

アームを動かす 順運動学と逆運動学

吉田 智章

TIrobo01-CQのアーム(マニピュレータともいう)は、RS-485で接続されたモータ・ドライバで駆動されます。その際、モータ制御モジュールに接続されたポテンショ・メータの値によって、モータの目標角度を指示します。より正確には、ポテンショ・メータの角度に応じて変化する電圧をA-Dコンバータで取り込んだ、その値を指示することになります。

アームをプログラムから自由に扱うには、アームの姿勢を把握すること、姿勢を指示することが必要になります。ここでは、アームの姿勢を表すのに、アームの根元から見た手先(ハンド)の位置を使うことにします。

● 関節角を把握する

モータ制御モジュールは、ポテンショ・メータの状態を目標値と

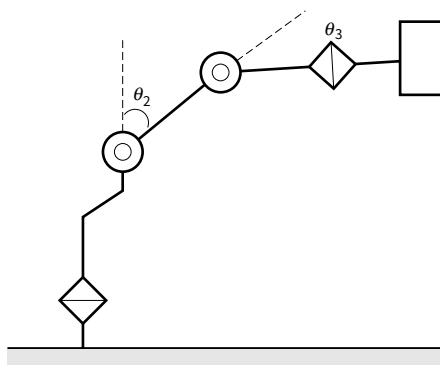


図1 アームの原点

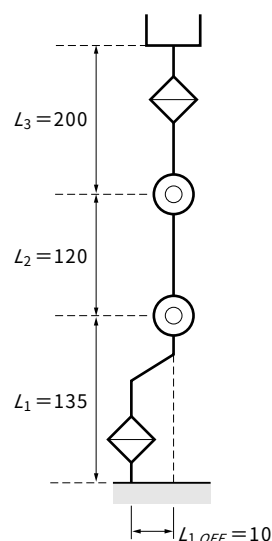


図2 アームの形状

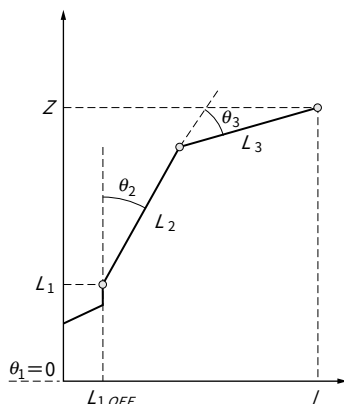


図3 アームの順運動学

する制御を行います。モータの先に取り付けられている関節などが具体的にどういう姿勢にあるのか、実際にはわかっていません。モータ制御モジュールに問い合わせてわかるのは、現在の関節角ではなく、ポテンショ・メータの状態だけなのです。そのため、実際の関節角を知るにはポテンショ・メータの状態を表す数値を換算する必要があります。

関節に取り付けたポテンショ・メータは、関節の動作した角度と同じだけ回転します。ポテンショ・メータの抵抗値は角度と線形に対応するので、関節の角度を求めるには適切なオフセットとゲインを求めればよいことになります。

まず、アームの各関節の原点方向を決めます(図1)。もちろん、これは決めただけなので、別の方向を原点に取ってもかまいません。

この原点方向に関節を向けたときのポテンショ・メータの値と、一定角度動かしたときのポテンショ・メータの値の変化量を各関節ごとに計測します。これらの値がオフセットとゲインになります。

● 姿勢を把握する — 順運動学

各関節角がわかると、アームの姿勢が一意に決まります。しかし関節角のままでは扱いにくいので、手先の位置を求めます。

アームの関節の配置とリンクの長さを図2に示します。グリップ(指)の開閉を含めると全部で5自由度ありますが、ここでは手首の回転とグリップの自由度を除いた3自由度のみを考えることにします。

通常は、根元の関節から順に平行移動、回転の行列をかけた変換行列を用いて手先の座標系と根元の座標系の変換を行います。しかし、TIrobo01-CQのマニピュレータは、根元の関節 θ_1 を無視すると残りは2次元平面内で話が閉じて簡単なので、幾何学的に直接求めることにしました。

まず、手先の高さ z と手先を地面に投影した点の根元からの距離 l を求めます。これは L_1 、 L_2 、 L_3 の長さ θ_2 、 θ_3 の角度から計算することができます。次に l と θ_1 の角度から手先を地面に投影した点の位置 (x, y) を求めます(リスト1、図3)。

リスト1 順運動学

```
tPos3 armFK(int *aArm)
{
    double th[3];
    tPos3 pos;
    double l;

    th[0]=armBaseAngle(aArm[0])/RAD2DEG; // J1 角度
    th[1]=-armShoulderAngle(aArm[1])/RAD2DEG; // J2 角度
    th[2]=-armElbowAngle(aArm[2])/RAD2DEG; // J3 角度

    // 手先を地面に投影した点と根本の距離
    l=L1OFF + L2*sin(th[1]) + L3*sin(th[1]+th[2]);
    pos.p[0]= l*cos(th[0]); // X
    pos.p[1]= l*sin(th[0]); // Y
    pos.p[2]= L1 +L2 *cos(th[1]) + L3*cos(th[1]+th[2]); // Z

    return pos;
}
```

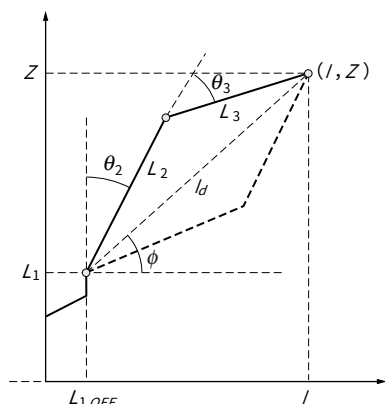


図4 アームの逆運動学

リスト2 逆運動学

```

tPos3 armIK(tPos3 aPos)
{
    tPos3 ang;
    double th[3];
    double phi,ld,l;

    th[0]=atan2(aPos.p[1],aPos.p[0]); //J1はx,yから直接求める
    // J2,J3 は余弦定理を使って求める
    l=sqrt(aPos.p[0]*aPos.p[0] + aPos.p[1]*aPos.p[1]);
    ld=sqrt((1-L1OFF)*(1-L1OFF) + (aPos.p[2]-L1)*(aPos.p[2]-L1));
    phi=atan2((aPos.p[2]-L1),10-L1OFF);
    th[1]=-M_PI/2 + phi + acos((ld*ld+L2*L2-L3*L3)/(2*ld*L2)); //J2
    th[2]=-M_PI + acos((L2*L2+L3*L3-Ld*ld)/(2*L2*L3)); // J3
    ang.p[0]=th[0];
    ang.p[1]=th[1];
    ang.p[2]=th[2];
    return ang;
}

```

● 姿勢を指示する — 逆運動学

手先の位置(と姿勢)から、それを満足する関節角を求めることを逆運動学を解くといいます。

逆運動学も、まずは x, y 座標から根元の θ_1 の角度を求めてしまえば、二次元平面内の2自由度アームとして計算できます(図4)。一つの手先位置に対してとりえるアームの姿勢は図の実線の姿勢と、対称の位置にある太い破線の姿勢の2通り考えられます。

しかし、ここで使用するアームの関節 θ_3 の動作範囲が限られてい

るため、実際には太い破線の姿勢をとることはできません。結局手先座標 x, y, z が決まると、(もし届くのなら) $\theta_1 \sim \theta_3$ の各関節角も一意に決定できます(リスト2)。

よしだ・ともあき 千葉工業大学 未来ロボット技術研究センター

基板 & 部品の頒布のお知らせ (編集部)

TiRobo01-CQ に利用されている下記の部品を有償で頒布します。

● センサ・モジュール

▶ 注文先: FITDESIGN (<http://www.fitdesign.biz/>)

▶ 予定価格(送料別)

● センサ・モジュール(TIR1-S/CQ): 8,500 円

● ICSP ケーブル: 400 円

● 回路増設用部品一式: 1,000 円

● モータ制御モジュール

▶ 注文先: (株)アールティ (<http://www.rt-net.jp/index.shtml>)

▶ 開発元: テクノチップス(株)

● ロボット・アーム・キット「ELEKIT MR-999」

▶ 注文先: (株)アールティ

▶ 開発元: (株)イーケイジャパン

● 統括制御モジュール(超小型 NetBSD サーバ)

▶ 注文先: (株)アールティ

▶ 予定価格(送料別): 52,500 円

▶ 開発元: (株)ブレインズ

※ なお、走行制御モジュールの基板頒布の予定はありません。走行制御モジュールに使用した SH-7045F マイコン・ボードは、(株)秋月電子通商で入手できます。

