学習したコンプライアンス係数がチャタリング動作をすることが分かる。これに対して、Fig. 16 および Fig. 17 に膝関節のダンパ係数の政策パラメータ w_1 とバネ係数の政策パラメータ w_3 を同時に学習した学習結果を示す。

Fig. 16 Weight of damper coefficient of knee joint w_1 simultaneously learned with w_3 by actor-critic

Fig. 17 Weight of spring coefficient of knee joint w_3 simultaneously learned with w_1 by actor-critic

ほかのコンプライアンス係数は Table 5 の値を用いた。Fig. 16 に見るように、 w_1 の学習はダンパ係数の政策パラメータ w_1 のみを学習した Fig. 14 と同じような経過をたどり、収束へ向かう様子が見える。一方、Fig. 17 に示すバネ係数の政策パラメータ w_3 は学習が進むと急速に大きな値となり、学習が収束しない。様々な試行実験より、ダンパ係数を固定し、バネ係数のみを学習した場合も w_3 が増大する学習結果となり、学習が収束しないことが分かっている。また、バネ係数は Table 5 に示すようにダンパ係数よりかなり大きな値をとる傾向にあることも分かっている。これらの実験結果は、バネ係数が政策パラメータと状態の積によって決まることから、大きな係数変動が起きると、目標軌道に対する補正量が結果的に高ゲインのチャタリングとして目標軌道にフィードバックされることを示している。この高ゲインのチャタリングによる目標軌道の補正のため、胴体部にさらなる振動が励起され、学習が収束しなくなると考えられる。一方、ダンパ係数のみの学習の場合、チャタリングのゲインが低いため、このような問題は生じない。

ダンパ係数の学習値が Fig. 15 によって与えられたときの実験結果を Fig. 18 に示す。実験結果は石を並べた不整地上での 4 歩分の z 軸方向の加速度を示している。実線が学習ありの結果で、破線が学習なしの結果を示す。学習しないコンプライアンス係数は Table 5 の値を用いた。z 方向の加速度の結果を学習なしと学習ありで比較すると、学習なしの場合に比べて学習を行った場合では、P-P 値が約 50% 低減した。また、目視での胴体部の振動については、学習なしの場合は、遊脚接地時に胴体の様子が引っかけりのある動きに対して、学習ありの場合は、遊脚接地時にも滑らかな動きを維持していた。

Fig. 19 には、TD 誤差を示した。TD 誤差は、運動初期に大きな値を取り政策パラメータの更新値を示す。さらに、この値

Fig. 18 Experimental results on gravel

Fig. 19 Experimental results of TD error

Fig. 20 Experimental results on wood waste

は小さくなっていくことによって学習が収束していることを意味している。Fig. 19の結果を見ると、運動開始から約5秒後まで大きく変化しているが、その後小さい変化へ変わり、学習値が収束し随時変化量を更新し最適な学習値へ向かうように値を求めている。その結果、先ほど示したFig. 14より、政策パラメータ w_1 が収束していることが分かる。

二つめの環境として、Fig. 13に示す木材を並べた不整地を新たに追加し、同様の実験を行った。実験条件は、砂利を敷き詰めた場合の一つめの環境と同じである。Fig. 20に、このときの実験結果を示す。Fig. 20は、z方向の加速度の結果を学習なしと学習ありで比較している。実線は学習ありの場合、破線は学習なしの場合の結果である。学習なしの場合に比べて学習を行ったものでは、P-P値が約33%低減した。

以上、異なる二つの環境において、提案するコンプライアンス制御手法のダンパ係数の調整に文献[10]のActor-Critic法を用いて、4脚ロボットの制御実験を行い、提案手法の有用性を示すことができた。

7. お わ り に

本研究では、Actor-Critic法を適用することにより、各関節

の仮想的なバネ-ダンパ系のダンパ係数を調整するコンプライアンス制御手法を提案した。この手法の有効性の検証として、学習ありの場合と学習なしの場合を比較し、二つの異なる環境を用意して実験を行った。まず、砂利道上にて学習ありと学習なしで比較し、z軸方向の加速度のP-P値が50%低減していることを確認した。これにより、提案手法の有効性を示した。また、木材で模した瓦礫を用いて同様の実験を行った。その結果、z軸方向の加速度のP-P値が33%低減した。このように、それぞれの環境下で固定係数に比較して良好なパフォーマンスを得た。

本研究における実験結果は、明確に異なる環境においては、おのおののダンパ係数を学習可能であることを示している。しかしながら、歩行・走行スピードが速くなる条件下においては、胴体の回転はもちろん多自由度系としての脚の非線形性や脚間の干渉などの影響が大きくなるため、本制御手法を直接適用することは難しい。すなわち、非線形性および干渉の影響を補償する制御系の開発が必要となる。また、本研究では、膝関節のコンプライアンス係数であるダンパ係数を学習し、実験において良好な結果を得たが、股関節およびバネ係数やバネ係数を含めたすべてのコンプライアンス係数を学習した場合、学習が収束しなかった。これらの問題の検討は今後の課題としたい。

謝 辞 本研究は、私立大学戦略的研究基盤形成支援事業の助成を受けて実施された。記して謝意を表す。

参 考 文 献

- [1] M. Hutter, C. Gehring, M. Bloesch, M.A. Hoepflinger, C.D. Remy and R. Siegwart: "A compliant quadrupedal robot for fast, efficient and versatile locomotion," Int. Conf. on Climbing and Walking Robots, pp.1-8, 2012.
- [2] H. Kimura, Y. Fukuoka and A.H. Cohen: "Adaptive dynamic walking of a quadruped robot on natural ground based on biological concepts," IEEE Trans. on Robotics, vol.26, no.5, pp.475-490, 2007.
- [3] S.H. Hyon: "Compliant terrain adaptation for biped humanoids without measuring ground surface and contact forces," IEEE Trans. on Robotics, vol.25, no.1, pp.677-688, 2009.
- [4] C. Ott, M.A. Roa and G. Hirzinger: "Posture and balance control for biped robots based on contact force optimization," IEEE Conf. on Humanoid Robotics, pp.26-32, 2011.
- [5] B. Ugurlu, K. Kotaka and T. Narikiyo: "Actively-Compliant Locomotion Control on Rough Terrain: Cyclic Jumping and Trotting Experiments on a Stiff-by-Nature Quadruped," IEEE Int. Conf. on Robotics and Automation, pp.3298-3305, 2013.
- [6] B. Ugurlu, et al: "Pattern generation and compliant feedback control for quadrupedal dynamic trot-walking locomotion: experiments on RoboCat-1 and HyQ," Autonomous Robotics, vol.38, issue 4, pp.415-437, 2015.
- [7] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry and S. Schaal: "Learning, planning and control for quadruped locomotion over challenging terrain," Int. Journal of Robotics Research, vol.30, no.2, pp.236-258, 2011.
- [8] T. Boaventura, C. Semini, J. Buchli, M. Frigero, M. Focchi and D.G. Caldwell: "Dynamic torque control of a hydraulic quadruped robot," IEEE Conf. on Robotics and Automation, pp.1889-1894, 2012.
- [9] R.S. Sutton and A.G. Barto: Reinforcement Learning: An Introduction. MIT Press, 1998.
- [10] 木村, 小林: "Actor に適正度の履歴を用いた Actor-Critic アルゴリズム 不完全な Value-Function のもとでの強化学習", 人工知能学会誌, vol.15, no.2, pp.267-285, 2000.

成清辰生 (Tatsuo Narikiyo)

1978 年名古屋大学応用物理学科卒業。1983 年名古屋大学大学院情報工学専攻単位取得退学。同年通産省工業技術院九州工業技術試験所入所。1990 年豊田工業大学助教授、1998 年同教授。制御工学、ロボット工学の研究と教育に従事。工学博士。

(日本ロボット学会正会員)

Barkan Ugurlu

Barkan Ugurlu received his Ph.D. degree in Electrical and Computer Engineering from Yokohama National University, Yokohama, Japan, in March 2010. From May 2010 to March 2013, he was a Post-Doctoral Researcher, at the Istituto Italiano di Tecnologia, Genova, Italy, and Toyota Technological Institute, Nagoya, Japan. Between March 2013 and February 2015, he was a Research Scientist at the Computational Neuroscience Laboratories, Advanced Telecommunications Research Institute International (ATR), Kyoto, Japan. He currently holds an Assistant Professor position at the Dept. of Mechanical Engineering, Ozyegin University, Istanbul, Turkey.

松本大河 (Taiga Matsumoto)

2014 年豊田工業大学先端工学学科卒業。現在トヨタ自動車株式会社勤務。在学中 4 脚ロボットの制御システムの研究に従事。

川西通裕 (Michihiro Kawanishi)

1996 年京都大学大学院工学研究科応用システム科学専攻博士課程を中途退学し同年神戸大学工学部機械工学科助手、2006 年より豊田工業大学工学部先端工学基礎学科准教授となり現在に至る。制御理論と数値最適化およびその応用に関する研究に従事。博士 (工学)。システム制御情報学会、日本機械学会、IEEE の会員。

(日本ロボット学会正会員)