

二足歩行ロボットの製作と制御

システム科学技術学部 機械知能システム学科
 2年 笠原 淳平
 指導教員 システム科学技術学部 情報工学科
 准教授 石井 雅樹
 助 教 伊東 嗣功
 教 授 堂坂 浩二

1 緒言

二足歩行や四足歩行を行うロボットでは、関節部分にアクチュエータを組み込む設計と
 なっていることが多い。この構造はアクチュエータから直接回転を取り出せるため合理的
 であるが、関節部が大きくなり動作が限定される。また、関節部の強度がサーボモータの回
 転軸強度に依存するという問題がある。

本研究では、上述の問題を解決するために生物の筋骨格構造を参考にした二足歩行ロボ
 ットの製作と、その制御を行うことを目指した。

2 機体の製作

2.1 使用したサーボモータ

ワイヤを巻き取るためのサーボモータとして、近藤科学製の「KRS-2552RHV ICS」を使用
 した。以下の表1に主要な仕様を示す。

表1. KRS-2552RHV ICS 仕様

項目	諸元
重量	41.5g
最大動作角	270° (±135°)
定格電圧	9 ~ 12V
最大トルク	14.0kg・cm (11.1V動作時)
最大スピード	0.14sec/60° (11.1V動作時)

KRS-2552RHVはシリアル制御方式のサーボモータであり、半二重シリアル通信により特定
 のコマンドを送ることで、その時点での角度、電流、温度を取得することができる。また、
 サーボモータごとに個別のIDを割り振ることで、デジチェーン接続が可能となる。

2.2 使用したワイヤ

関節を駆動するワイヤの特性としては、強度が高く、伸び率が低いことが望ましい。そこ
 で、釣り糸として使用されているPEラインを採用した。PEラインの伸び率は一般に4%以下
 といわれており、伸び率の非常に小さい単糸を拵った構造となっている。その構造上、結束
 強度は直線強度の50%以下になってしまうため、多少の余裕をもって太さ10号のものを使用
 した。

2.3 機体設計

ヒトやイヌなどの生物は、筋肉の収縮運動により骨端に固定された腱を引っ張ることで関節の曲げ動作を行っている。今回製作したロボットは、フレームに固定したワイヤをサーボモータで巻き取ることで関節を駆動する設計とした。

2.3.1 膝関節

ワイヤによる駆動を行うためには、関節部のモーメントアームを確保する必要がある。そのため、膝関節は以下の図1,2のような設計とした。

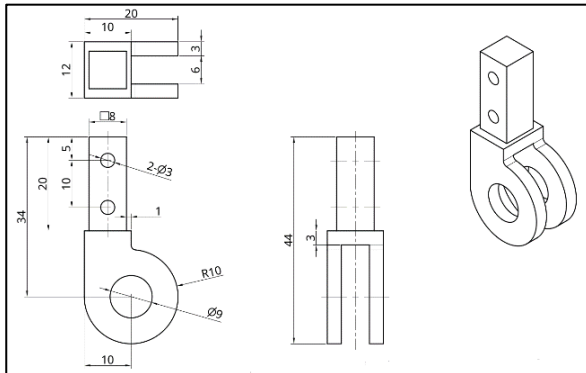


図1. 膝関節 上側

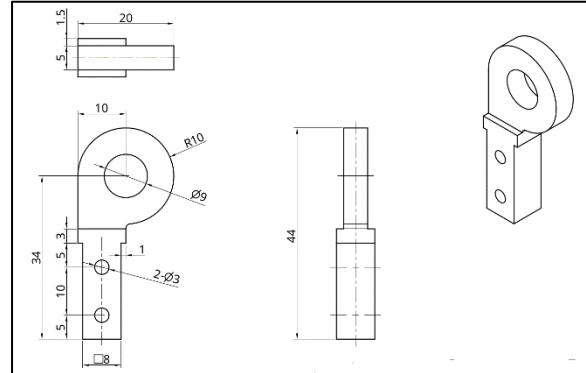


図2. 膝関節 下側

2.3.2 サーボモータ周辺

サーボモータをフレームに固定するためのケースと、ワイヤを巻き取るためのサーボホーンは3Dプリンタで製作した。成形誤差を考慮するために何度か試行錯誤を行う必要があったが、3Dプリンタを用いることで複雑な形状の部品を製作し、サーボモータの固定部をコンパクトにまとめることが可能となった。

3Dプリンタで部品を製作するにあたって、出力時間を減らすために可能な限り部品の体積が少なくなるような設計を行った。それでも、大腿部外側のサーボモータ固定部品を出力するには4時間程の時間を要した。

2.4 組み立て

最終的に組み立てた機体の全体画像を図3に示す。機体の全重量は約2.1kgであり、全長は約46cm（基板コネクタ上部まで）である。

機体の調整を行う際に、当初は腰部のフレームに直接糸を結び付けて吊るしていたが、安定性が低く整備し難かったため簡易的な吊り天秤を作成した。機体側にも吊り天秤を使用するための部品を取り付けることで、機体の整備性が格段に向上した。

駆動用のワイヤはできる限り弛みを小さくする必要があったため、両手にピンセットを持って微調整をしながら機体に結び付けた。現在の設計では、取り付けスペースの関係でワイヤ長を調整するための部品が付いていな

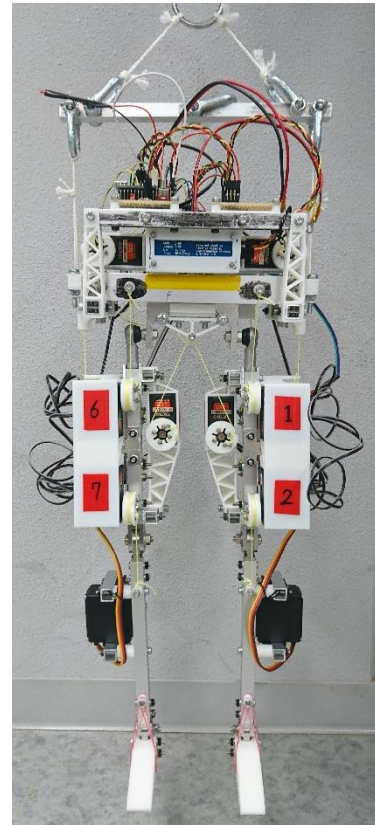


図3. 機体画像

いが、ワイヤの初期調整や伸びによる影響を抑えるためにも必要だと感じた。

3 制御

3.1 構成

ロボットのシステム構成は以下の図4のようにした。

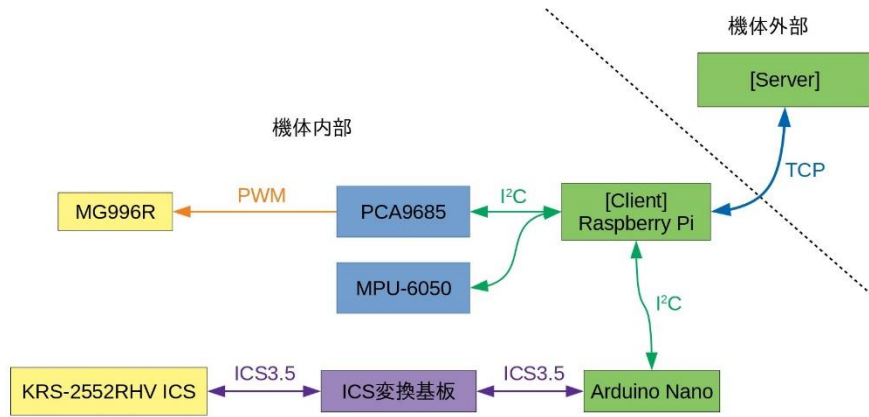


図4. システム構成

図中のPCA9685はPWM信号出力ドライバであり、サーボモータMG996Rの制御のために用いた。

MPU-6050は6軸センサであり、姿勢計算を行うため腰部に固定している(図5)。

ICS3.5はKRS-2552RHVに制御コマンドを送るための通信規格であり、ソフト側でループバックを処理する手間を省くためにICS変換基板を介して通信を行っている。

Raspberry Pi側のプログラムはすべてRustで記述したが、使用したライブラリの問題でシリアル通信が上手く動作しなかったため、I²C通信で一度Arduinoに制御データを転送してからサーボモータを制御するようにした。I²Cの通信速度は100kbpsと低速なので、データ送信の際は16bit符号なし整数のポジションデータを2byteに分割して送信している。また、Raspberry PiとArduinoは動作電圧が異なるため、Nch MOSFETを用いたレベル変換回路を作成した。Nch MOSFETを用いたのは、I²Cバスが双方向通信を行っているためである。

二足歩行のための制御計算をRaspberry Pi単体で行うことは負荷が大きいと考えたため、外部のサーバーとEthernetにより接続できるようにした。サーバーには姿勢角や関節の角度など、小数で表されるデータを送る必要があったため、数値をUTF-8符号化形式で変換してから送信している。

3.3 腰部の姿勢推定

腰部の姿勢を求めるためには、センサから取得した角速度を積分する必要がある。オイラー角による姿勢表現は計算過程が複雑に感じたため、今回は四元数による姿勢表現を用いた。

四元数の基底元をそれぞれ $1, i, j, k$ として、四元数 \tilde{q} を以下のように表記する。

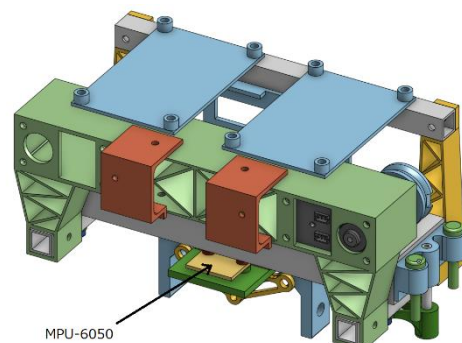


図5. MPU-6050 固定位置

$$\tilde{q} = q_0 + q_1i + q_2j + q_3k \quad (1)$$

ここで、基底間の積は

$$i^2 = j^2 = k^2 = ijk = -1 \quad (2)$$

の性質を満たす。

角速度ベクトルを ω [rad/s]とすると、四元数の時間微分は回転の合成より導出することができ[1]、

$$\frac{d\tilde{q}}{dt} = \frac{1}{2}\omega\tilde{q} \quad (3)$$

となる。これより、微小時間 Δt においては、更新前の四元数 \tilde{q} に対して

$$\tilde{q} \leftarrow \tilde{q} + \frac{\Delta t}{2}\omega\tilde{q} \quad (4)$$

とすることで四元数を更新する。回転を表すクォータニオンにはノルムが1という制約があるが、(4)式は近似式であるため最後に正規化を行う必要がある。

4 結言

生物の筋骨格構造を参考にした二足歩行ロボットと、制御を行うための基盤となる回路・プログラムの製作を行った。

ワイヤ駆動を採用したことで、関節部の小型化とサーボモータの軸受強度に依存しない設計が可能となった。しかし、全体的な動作の自由度に関しては、関節部にアクチュエータを組み込むタイプよりも低くなってしまった。例えば、現状の機体ではサーボモータの干渉により膝関節を100度以上曲げることができない。この問題は、下腿部のサーボモータを現状よりも下の位置に固定することで多少改善できるが、その場合でも人間と同様の曲げ角を達成することは困難である。根本的な解決のためには、今回使用したサーボモータよりも小型のものが必要となるが、単純な歩行のみであれば現状の動作角でも充分可能だと感じた。

現時点で機体状況を知るために取得できる情報は脚部の角度と腰部の姿勢のみであり、周囲の状況は気温しか知ることが出来ない。高度な制御を行うためには、外力を取得できるようにすることが重要である。当初の計画では、実際に姿勢制御を行うところまでを目標としていたが、現状では制御プログラムを製作するための時間と知識が不足しているため、本稿はここまでの成果をもって取りまとめた。

参考文献

- [1] 矢田部 学：クォータニオン計算便利ノート，MSS技法 Vol.18 (2007/03/01 発行)
(<http://www.mss.co.jp/technology/report/pdf/18-07.pdf>)