

講義予定

1. ロボットアームの運動方程式とその性質
2. 冗長な一般化座標・拘束力・低次元化
3. 非線形システムの安定性解析
4. モデルベース軌道追従制御法 (分解速度制御・分解加速度制御)
5. VSS 軌道追従制御
6. 関節に剛性を有する場合の制御法・フレキシブルアームの制御
7. インピーダンス制御
8. 力と位置のハイブリッド制御
9. マスタ・スレーブシステム (VIM・通信遅れのある場合)
10. 適応制御 (モデルベース)
11. 適応制御 (ANN ベース)
12. 学習制御
13. 車両型移動ロボットの制御 (零ダイナミクスと零点の関係)

第 1 章

Lagrange の運動方程式とその性質

1.0.1 Lagrange の運動方程式の基本的性質

一般座標 q , それに双対な一般化力 τ によって Lagrange の運動方程式を記述すると

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau \quad (1.1)$$

となる。ここで、 $D(q)\ddot{q}$ は慣性項、 $C(q, \dot{q})\dot{q}$ は中心・コリオリ力項、 $G(q)$ は重力項である。(導出等詳細に関してはロボット工学第一資料参照のこと)

この運動方程式に関する性質をまとめると次のようになる。

[性質 1] $D(q)$ は対称行列。

[性質 2] $D(q)$ は正定行列であり、関節が回転ジョイントである場合には次の不等式満足する正の定数 λ_m, λ_M が存在する。

$$0 < \lambda_m \leq \lambda(D(q)) \quad (1.2)$$

$$\bar{\lambda}(D(q)) \leq \lambda_M < \infty \quad (1.3)$$

ただし、 $\lambda(M)$ は行列 M の最小固有値、 $\bar{\lambda}(M)$ は最大固有値を表す

[性質 3] $C(q, \dot{q})$ は \dot{q} に関して有界、つまり

$$\|C(q, \dot{q})\| \leq C_M \|\dot{q}\| \quad (1.4)$$

[性質 4] $C(q, \cdot)$ は次の性質を満たす。

$$C(q, x)y = C(q, y)x \quad (1.5)$$

$$C(q, x + \alpha y) = C(q, x) + \alpha C(q, y) \quad (1.6)$$

[性質 5] $K := \dot{D}(q) - 2C(q, \dot{q})$ は歪対称、つまり

$$K + K^T = 0 \text{ または } x^T K x = 0 \text{ for } \forall x \quad (1.7)$$

[性質 6] 重力項 $G(q)$ は有界な q に対して有界となる。

[性質 7] 適当なパラメータベクトル a を選択することにより、(1.1) の右辺は、既知関数を要素とする行列 Y と a との積で表現することが出来る。つまり、

$$Y(q, \dot{q}, \ddot{q})a = \tau \quad (1.8)$$

1.0.2 運動方程式の座標変換

前節では、一つの一般化座標 q によって運動方程式を記述し、その性質を議論したが、制御問題によっては最初に定式化した座標とはことなる座標によって系を記述した方が便利な場合がある。例えば、ロボットアームの制御を考えた場合、運動方程式はロボットアームの手先の座標によって記述された方が都合がよい。また、異なる座標を導入することにより、変数が干渉しないシンプルな表現を得ることが可能な場合もある。そのために、ここでは一つの座標で記述された運動方程式を他の座標による表現によって記述する方法を説明する。

今、形式的に元の座標を q 、新しい座標を x により記述し、(非線形な)座標変換

$$x = f(q) \quad (1.9)$$

により関係づけられているとする。ただし、 f は局所的に微分同形 (微分可能な逆関数が存する) であるとする。これより、両辺を時間で微分して、

$$\dot{x} = \frac{\partial f}{\partial q} \dot{q} = J(q) \dot{q}, \quad J(q) := \frac{\partial f}{\partial q} \quad (1.10)$$

$$\ddot{x} = J(q) \ddot{x} + \dot{J} \dot{q} \quad (1.11)$$

なる関係が得られ、これを利用して次の関係が得られる。

$$q = f^{-1}(x) \quad (1.12)$$

$$\dot{q} = J^{-1} \dot{x} \quad (1.13)$$

$$\ddot{q} = J^{-1} (\ddot{x} - \dot{J} J^{-1} \dot{x}) \quad (1.14)$$

これらの関係を形式的に式 (1.1) に代入すると

$$D(f^{-1}(x)) J^{-1} \ddot{x} + (C(f^{-1}(x), J^{-1} \dot{x}) - D(f^{-1}(x)) \dot{J}) J^{-1} \dot{x} + G(f^{-1}(x)) = \tau \quad (1.15)$$

となる。しかし、このままでは右辺は元の一般化座標 q に対する一般化力のままであるので、これを新しい座標 x に対応するもの書き換える。今、一般化座標、一般化力の定義によりある力 τ によって δq だけ q が変化したとすると、その時の仕事 δW は

$$\delta W = \tau^T \delta q \quad (1.16)$$

である。この τ と同等な力が系に作用して、新しい座標で δx だけ変化したとすると、変化量は

$$\delta x = J \delta q \rightarrow \delta q = J^{-1} \delta x \quad (1.17)$$

の関係があり、なされた仕事は等しくなければならないので

$$\delta W = \tau^T \delta q = \tau^T J^{-1} \delta x = (J^{-T} \tau)^T \delta x \quad (1.18)$$

となる。従って、新しい座標に等価的に働く一般化力を f とすると

$$f = J^{-T} \tau \quad (1.19)$$

でなければならないことが分かる。従って、式 (1.15) の量辺に J^{-T} を左から掛けることにより新しい座標での力の釣合い式になる。つまり、

$$J^{-T} D(f^{-1}(x)) J^{-1} \ddot{x} + J^{-T} (C(f^{-1}(x), J^{-1} \dot{x}) - D(f^{-1}(x)) \dot{J}) J^{-1} \dot{x} + J^{-T} G(f^{-1}(x)) = J^{-T} \tau \quad (1.20)$$

ここで、

$$\begin{aligned} M_x(x) &:= J^{-T} D(f^{-1}(x)) J^{-1} \\ C_x(x, \dot{x}) &:= J^{-T} (C(f^{-1}(x), J^{-1} \dot{x}) - D(f^{-1}(x)) \dot{J}) J^{-1} \\ G_x(x) &:= J^{-T} G(f^{-1}(x)) \end{aligned}$$

と定義すると、新しい座標での運動方程式は形式的に式 (1.1) と全く同じ形の次の式となる。

$$M_x(x) \ddot{x} + C_x(x, \dot{x}) \dot{x} + G_x(x) = f \quad (1.21)$$

この新しく導出された運動方程式も局所的には元の運動方程式と同様 [性質 1] ~ [性質 7] が成り立つ。(例えば、もとの運動方程式の [性質 5] を導いてみよ。)

1.0.3 拘束運動の表現 (冗長自由度による運動方程式表現)

人間型のロボットのように、関節によって剛体が複雑にからみあっている系の場合、歩行や転倒によって運動の拘束が変化する可能性がある。そのような場合、状況に応じた独立な一般化座標を導入して系の運動方程式を記述するよりも、独立でない一般化座標を導入して冗長に系を記述した方が運動方程式を簡潔な形により記述する方が適している場合がある。ここでは、冗長な一般化座標を導入し、それらの従属関係を速度拘束の形により明示的に記述し、運動方程式を得る方法について説明する。

速度拘束の表現

2つの一般化座標 $q_1, q_2 (q_1 \in R^{m_1}, q_2 \in R^{m_2})$ があり、式

$$f(q_1, q_2) = 0, \quad f : R^{m_1} \times R^{m_2} \rightarrow R^n \quad (1.22)$$

により拘束が表されている場合を考える。もちろん、特殊なケースとして

$$q_2 = f'(q_1) \quad (1.23)$$

と表される場合もあるが、これは

$$-f'(q_1) + q_2 = 0$$

と考えることにより、式 (1.22) と同じになる。式 (1.22) の両辺を時間で微分すると

$$\frac{\partial f}{\partial q_1} \dot{q}_1 + \frac{\partial f}{\partial q_2} \dot{q}_2 = 0 \quad (1.24)$$

となり、速度の拘束式の形の書き表すことができる。これを簡単のために、

$$J_c \dot{q} = 0 \quad (1.25)$$

ただし、

$$J_c := \begin{bmatrix} \frac{\partial f}{\partial q_1} & \frac{\partial f}{\partial q_2} \end{bmatrix}$$

$$\dot{q} := \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}$$

とする。(今はこの速度の拘束を (q_1, q_2) の関数より導いたが、三輪移動ロボットなどのように直接この関係式が与えられる場合もある。)

拘束条件付運動方程式

今、 q に関して全く拘束がなく独立であるとして運動方程式を立てると、形式的には式 (1.1) となる。しかし、この運動方程式によって座標が変化すると、一般に式 (4.23) を満たさない。実際のシステムが式 (4.23) を満たすために、外から加わる力 τ 以外に、拘束を満たすための力を考える。この力のことを拘束力といい形式的に F_c と表すことにする。つまり、運動方程式を

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau - F_c \quad (1.26)$$

とする。この F_c は形などを保つ拘束の力であるので、系に対して仕事をしてはならない。今考えている座標に対する F_c がする仕事は

$$\delta W = F_c^T \delta q \quad (1.27)$$

であるので、これは零でなければならない。これを満たす基底ベクトルは式 (4.23) の J_c を

$$J_c = \begin{bmatrix} c_1^T \\ \cdot \\ c_r^T \end{bmatrix} \quad (1.28)$$

と表すと、

$$F_c = \sum_{i=1}^r c_i \lambda_i = J_c^T \lambda, \quad \lambda = \begin{bmatrix} \lambda_1 \\ \cdot \\ \lambda_r \end{bmatrix} \quad (1.29)$$

と表される。ただし、 λ_i は未知な実数である。これは、 $J_c \dot{q} = 0$ より明らかである。この F_c を用いて運動方程式を表すと、

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau - J_c \lambda \quad (1.30)$$

となる。元の方程式では、状態 (q, \dot{q}) と τ が与えられると、加速度が分かればそれらを時間で積分することにより、以後の状態が分かるということで、未知数は \ddot{q} だけであった。この場合には、方程式の数と \ddot{q} の要素の数が等しく、 $M(q)$ が正則であるかぎり階は一意に解ける。しかし、 λ の導入により未知数の数が方程式の数よりも多くなり、このままでは (\ddot{q}, λ) を解くことができない。そこで、これを解くために、他の方程式を得る必要がある。このために、速度拘束式を時間で微分することにより、

$$J_c \ddot{q} = -\dot{J}_c \dot{q} \quad (1.31)$$

なる式を得、 \ddot{q} に関するもう一つの方程式を得ることができる。これと元の運動方程式を連立して、本質的な未知変数である (\ddot{q}, λ) に関して方程式を行列表現すると

$$\begin{bmatrix} D(q) & J_c^T \\ J_c & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ \lambda \end{bmatrix} = \begin{bmatrix} -C(q, \dot{q})\dot{q} - G(q) + \tau \\ -\dot{J}_c \dot{q} \end{bmatrix} \quad (1.32)$$

となる。ここで、

$$\text{rank} \begin{bmatrix} D(q) & J_c^T \\ J_c & 0 \end{bmatrix} = m_1 + m_2 + r \quad (1.33)$$

であれば運動は一意に定まることになる。

形式的な自由度の低減

上述の方法では、 λ を明示的に求めることができる。しかし、 λ を求める必要がない場合には、以下のように考える方程式の次元を形式的に減らすことができる。今簡単のために $r = m_2$ であるとし、

$$J_c = [J_1 \quad J_2] \quad (1.34)$$

としたとき、 $\text{rank} J_2 = r$ であると仮定する。 (q_1, q_2) の要素適当な列ベ替えにより必ずこの条件を満たすようにできる。) こう仮定すると、 J_2 は正方な可逆行列となるので、 J_c に左から J_2^{-1} を乗じると

$$J_2^{-1} J_c = [J_2^{-1} \quad I_r] =: [\bar{J}_1 \quad I_r] \quad (1.35)$$

となる。この行列を \bar{J}_c と定義して、運動方程式を書き換えると

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau - \bar{J}_c^T \bar{\lambda} \quad (1.36)$$

ただし、

$$\bar{\lambda} := J_2^{-T} \lambda$$

である。ここで \bar{J}_c を用いると、式 (4.23) より

$$\dot{q}_2 = -\bar{J}_1 \dot{q}_1 \quad (1.37)$$

と書き表せることが分かる。従って、 \dot{q} は

$$\dot{q} = \begin{bmatrix} I_{m_1} \\ -\bar{J}_1 \end{bmatrix} \dot{q}_1 =: J \dot{q}_1, \quad J := \begin{bmatrix} I_{m_1} \\ -\bar{J}_1 \end{bmatrix} \quad (1.38)$$

と書きあらせる。これより形式的に拘束力を消去し、式 (1.36) を q_1 のみで表現することができる。ただし、

$$q_2(t) = q_2(0) - \int_0^t \bar{J}_1 \dot{q}_1(\tau) d\tau \quad (1.39)$$

により計算するものとする。

実際、式 (1.37) より

$$\ddot{q} = J\ddot{q}_1 + \dot{J}\dot{q}_1 \quad (1.40)$$

であるので、これを式 (1.36) に代入して、左から J^T を掛けると、

$$J^T J_c^T = [I_{m_1} \quad -\bar{J}_1^T] \begin{bmatrix} \bar{J}_1^T \\ I_r \end{bmatrix} = \bar{J}_1^T - \bar{J}_1^T = 0 \quad (1.41)$$

になることを注意すると

$$J^T D J \ddot{q}_1 + J^T (C J + D \dot{J}) \dot{q}_1 + J^T G = J^T \tau \quad (1.42)$$

となる。ここで、

$$\begin{aligned} D_1 &:= J^T D J \\ C_1 &:= J^T (C J + D \dot{J}) \\ G_1 &:= J^T G \\ \tau_1 &:= J^T \tau \end{aligned}$$

と定義すると、 m_1 次元の運動方程式

$$D_1 \ddot{q}_1 + C_1 \dot{q}_1 + G_1 = \tau_1 \quad (1.43)$$

を得る。ここで注意しなければならないのは、速度拘束が最初に与えられた場合には必ずしも

$$q_2 = f'(q_1) \quad (1.44)$$

というように表現できないことがあるため、 $D_1(q)$ は必ずしも q_1 のみの関数表現が得られるとは限らない。(もちろん、最初から $q_2 = f'(q_1)$ によって表される場合には当然その関係が使える。)

完全非弾性衝突のモデリング

ロボットの足をゴムなどの跳ね返りのないもので作り、歩行時に足をついた瞬間に速度が不連続に変化するような現象をモデリングすることを考える。これは、複雑なリンク機構の系が衝突する際の運動解析には不可欠である。ここでは、衝突は離散時間的に起こるとし、一つの衝突の起きた時刻を 0 時刻と考え、衝突前の速度を \dot{q}_- 、衝突直後の速度を \dot{q}_+ により表す。完全非弾性衝突の場合、ある自由度は衝突の際に拘束されてしまうと考えられる。つまり、

$$J_c(q)\dot{q}_+ = 0 \quad (1.45)$$

が満たされるとする。この現象を解析するために、 $(0_-, 0_+)$ の間にデルタ関数的な拘束力が働くと仮定し、これを

$$F_c = J_c^T \hat{\lambda} \delta(t), \quad \hat{\lambda} \in R^r \quad (1.46)$$

と表す。ただし、 $\hat{\lambda}$ は定数ベクトルである。これを運動方程式に入れると、拘束運動を考えたときと同様に

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau - J_c^T \hat{\lambda} \delta(t)$$

と表される。ここで外部入力 (制御入力) には δ 関数が含まれないとし、両辺を時間により 0_- から 0_+ まで積分すと、

$$\int_{0_-}^{0_+} (D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q))dt = \int_{0_-}^{0_+} (\tau - J_c^T \hat{\lambda} \delta(t))dt \quad (1.47)$$

となり、ここで $0_- \rightarrow 0, 0_+ \rightarrow 0$ とすると \dot{q} は不連続かもしれないが有限の値を取るため積分には影響せず、また q はその積分区間で一定とみなせるため、

$$D(q)q_+ - D(q)q_- = -J_c^T \hat{\lambda} \quad (1.48)$$

となる。式 (1.45) と式 (1.48) を連立すると

$$\begin{bmatrix} D(q) & J_c^T \\ J_c & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_+ \\ \hat{\lambda} \end{bmatrix} = \begin{bmatrix} D(q)\dot{q}_- \\ 0 \end{bmatrix} \quad (1.49)$$

となる。そこで、この方程式が解を持てば衝突現象は一意に解くことができる。これを $\hat{\lambda}$ について解くと

$$\hat{\lambda} = (J_c D^{-1} J_c^T)^{-1} J_c \dot{q}_- \quad (1.50)$$

となり、これを用いて \dot{q}_+ は

$$\dot{q}_+ =: P \dot{q}_-, \quad P := (I_n - D^{-1} J_c^T (J_c D^{-1} J_c^T)^{-1} J_c) \quad (1.51)$$

となる。この P は \dot{q}_- を \dot{q}_+ の拘束空間に射影している射影行列となっている。(PP = P であることを確かめてみよ。)

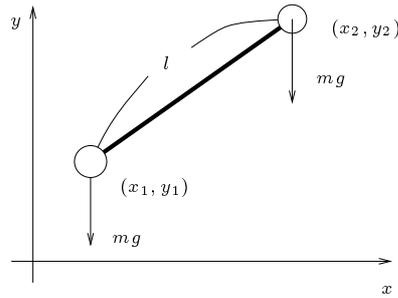


図 1.1: 拘束された 2 質点系

1.0.4 例

上記のことを簡単な例によって確かめてみよう。

冗長な座標を用いた運動方程式 図 1.0.4 のようにそれぞれの質量が m の 2 つの質点が重さのない長さ l 棒によってつながって運動している状況を考える。それぞれの質点について拘束を無視して運動方程式を立てると、質点 1 に関して

$$m\ddot{x}_1 = 0 \quad (1.52)$$

$$m\ddot{y}_1 = -mg \quad (1.53)$$

質点 2 に関しては

$$m\ddot{x}_2 = 0 \quad (1.54)$$

$$m\ddot{y}_2 = -mg \quad (1.55)$$

となる。これに対して実際には次のホロノミックな拘束：

$$(x_1 - x_2)^2 + (y_1 - y_2)^2 = l^2 \quad (1.56)$$

が存在する。これを両辺時間で微分して、速度拘束式を計算すると、次の式を得る。

$$[2(x_1 - x_2) \ 2(y_1 - y_2) \ -2(x_1 - x_2) \ -2(y_1 - y_2)] \begin{bmatrix} \dot{x}_1 \\ \dot{y}_1 \\ \dot{x}_2 \\ \dot{y}_2 \end{bmatrix} = 0 \quad (1.57)$$

従って、拘束力を用いた実際の運動方程式は

$$\begin{bmatrix} m & & & \\ & m & & \\ & & m & \\ & & & m \end{bmatrix} \begin{bmatrix} \ddot{x}_1 \\ \ddot{y}_1 \\ \ddot{x}_2 \\ \ddot{y}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ mg \\ 0 \\ mg \end{bmatrix} = - \begin{bmatrix} 2(x_1 - x_2) \\ 2(y_1 - y_2) \\ -2(x_1 - x_2) \\ -2(y_1 - y_2) \end{bmatrix} \lambda \quad (1.58)$$

となる。運動中の拘束力の成分 λ は次の 2 つの式：

$$\begin{bmatrix} \ddot{x}_1 \\ \ddot{y}_1 \\ \ddot{x}_2 \\ \ddot{y}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ -g \\ 0 \\ -g \end{bmatrix} - \frac{2}{m} \begin{bmatrix} (x_1 - x_2) \\ (y_1 - y_2) \\ -(x_1 - x_2) \\ -(y_1 - y_2) \end{bmatrix} \lambda \quad (1.59)$$

$$2[(x_1 - x_2)(y_1 - y_2) - (x_1 - x_2) - (y_1 - y_2)] \begin{bmatrix} \ddot{x}_1 \\ \ddot{y}_1 \\ \ddot{x}_2 \\ \ddot{y}_2 \end{bmatrix} = -2 \{ (\dot{x}_1^2 - \dot{x}_2^2) + (\dot{y}_1^2 - \dot{y}_2^2) \} \quad (1.60)$$

を連立して、

$$\lambda = \frac{m}{4l^2} \{(\dot{x}_1^2 - \dot{x}_2^2) + (\dot{y}_1^2 - \dot{y}_2^2)\} \quad (1.61)$$

を得る。

形式的な低次元化 今運動方程式の変数の数を減少させるために、式の中から \dot{y}_2 を消去することを考える。そのため $y_1 - y_2 \neq 0$ を仮定しよう。そうすると、拘束式より

$$\dot{y}_2 = \frac{1}{y_1 - y_2} ((x_1 - x_2)\dot{x}_1 + (y_1 - y_2)\dot{y}_1 - (x_1 - x_2)\dot{x}_2) \quad (1.62)$$

となる。これより

$$\begin{bmatrix} \dot{x}_1 \\ \dot{y}_1 \\ \dot{x}_2 \\ \dot{y}_2 \end{bmatrix} = J \begin{bmatrix} \dot{x}_1 \\ \dot{y}_1 \\ \dot{x}_2 \end{bmatrix} \quad (1.63)$$

ただし、

$$J := \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ (x_1 - x_2)/\Delta & (y_1 - y_2)/\Delta & -(x_1 - x_2)/\Delta & \end{bmatrix}, \quad \Delta := y_1 - y_2 \quad (1.64)$$

なる関係を用いて運動方程式を書き直すと

$$\begin{bmatrix} 1 & & (x_1 - x_2)/\Delta \\ & 1 & (y_1 - y_2)/\Delta \\ & & 1 & -(x_1 - x_2)/\Delta \end{bmatrix} \begin{bmatrix} m \\ m \\ m \end{bmatrix} \begin{bmatrix} 1 & & \\ & 1 & \\ (x_1 - x_2)/\Delta & (y_1 - y_2)/\Delta & -(x_1 - x_2)/\Delta \end{bmatrix} \begin{bmatrix} \ddot{x}_1 \\ \ddot{y}_1 \\ \ddot{x}_2 \end{bmatrix} + m J^T \dot{j} \begin{bmatrix} \dot{x}_1 \\ \dot{y}_1 \\ \dot{x}_2 \end{bmatrix} = -2 \begin{bmatrix} 1 & & (x_1 - x_2)/\Delta \\ & 1 & (y_1 - y_2)/\Delta \\ & & 1 & -(x_1 - x_2)/\Delta \end{bmatrix} \begin{bmatrix} x_1 - x_2 \\ y_1 - y_2 \\ -(x_1 - x_2) \\ -(y_1 - y_2) \end{bmatrix} \lambda \quad (1.65)$$

となる。ここで右辺は零となるので、結局

$$m \begin{bmatrix} 1 + \frac{(x_1 - x_2)^2}{\Delta^2} & \frac{(x_1 - x_2)(y_1 - y_2)}{\Delta^2} & -\frac{(x_1 - x_2)^2}{\Delta^2} \\ & 1 + \frac{(y_1 - y_2)^2}{\Delta^2} & -\frac{(y_1 - y_2)(x_1 - x_2)}{\Delta^2} \\ & * & 1 + \frac{(x_1 - x_2)^2}{\Delta^2} \end{bmatrix} + m J^T \dot{j} \begin{bmatrix} \dot{x}_1 \\ \dot{y}_1 \\ \dot{x}_2 \end{bmatrix} + \begin{bmatrix} \frac{(x_1 - x_2)}{\Delta} mg \\ (1 + \frac{y_1 - y_2}{\Delta}) mg \\ -\frac{(x_1 - x_2)}{\Delta} mg \end{bmatrix} \quad (1.66) \quad (1.67)$$

となる。ここで注意しなければならないことは、適当に変数を消去すると、今の場合のように、 $y_1 - y_2 = 0$ の時 \dot{x}_2 を消去するなど、他の変数を消去するように新しい座標を決定しなければならない。

大域的な変数による運動方程式の低次元化 前の例では、単純に 1 つの変数を他の変数によって表すことにより変数の数を減らすことを考えたが、ここでは違う変数を導入することによって運動方程式の変数の数を減らすことを考える。このために新しい変数として棒の回転角 θ を導入する。これによって、

$$x_2 = x_1 + l \cos \theta \quad (1.68)$$

$$y_2 = y_1 + l \sin \theta \quad (1.69)$$

$$\dot{x}_2 = \dot{x}_1 - l \sin \dot{\theta} \quad (1.70)$$

$$\dot{y}_2 = \dot{y}_1 + l \cos \dot{\theta} \quad (1.71)$$

となる。この関係を行列表現して、

$$\begin{bmatrix} \dot{x}_1 \\ \dot{y}_1 \\ \dot{x}_2 \\ \dot{y}_2 \end{bmatrix} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ 1 & & -l \sin \theta & \\ & & 1 & l \cos \theta \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{y}_1 \\ \dot{\theta} \end{bmatrix} \quad (1.72)$$

して、前の例と同様に運動方程式を座標変換すると

$$m \begin{bmatrix} 2 & & -l \sin \theta \\ & 2 & l \cos \theta \\ -l \sin \theta & l \cos \theta & l^2 \end{bmatrix} \begin{bmatrix} \ddot{x}_1 \\ \ddot{y}_1 \\ \ddot{\theta} \end{bmatrix} + m \begin{bmatrix} -l \cos \theta \dot{\theta} \\ -l \sin \theta \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 2mg \\ l \cos \theta mg \end{bmatrix} = 0 \quad (1.73)$$

となる。更に、重心に関して運動方程式を書き直すために重心の座標 (x_g, y_g) を導入すると

$$x_1 = x_g - \frac{l}{2} \cos \theta \quad (1.74)$$

$$y_1 = y_g - \frac{l}{2} \sin \theta \quad (1.75)$$

の関係が成り立つので、この関係を用いて運動方程式を座標変換すると、重心回りにすぐに立てることのできる Newton の運動方程式・ Euler の運動方程式である

$$m \begin{bmatrix} 2 & & \\ & 2 & \\ & & \frac{l^2}{2} \end{bmatrix} \begin{bmatrix} \ddot{x}_g \\ \ddot{y}_g \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 2mg \\ 0 \end{bmatrix} = 0 \quad (1.76)$$

が得られる。(導いてみよ。)

完全非弾性衝突を伴う衝突 今、考えている 2 質点系が回転しながら跳ね返り係数が 0 の床に衝突してそのまま運動することを考える。ここでは回転しながら (x_1, y_1) の座標で示される質点が衝突すると考えて、式 (1.73) で表される運動方程式を用いることにする。壁と衝突することによって一瞬 (x_1, y_1) が拘束されると考えると、衝突後の速度を $(\dot{x}_{1+}, \dot{y}_{1+}, \dot{\theta}_+)$ とすると衝突の際の速度拘束は

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{x}_{1+} \\ \dot{y}_{1+} \\ \dot{\theta}_+ \end{bmatrix} = 0 \quad (1.77)$$

となる。衝突の前の速度を $(\dot{x}_{1-}, \dot{y}_{1-}, \dot{\theta}_-)$ 、拘束インパルスの係数を (λ_1, λ_2) とすると、つぎの関係式が成り立つ。

$$\begin{bmatrix} 2m & 0 & -ml \sin \theta \\ 0 & 2m & ml \cos \theta \\ -ml \sin \theta & ml \cos \theta & ml^2 \end{bmatrix} \begin{bmatrix} \dot{x}_{1+} \\ \dot{y}_{1+} \\ \dot{\theta}_+ \end{bmatrix} = \begin{bmatrix} 2m & 0 & -ml \sin \theta \\ 0 & 2m & ml \cos \theta \\ -ml \sin \theta & ml \cos \theta & ml^2 \end{bmatrix} \begin{bmatrix} \dot{x}_{1-} \\ \dot{y}_{1-} \\ \dot{\theta}_- \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} \quad (1.78)$$

この関係と衝突後の速度拘束とを連立して方程式を (λ_1, λ_2) について解くと

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = m \begin{bmatrix} 1 + \cos^2 \theta & \sin \theta \cos \theta \\ \sin \theta \cos \theta & 1 + \sin^2 \theta \end{bmatrix} \begin{bmatrix} \dot{x}_{1-} \\ \dot{y}_{1-} \end{bmatrix} \quad (1.79)$$

となる。ここで注意すべきことは、拘束力はモーメントの釣合いの式の一番下の部分には影響を与えず、その部分の量は衝突の前と後では変化しないということである。この例では、

$$-ml \sin \theta \dot{x}_{1+} + ml \cos \theta \dot{y}_{1+} + ml^2 \dot{\theta}_+ = [0 \ 0 \ 1] \begin{bmatrix} l \cos \theta \\ l \sin \theta \\ 0 \end{bmatrix} \times \begin{bmatrix} m \dot{x}_{1+} \\ m \dot{y}_{1+} \\ 0 \end{bmatrix} = -ml \sin \theta \dot{x}_{1-} + ml \cos \theta \dot{y}_{1-} + ml^2 \dot{\theta}_- \quad (1.80)$$

となり、 (x_1, y_1) 回りの回転のモーメントが保存されることになる。

第 2 章

非線形システムの安定性解析

線形時不変なシステムの場合、安定性を解析するにはシステムの極を調べれば良いが、入力に関する線形性、初期値に関する線形性のない非線形システムの場合にはそのような手法を用いることができない。また、線形システムの場合には、有界な外乱信号が安定なシステムに入った場合には、システムを不安定にすることがないが、非線形システムの場合にはそのようなことが成り立たない。よって、非線形システムの安定性を解析するには、他の手法を用いる必要があり、また取扱も慎重にする必要がある。

ここでは、次の状態空間表現で与えられる時不変な非線形システムを考える。

$$\dot{x} = f(x), \quad x(0) = x_0 \quad (2.1)$$

ただし、 x_e は平衡状態で

$$f(x_e) = 0 \quad (2.2)$$

を満たすとする。

まず用語の定義をしよう。時不変線形なシステムの場合、安定と言えば通常指数安定のことであるが、非線形システムを扱う場合には考える安定性が多様になる。

[Lyapunov の意味での安定 (安定)]

次の関係が成り立つ時、平衡点 x_e は Lyapunov の意味で安定という。(図 2.1 参照)

$$\forall \epsilon > 0, \exists \delta(\epsilon) > 0, \|x_0 - x_e\| < \delta \rightarrow \|x(t) - x_e\| < \epsilon \quad (2.3)$$

Lyapunov の意味での安定性が成立していても、 $x(t)$ は時間が経っても x_e に収束する保証はないことに注意する。

[局所漸近安定]

システムが Lyapunov の意味で安定で、

$$\exists \delta > 0, \|x_0 - x_e\| < \delta \rightarrow \lim_{t \rightarrow \infty} \|x(t) - x_e\| = 0 \quad (2.4)$$

のとき、平衡点 x_e は局所漸近安定であるという。

この安定性は、状態は最終的には x_e に落ち着くことを保証しているが、どれくらいの早さで x_e に収束するかは一切保証していない。その意味で、あまり実用的でない場合がある。ただし、時不変線形システムの場合、漸近安定性は指数安定性を意味する。

[局所指数安定]

局所漸近安定で、

$$\exists \alpha > 0, c > 0 \rightarrow \|x(t) - x_e\| \leq ce^{-\alpha t} \|x_0 - x_e\| \quad (2.5)$$

上記の条件が全ての初期値 x_0 に対して成り立つ時、それぞれの安定性は大域的安定、大域的漸近安定、大域的指数安定と呼ばれる。

これらの安定性の中で、漸近安定性を判別する方法として、次の定理で与えら得る方法がよく用いられている。

[Lyapunov の安定定理]

(2.1) で与えられるシステムに対して、次の条件を満たすスカラー関数 $V(x)$ が x_e の近傍で存在するとき、 x_e は局所漸近安定である。

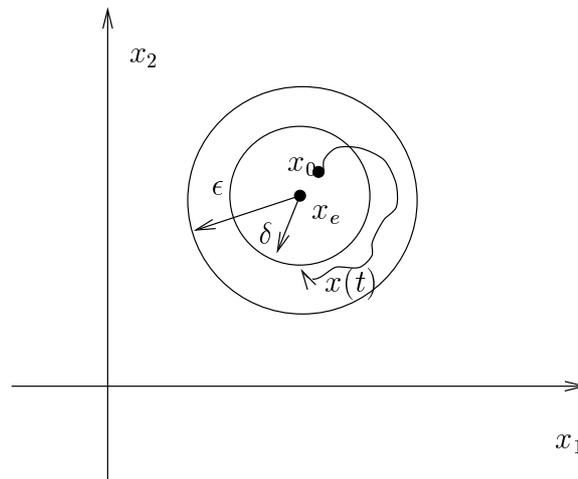


図 2.1: 安定性の定義

1. $V(x) > 0, V(x_e) = 0$
2. $V(x)$ は時間で連続微分可能で、 $\dot{V}(x) < 0$ ($x \neq x_e$)

このような性質を満たす $V(x)$ を Lyapunov 関数という。このような関数の一般的な決定方法は存在しないが、物理的なエネルギーに関するものが Lyapunov 関数として使われる場合が多い。

この定理の応用例として、次のシステムを考える。

$$\dot{x} = -k \sin(x), \quad k > 0 \quad (2.6)$$

このシステムの一つの平衡点は $x_e = 0$ である。この平衡点が漸近安定であるかどうかを調べるために、次の Lyapunov 関数の候補を考える。

$$V(x) = \frac{1}{2}x^2 \quad (2.7)$$

この関数は明らかに $V(x) > 0$ ($-\pi < x < \pi, x \neq 0$) である。また、

$$\dot{V}(x) = x\dot{x} = -kx \sin(x) < 0 \quad (-\pi < x < \pi, x \neq 0) \quad (2.8)$$

となる。従って、この平衡点は局所漸近安定である。

このように、Lyapunov の安定定理を用いると、方程式の解を実際に解くことなく平衡点の安定性が判定できる。

次に、次の微分方程式で表される質量、ダンパ、バネから構成される機械系の安定性を考える。

$$m\ddot{x} + c\dot{x} + kx = 0 \quad (2.9)$$

この 2 次のシステムの極は特性方程式

$$ms^2 + cs + k = 0 \quad (2.10)$$

の解で与えられるが、Routh・Hurwitz の安定性判別より、このシステムが安定であるための必要十分条件はパラメータ m, c, k 全てが正であることである。このことは、普段見かける質量、ダンパ、バネ系が時間が経つと振動を止めることより明かでもある。このシステムの安定性を Lyapunov の安定定理によって判別することを考えよう。このために、このシステムの持っている力学的エネルギーを Lyapunov 関数の候補として考えてみよう。つまり、

$$V(x) = \frac{1}{2}m\dot{x}^2 + \frac{1}{2}kx^2 \quad (2.11)$$

とする。このシステムの状態は $x = [x, \dot{x}]^T$ 、平衡状態は $x_e = [0, 0]^T$ である。従って、 $V(x) > 0$ ($x \neq x_e$) であることは明かである。一方、 $V(x)$ の解軌道に沿った時間変化は

$$\dot{V}(x) = \dot{x}m\ddot{x} + kx\dot{x}$$

であるが、(2.10)の関係式を代入すると

$$\dot{V}(x) = \dot{x}(-c\dot{x} - kx) + kx\dot{x} = -c\dot{x}^2 \leq 0 \quad (2.12)$$

となる。つまり、 $x = [x, 0]^T$ という状態に対しては、常に $\dot{V}(x) = 0$ となり、 $V(x) < 0$ ($x \neq x_e$) という条件を満たさない。このように、物理的直観から導かれた Lyapunov 関数の候補が Lyapunov 関数にならない場合がよくある。このように Lyapunov の安定定理は使いにくい面がある。この点を改善するために、Lyapunov の定理を拡張した LaSalle の定理がある。

[LaSalle の定理]

(2.1) で与えられるシステムに対して、次の条件を満たすスカラー関数 $V(x)$ が x_e の近傍で存在するとき、 x_e は局所漸近安定である。

1. $V(x) > 0, V(x_e) = 0$
2. $V(x)$ は時間で連続微分可能で、 $\dot{V}(x) \leq 0 (x \neq x_e)$
3. $\dot{V}(x) \equiv 0$ を仮定した時、微分方程式の解が $x(t) = x_e$ だけである。

この定理が Lyapunov の安定定理と本質的に異なる点は、 $\dot{V}(x)$ の条件に等号を許している点である。この定理を先程の機械系に適用してみると、2つ目までの条件は満たしているので、最後の条件のみをチェックすればよい。

$$\dot{x}(t) \equiv 0 \quad (2.13)$$

を仮定すると、

$$\ddot{x}(t) \equiv 0 \quad (2.14)$$

となる。さらにこの関係より (2.10) を用いると

$$kx = 0 \quad (2.15)$$

となる。今、 $k > 0$ であるので $x = 0$ となり、 $\dot{V} \equiv 0$ を仮定した際の解は $x(t) = x_e$ となり、他の解は存在しない。従って、LaSalle の定理より x_e は漸近安定であることが分かる。

2.1 時変システムの安定性判別

2.1.1 Lyapunov の安定定理

次に時変システムの安定性についての条件を与えるため、次の時変システムを考える。

$$\dot{x} = f(x, t), \quad x(0) = x_0 \quad (2.16)$$

ただし、

$$f(x_e, t) = 0 \quad (2.17)$$

であるとする。このシステムの安定性を判別する条件を与えるために、次の関数のクラスを定義する。

[クラス K 関数]

R^+ から R^+ へのスカラー関数 α が次の条件を満たすとき、関数 α はクラス K の関数であるという。

1. $\alpha(0) = 0$
2. $\alpha(p) > 0 \quad \forall p > 0$
3. α は非減少関数

この関数を用いて時変系のシステムの Lyapunov の定理は次の定理によって与えられる。

[Lyapunov の安定定理 (時変システム)]

考えているシステムが次の条件を満たす時、 $x_e = 0$ は (局所) 一様漸近安定である。

1. あるクラス K 関数 α, β が存在して $V(x, t)$ が次の条件を満たす。

$$\alpha(\|x\|) \leq V(x, t) \leq \beta(\|x\|) \quad (2.18)$$

2. あるクラス K 関数 γ が存在して、 $\dot{V}(x, t)$ が次の条件を満たす。

$$\dot{V}(x, t) \leq -\gamma(\|x\|) \quad (2.19)$$

ここで注意すべき点はクラス K 関数は引数に対する静的な関数で、時間 t には依存しないことである。時変システムの Lyapunov の安定定理の応用を一般に難しくしているのは、このような時間に依存しない関数を見つけないといけないことである。

2.1.2 Barbalet(バーバラ)の補題

Lyapunov の安定定理は物理的なエネルギーの拡張である Lyapunov 関数の減少によってシステムの安定性を議論するものであった。しかし、実際の応用では Lyapunov 関数を見つけることが容易でない場合も多い。そのような時に、ある関数が収束することを示して、その時間微分が零になることを利用してシステムの安定性を示すことができる。そのために良く使われるのが Barbalet の補題である。この補題はある滑らかな関数が収束すればその微分は零に収束するというものであるが、一般の関数に対してはこのような事が成り立たないので、まずその点を指摘しておく。

次の 3 つの性質については銘記しておく必要がある。時間関数 $f(t)$ に関して次の性質がある。

1. $\dot{f} \rightarrow 0 \not\Rightarrow f$ が $t = \infty$ で収束する。例えば、 $f(t) = \log t$ など。
2. f が $t = \infty$ で収束する $\not\Rightarrow \dot{f} \rightarrow 0$ 例えば、 $f(t) = \exp(-at) \cos(\exp(bt)) (b > a > 0)$ 。
3. $f(t)$ は下に有界で $\dot{f} \leq 0 \Rightarrow f$ は $t = \infty$ で収束する。

Barbalet の補題を説明するために、関数の一様連続性について思い出しておこう。

[関数の一様連続性]

関数 $f(t)$ が次の条件を満たすとき、一様連続であるという。

$$\forall R > 0, \exists \eta(R) > 0, t_1 \geq 0, t_2 \geq 0, |t_1 - t_2| < \eta \Rightarrow |f(t_1) - f(t_2)| < R$$

一様連続性の定義の本質は R が t に依存していないことである。例えば、 $\exp(t)$ は連続ではあるが、一様連続ではない。これらの準備の下で Barbalet の補題はつぎのように与えられる。

[Barbalet の補題]

関数 f が微分可能で、 $t = \infty$ で収束し、 \dot{f} が一様連続であるなら $\dot{f} \rightarrow 0 (t \rightarrow \infty)$ 。

この補題の系としてつぎのものが良く用いられる。

[系]

関数 f が微分可能で、 $t = \infty$ で収束し、 \dot{f} が有界であるなら $\dot{f} \rightarrow 0 (t \rightarrow \infty)$ 。

この系は、微分が有界な関数は必ず一様連続になることより導かれる。

更に信号の入出力特性より導かれる定理として次のものが良く用いられる。表現を簡単にするために、関数の集合を幾つか定義する。

$$L_p(0, \infty) = \left\{ f(t) \left| \int_0^\infty |y(\tau)|^p d\tau < \infty \right. \right\} \quad (2.20)$$

$$L_\infty(0, \infty) = \{ f(t) \mid \text{ess.sup} |f(t)| < \infty \text{ for } t \in (0, \infty) \} \quad (2.21)$$

ただし、積分は Lebesgue(ルベーグ) 積分の意味で考える。また、 ess.sup は可測でない部分を除いた部分での sup であり、連続な関数の場合には sup に一致する。もう一つ必要な定義として、伝達関数に関するものを与える。伝達関数 $H(s)$ が安定で虚軸上に極を持たないときその伝達関数は strictly stable (厳密に安定) であるといい、 $H(\infty) = 0$ のとき、 strictly proper (厳密にプロパー) であるという。

これらの準備の下に関数の収束性は次のように表現できる。

[補題 1]

厳密にプロパーで厳密に安定な伝達関数 $H(s)$ があり、 $u \in L_2(0, \infty)$ である時、

$$Y(s) = H(s)U(s) \quad (2.22)$$

によって y を与えると、 $y \in L_2(0, \infty) \cap L_\infty(0, \infty)$ であり、 $\dot{y} \in L_2(0, \infty)$ 。

これらの関数空間の関係と先程の Barbalet の補題を組み合わせたものとして次のものがある。

[補題 2]

関数 f が $f \in L_2(0, \infty)$ でその微分が $\dot{f} \in L_\infty$ であるとき、 f は一様連続で $f(t) \rightarrow 0(t \rightarrow \infty)$ 。

補題 2 は $g(t) = \int_0^t |f(\tau)|^2 \tau$ なる関数を考えると、これは上に有界であり、単調増加関数であるので収束する。さらにその微分が $\dot{g} \in L_\infty$ より一様連続であるので、Barbalet の補題を用いて $\dot{g} \rightarrow 0$ が言えるのである。

これらの補題を用いることによって、 $f(t)$ として状態のノルムを取ることによって漸近安定性を議論することが可能になる。

第 3 章

軌道追従制御

ロボットアームによって作業をさせようとした場合、手先の座標を制御する必要がある。この制御の方法に

1. PTP(Point To Point) 制御
2. CP (Continuous Path) 制御

がある。PTP 制御は障害物ないような作業空間において、手先をある位置から他の位置に移動させる制御で、途中の軌道は考慮しない制御である。CP 制御は空間内に定められた連続な軌道を追従させる制御であり、複雑な環境下での作業では必ず必要となる制御である。

また、これらの制御を実現するために、フィードバック系をどの座標系で構成するかによっても制御は以下のように分類される。

1. 関節空間でのフィードバック
2. 作業空間でのフィードバック

関節空間でのフィードバックでは、まず、手先の目標の位置 x_d を関節での目標角度 q_d に逆運動学によって変換する。そして、現在の q を観測することによってフィードバック系を構成するものである。一方、作業空間でのフィードバックでは、現在の手先の位置 x を計測、または q より順運動学によりもとめ x_d との偏差により直接フィードバック系を構成するものである。しかし、これらの制御系は各々長所と短所を持っている。関節空間でのフィードバックの制御の利点は

- 実時間での計算量が少ない。
- 関節にモデル化していない弾性特性があっても不安定になりにくい。
- 特異姿勢の影響を考慮しなくても良い。

欠点は

- 陽に逆運動学を解く必要がある。

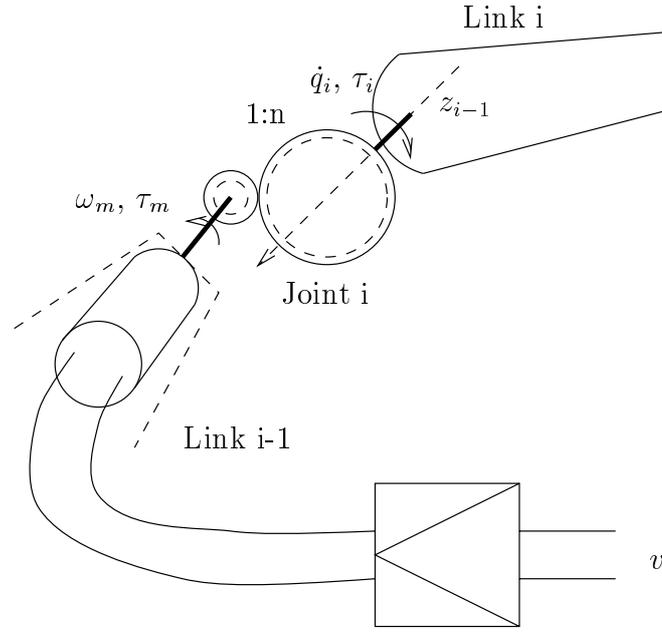
一方、作業空間でのフィードバックの特徴は関節空間での特徴を逆にしたものとなる。ここでは、最初に関節空間における PTP 制御を説明した後、作業空間での CP 制御について説明する。

3.1 関節独立制御

今、手先の目標位置 x_d に対応する関節角 q_d が与えられているとし、全ての関節が回転関節で、各々の関節がモータによって駆動されているとする。リンク $i - 1$ とリンク i を考えて、リンク $i - 1$ にモータステータが固定され、モータのロータが減速器によってリンク i につながれているとする。(3.1) (モータの回転とトルクの方に注意。)

モータの回転速度を ω_m 、慣性モーメントを J_m 、粘性を μ_m 、トルク定数を K とすると直流モータの運動方程式は次のようになる。

$$J_m \dot{\omega}_m + \mu_m \omega_m = K v - \tau_i \quad (3.1)$$

図 3.1: リンク i を駆動するモータ

一方、ギア比を 1対 n とすると

$$\omega_m = n\dot{q}_i \quad (3.2)$$

$$n\tau_m = \tau_i \quad (3.3)$$

であるから、この関係を (3.1) に代入して、 \dot{q}_i の方程式に直すと

$$n^2 J_m \ddot{q} + n^2 \mu_m \dot{q} = nKv - \tau_i \quad (3.4)$$

となる。モーター系を考慮しないマニピュレータの運動方程式は

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau$$

であり、 i 行目の方程式を取り出すと

$$\sum_j d_{ij} \ddot{q}_j + h_i(q, \dot{q}) = \tau_i \quad (3.5)$$

となる。ただし、 $h_i(q, \dot{q})$ はコリオリ、中心力、重力項を全て足し合わせた項である。ここで、

$$f_i(q, \dot{q}, \ddot{q}) = \sum_{j=1, j \neq i} d_{ij} \ddot{q}_j + h_i(q, \dot{q}) \quad (3.6)$$

と定義し、(3.4) を (3.5) に代入すると

$$(d_{ii}(q) + n^2 J_m) \ddot{q}_i + n^2 \mu_m \dot{q}_i + f_i(q, \dot{q}, \ddot{q}) = nKv \quad (3.7)$$

となる。今、適当な定数の姿勢 $\bar{q}(q$ のノミナル値) を選んで

$$\bar{d}_{ii} = d_{ii}(\bar{q}) \quad (3.8)$$

を一定の値とするか、 n が産業用ロボットのような場合数十から 100 の値をとるので、 $d_{ii}(q)$ の方を無視して $\bar{d}_{ii} = 0$ とすると、方程式は最終的に

$$(\bar{d}_{ii} + n^2 J_m) \ddot{q}_i + n^2 \mu_m \dot{q}_i = \tau_i - f_i(q, \dot{q}, \ddot{q}) \quad (3.9)$$

となる。 f_i の項を外乱と見なして、系を伝達関数で表すと次のようになる。

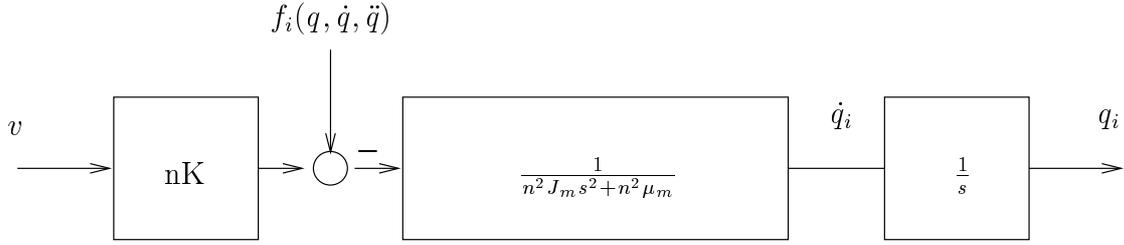


図 3.2: 関節の動特性の伝達関数表現

この伝達関数は、入力に外乱 f_i を持つ、トルク定数 nK 、慣性モーメント $\bar{d}_{ii} + n^2 J_m$ 、粘性摩擦係数 $n^2 \mu$ を持つモータと同じものである。従って、この系に対してモータのサーボ系を組めばよいことになる。 $(f_i$ には重力項が含まれているので、1型のサーボ系にしないと定常偏差が残ることになる。)しかし、外乱と考えている f_i は実際には外生信号ではなく、 (q, \dot{q}, \ddot{q}) の関数であるので、 q が \bar{q} から大きくずれた場合には、応答特性が著しく悪化したり、最悪の場合は不安定になってしまう場合がある。つまり、このように設計した制御系は局所的にしか安定性が保証されないのである。これに対して、以降では大域的に安定性が保証される制御系について考える。

3.2 重力補償付き PD 制御

前節で説明した安定定理を利用すると、非常に簡単な制御で大域的に位置の偏差がゼロになる PTP 制御系を導出することができる。今初期角度が q_0 で速度がゼロの状態から角度 q_d で、速度がゼロの状態にマニピュレータを制御したいとする。この時、制御入力 τ を次のように決定する。

$$\tau = -K_d \dot{q}(t) - K_p(q(t) - q_d) + G(q) \quad (3.10)$$

ただし、 K_d, K_p は正定対称な行列とする。この時、システムの状態を $x = [q^T, \dot{q}^T]^T$ とすると、 x は望ましい状態 $x_d = [q_d^T, 0^T]^T$ に漸近収束する。証明は以下の様に、LaSalle の定理を用いて行われる。

制御入力 (3.10) をマニピュレータの運動方程式に代入すると

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + K_d\dot{q} + K_p(q - q_d) = 0 \quad (3.11)$$

ここで、 q_d が定数であったことを思い出すと、 $e := q - q_d$ とすると $\dot{q} = \dot{e}$, $\ddot{e} = \ddot{q}$ となる。この関係を上の式に代入して、 (e, \dot{e}) の方程式にすると

$$D(e + q_d)\ddot{e} + C(e + q_d, \dot{e})\dot{e} + K_d\dot{e} + K_p e = 0 \quad (3.12)$$

となる。今、式の中には q_d が入っているが、 q_d が定数であるため、この微分方程式は状態を (e, \dot{e}) とする時不変な非線形システムとなる。このシステムに対して、 $(0, 0)$ が漸近安定であることを示せば、目的を達成することを示せる。このために、次の Lyapunov 関数の候補を考える。

$$V(e, \dot{e}) = \frac{1}{2} \dot{e}^T D(q) \dot{e} + \frac{1}{2} e^T K_p e \quad (3.13)$$

運動方程式の性質より、 $D(q)$ は正定対象であるのであり、 K_p の条件から $V(e, \dot{e}) > 0$ ($e, \dot{e} \neq (0, 0)$) である。次に V の時間微分を考えると、

$$\dot{V} = \dot{e}^T D(q) \ddot{q} + \frac{1}{2} \dot{e}^T \dot{D}(q) \dot{e} + e^T K_p \dot{e} \quad (3.14)$$

ここで、閉ループシステムの運動方程式の関係 (3.12) を代入すると

$$\dot{V} = \dot{e}^T (-C(e + q_d, \dot{e})\dot{e} - K_d\dot{e} - K_p e) + \frac{1}{2} \dot{e}^T \dot{D}(q) \dot{e} + e^T K_p \dot{e} \quad (3.15)$$

となる。消去出来る項を消して整理すると、

$$\dot{V} = \frac{1}{2}\dot{e}^T(\dot{D}(e+q_d) - 2C(e+q_d, \dot{e}))\dot{e} - \dot{e}^T K_d \dot{e} \quad (3.16)$$

ところが、 $\dot{D}(e+q_d) - 2C(e+q_d, \dot{e}) = \dot{D}(q) - 2C(q, \dot{q})$ であり、これは歪み対称であるのでこの二次形式はゼロとなる。結局

$$\dot{V} = -\dot{e}^T K_d \dot{e} \leq 0 \quad (3.17)$$

となる。これで LaSalle の定理の 2 番目までの条件が成立したことになる。次に最後の条件が成立していることを見るために

$$\dot{V} = -\dot{e}^T K_d \dot{e} \equiv 0 \quad (3.18)$$

を仮定する。これと、 K_d の仮定より $\dot{q} \equiv 0$ となり、 $\ddot{q} \equiv 0$ となる。この関係を (3.12) に代入すると

$$K_p e = 0 \quad (3.19)$$

となり、 K_p の仮定より $e = 0$ となる。これは $(e, \dot{e}) = (0, 0)$ が $\dot{V} \equiv 0$ を仮定した際の (3.12) の唯一解であることを示している。よって $(e, \dot{e}) = (0, 0)$ は漸近安定である。この性質は初期値の取り方によらないから、大域的漸近安定である。ここで、この制御系は K_d, K_p を対角行列に取ると一見関節毎の独立制御になるが、重力補償項を計算する際に、全体の姿勢 q を必要とするので関節独立制御にはなっていないことに注意する。

3.3 分解速度制御

上で説明した方法は、望ましい手先の位置 $x_d(t)$ が与えられた場合、これを逆運動学問題を解いて $q_d(t)$ に変換して関節角度でのフィードバック系を構成する方法であった。しかし、逆運動学問題は一般に解析的に解くことはできず、またあらかじめ $q_d(t)$ を計算しておくことは経済的でない。また、実用的なマニピュレータシステムでは、位置やトルクではなく速度サーボ系が組まれていることが多く、速度レベルで制御することが容易になっている場合が多い。そのような場合に有効な方法として分解速度制御がある。(ただし、この方法では、任意の時間目標値への大域的漸近収束性を示すことはできない。)

まず、逆運動学の数値的解法から考えよう。今多変数の座標変換関数 f が

$$x = f(q) \quad (3.20)$$

によって与えられているとしよう。この関数が与えられるとき、 x_d から q_d を求めるのが逆運動学問題であった。今、特殊な姿勢として x_s に対する q_s が与えられているとしよう。つまり、

$$x_s = f(q_s) \quad (3.21)$$

ここで、 x_s の微小変化として δx を考えると

$$x_s + \delta x = f(q_s + \delta q) \quad (3.22)$$

が微小な δq に対して成り立つ。今、 f が解析的な関数であるとすると、

$$x_s + \delta x \approx f(q_s) + J(q) \delta q, \quad J := \frac{\partial f}{\partial q} \quad (3.23)$$

が成り立つ。したがって、 J が正則であると仮定すると

$$\delta q = J^{-1} \delta x \quad (3.24)$$

となり、手先での移動量 δx に対応する関節での移動量が計算できる。したがって、 x_d に対して微小な偏差 δx が生じた場合には、 $-J^{-1} \delta x$ だけ関節で移動させてやればよい。一方、手先に初期偏差がなく、

$$\dot{q}_d = J^{-1}(q_d) \dot{x}_d \quad (3.25)$$

で、 $\dot{q} = \dot{q}_d$ が実現されていれば

$$\dot{x} = J(q_d)J^{-d}\dot{x}_d, \quad x(t) = x_s + \int_0^t \dot{x}_d d\tau \quad (3.26)$$

となり、なり完全に目標値に一致する。これらのことより、分解速度制御では関節での目標速度 \dot{q}_d を

$$\dot{q}_d = J^{-1}(q)(\dot{x}_d - K_p(x - x_d)) \quad (3.27)$$

により与える。この目標値に対して、 \dot{q} が正確に追従すれば

$$\dot{q} \approx J^{-1}(q)(\dot{x}_d - K_p(x - x_d)) \quad (3.28)$$

となり、

$$\dot{x} = \dot{x}_d - K_p(x - x_d) \quad (3.29)$$

となって、 $e := x - x_d$ とすると

$$\dot{e} + K_p e = 0 \quad (3.30)$$

が近似的に成立し、 K_p を正定に取っておけば指数的に偏差は収束することになる。このように、望ましい運動を速度のレベルに落して考えるのが分解速度制御である。しかし、今述べて来た方法は近似的にしか成り立たないため、大域的な指数安定性を示すことは難しい。

3.4 計算トルク法 (分解加速度制御)

前節で説明した重力補償付き PD 制御は、基本的に PTP 制御であり、CP 制御には向かない。また、漸近収束は保証されているものの、動きはじめなどの各軸の干渉が残っており、挙動の予測が非常に難しい。また、前節で説明した分解速度制御では、大域的な安定性を厳密な意味で示すことはできない。ここでは、分解速度制御の考え方を加速度のレベルに拡張し、各軸の特性の干渉をなくして、基本的に線形システム制御理論を適用できるように新しい入力から状態までの関係が線形になるような制御則を説明する。

計算トルク法の制御入力は次のようになる。

$$\tau = D(q)v + C(q, \dot{q})\dot{q} + G(q) \quad (3.31)$$

ただし、 v は新しい入力で、加速度と同じ次元を持つ信号である。この入力をシステムに入れると、

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = D(q)v + C(q, \dot{q})\dot{q} + G(q) \quad (3.32)$$

であり、 $D(q)$ が可逆であるため、

$$\ddot{q} = v \quad (q_i = v_i, i = 1, \dots, n) \quad (3.33)$$

となり、加速度を直接制御することになる。または、 v から q が二重積分器の伝達関数となっていると解釈することもできる。(図 3.3 参照。) このシステムに対して、 q が q_d に追従する制御系を設計すれば良いので、

$$e := q - q_d, \quad (e_i = q_i - q_{di}) \quad (3.34)$$

として、

$$v = \ddot{q}_d - K_d \dot{e} - K_p e \quad (3.35)$$

とする。ただし、 K_d, K_p は正定対称なフィードバックゲインである。(3.33) に上の式を代入すると

$$\ddot{e} + K_d \dot{e} + K_p e = 0 \quad (3.36)$$

となる。このシステムは指数安定であるが、漸近安定性は

$$V = \frac{1}{2} \dot{e}^T \dot{e} + \frac{1}{2} e^T K_p e \quad (3.37)$$

を Lyapunov 関数として前の節と同様に示すことができる。この方法では、初期の状態の誤差がゼロの場合、偏差は恒等的にゼロとなり、制御入力は完全な追従に必要なトルクを計算していることになる。このため、この方法は計算

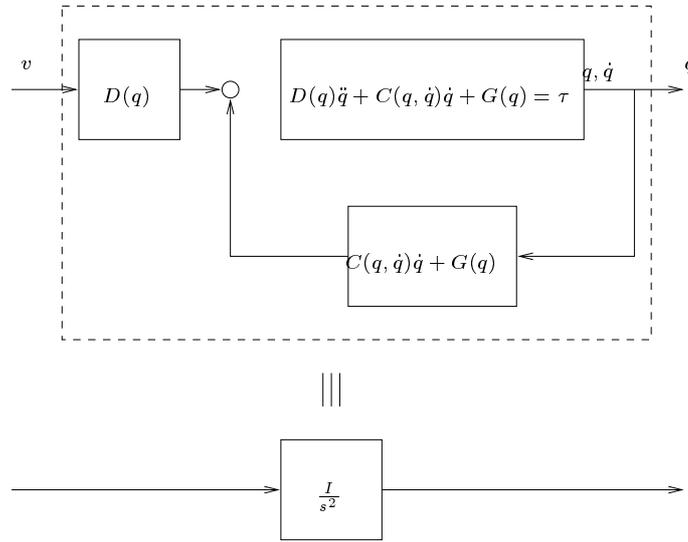


図 3.3: 新しい入力から見た等価伝達関数

トルク法と呼ばれる。また、必要な信号を加速度のレベルで考えて制御するので、分解加速度制御と呼ばれることもある。また、

$$K_d = \text{diag}(k_{d1}, \dots, k_{dn}), K_p = \text{diag}(k_{p1}, \dots, k_{pn}) \quad (3.38)$$

とすると各 e_i に対して、

$$\ddot{e}_i + k_{di}\dot{e}_i + k_{pi}e_i = 0 \quad (3.39)$$

となり、各軸の運動が他の軸とは完全に非干渉化される。しかし、実際には、モデル化誤差が存在して、制御入力

$$\tau = \hat{D}(q)v + \hat{C}(q, \dot{q})\dot{q} + \hat{G}(q) \quad (3.40)$$

としか計算することができない。ここで、 $\hat{\cdot}$ の付いている項はその関数が同定または数値計算により決定されたパラメータを用いて計算されていることを示している。この誤差を吸収するための一つの方法としては、(3.35)の制御則に積分動作を付加して、目標値の加速度項と偏差のPID項より構成する方法である。

$$v = \ddot{q}_d - K_d\dot{e} - K_p e - K_i \int e(\tau)d\tau \quad (3.41)$$

K_d, K_p, K_i の決定法は

$$\frac{d}{dt} \begin{bmatrix} \int e d\tau \\ e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} 0 & I & 0 \\ 0 & 0 & I \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \int e d\tau \\ e \\ \dot{e} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix} v \quad (3.42)$$

に対して極配置問題や最適レギュレータ問題を解けばよい。

次に、同様な考え方で作業空間でのフィードバックを考える。そのために、まず運動方程式を座標変換する。今、関節での座標と作業空間での座標が関係づけられているとする。

$$x = f(q) \quad (3.43)$$

$$\dot{x} = J\dot{q} \quad (3.44)$$

$$\ddot{x} = J\ddot{q} + \dot{J}\dot{q} \quad (3.45)$$

これより、 $J(q)$ の逆行列が存在すると仮定し、

$$\dot{q} = J^{-1}\dot{x} \quad (3.46)$$

$$\ddot{q} = J^{-1}(\ddot{x} - \dot{J}\dot{q}) \quad (3.47)$$

として、運動方程式に代入すると

$$J^{-T}D(q)J^{-1}\ddot{x} + J^{-T}(C(q, \dot{q}) - J^{-T}D(q)\dot{J})J^{-1}\dot{x} + J^{-T}G(q) = J^{-T}\tau \quad (3.48)$$

となる。ここで、形式的に

$$D_x(x) := J^{-T}D(q)J^{-1} \quad (3.49)$$

$$C_x(x, \dot{x}) := J^{-T}(C(q, \dot{q}) - J^{-T}D(q)\dot{J})J^{-1} \quad (3.50)$$

$$G_x(x) := J^{-T}G(q) \quad (3.51)$$

$$\tau_x := J^{-T}\tau \quad (3.52)$$

と定義すると、方程式は

$$D_x(x)\ddot{x} + C_x(x, \dot{x})\dot{x} + G_x(x) = \tau_x \quad (3.53)$$

となり、関節の座標で表現した運動方程式と同じ形になる。したがって、作業座標系での目標位置を x_d とし、

$$e_x := x - x_d \quad (3.54)$$

で定義すると、制御入力を

$$\tau_x = D_x(x)v_x + C_x(x, \dot{x})\dot{x} + G_x(x) \quad (3.55)$$

$$v_x = \ddot{x}_d - K_{dx}\dot{e}_x - K_p e_x \quad (3.56)$$

とすれば良いことがわかる。ここで重要なことは、この制御入力を計算するためには、 x_d から q_d を求めるという逆運動学を解く必要がないということである。また、この変換は座標変換であるので、 $D_x(x)$ の正定や、 $\dot{M}_x - 2C_x$ が歪み対称であるような性質は保存される。(示してみよ。)

3.5 スライディングモード制御

前の節では、システムのパラメータが相当な精度で分っている時に非常に有効な制御則について述べたが、実際にはシステムのパラメータが十分な精度で測定されているとは限らない。このようなシステムに対して、入力の変幅が十分に大きい時に、パラメータに大きな誤差が含まれていても良い制御性能が得られる方法について述べる。

一般に、システムの状態が観測できるときに、状態に応じて制御則を切り替える制御を可変構造制御 (Variable Structure Control:VSC) という。このような制御を用いると、一つのベクトル場だけでは不安定なシステムでも、不安定な2つのベクトル場を状態に応じて切り替えることにより安定なシステムを得ることができることがある。(図3.4参照。)この制御一つにスライディングモード (Sliding Mode Control:SMC) 制御があり、これはシステムの状態を状態空間のなかに設定された超平面に拘束するように、状態の拘束面に対する相対位置関係によって制御則を切り替えるものである。

制御則を導出するために、次の仮定を置く。全ての関節が回転関節で次の不等式を満たす定数 $\bar{\lambda}, \underline{\lambda}$ の値が分かっているとす。

$$\bar{\lambda} \geq \sup_q \lambda(M(q)) \quad (3.57)$$

$$\underline{\lambda} \leq \inf_q \lambda(M(q)) \quad (3.58)$$

ただし、 $\lambda()$ は行列の固有値を表すものとする。

ここで制御則を次のようにする。

$$\tau = \hat{C}(q, \dot{q})\dot{q} + \hat{G}(q) - k(q, \dot{q}, \ddot{q}_d)Sgn(S) \quad (3.59)$$

ただし、

$$Sgn(S) = \begin{cases} \frac{S}{\|S\|} & \|S\| \neq 0 \\ 0 & \|S\| = 0 \end{cases} \quad (3.60)$$

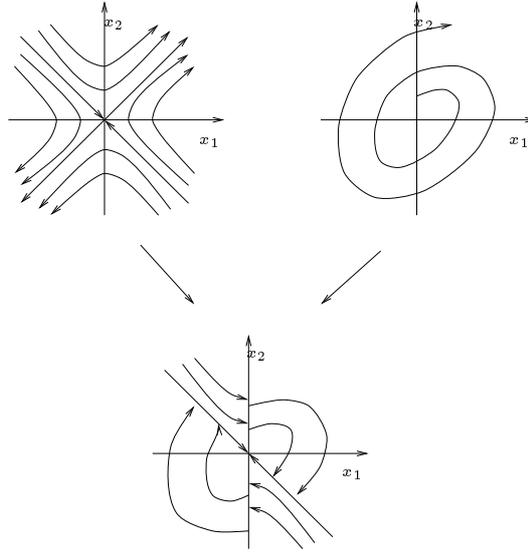


図 3.4: 可変構造制御による漸近安定化

で、 Λ を正定行列として S は次式で定義される。

$$S(e, \dot{e}) := \dot{e} + \Lambda e \quad (3.61)$$

また、次の不等式を満足する誤差の評価関数 $h(q, \dot{q})$ が分っているとし、

$$\|C(q, \dot{q})\dot{q} + G(q) - \hat{C}(q, \dot{q})\dot{q} + \hat{G}(q)\| =: \|\tilde{h}\| \leq \bar{h}(q, \dot{q}) \quad (3.62)$$

ゲイン $k(\cdot)$ は

$$k(q, \dot{q}, \dot{q}_d, \ddot{q}_d) = \bar{\lambda} \left(\epsilon + \frac{\bar{h}(q, \dot{q})}{\underline{\lambda}} + \|\ddot{q}_d\| + \|\Lambda \dot{e}\| \right) \quad (3.63)$$

で定義する。

この制御則の収束性を見るために、次の指標関数 $V(S)$ を考える。(Lyapunov 関数でないことに注意。)

$$V(S) := \frac{1}{2} \|S\|^2 = \frac{1}{2} S^T S \quad (3.64)$$

この関数の時間微分を取ると、

$$\dot{V} = \|S\| \|\dot{S}\| = S^T (D^{-1}(q)(u - \tilde{h}) - \ddot{q}_d + \Lambda \dot{e}) \quad (3.65)$$

となる。右辺をノルムで評価すると、

$$\dot{V} = -k S^T D^{-1} S / \|S\| + S^T (D^{-1} \tilde{h} - \ddot{q}_d + \Lambda \dot{e}) \quad (3.66)$$

$$\geq -k \frac{\|S\|}{\bar{\lambda}} + \|S\| \left(\frac{\bar{h}}{\underline{\lambda}} + \|\ddot{q}_d\| + \|\Lambda \dot{e}\| \right) \quad (3.67)$$

$$= -\epsilon \|S\| \quad (3.68)$$

となる。 $\|S\| \neq 0$ として両辺を $\|S\|$ で割ると

$$\|S\| \leq -\epsilon \quad (3.69)$$

したがって、両辺を積分すると

$$\|S\| \leq -\epsilon t + \|S\|(0) \quad (3.70)$$

したがって、 $t \geq \|S\|(0)/\epsilon$ で $\|S\|$ はゼロとなり、それ以後はゼロとなる。したがって、 $\|S\| = 0$ の状態では

$$\dot{e} = -\Lambda e \quad (3.71)$$

であるから、指数的に ϵ はゼロになる。この性質は対象のダイナミクスにあまり依存せず成り立つことに注意しよう。以上の議論は形式的なもので、微分方程式の右辺が局所連続的に切り替わらないなどの問題で数学的には厳密な取り扱いを要求される。しかし、本質的には上の議論が当てはまる。さらに、実際の応用では、時間ゼロで値が切り替わるような $S/\|S\|$ が構成できなため、チャタリングと呼ばれる振動を励起してしまう。このために、 $S/\|S\|$ を飽和関数である

$$\text{Sat}(S) = \frac{S}{\|S\| + \epsilon} \quad (3.72)$$

とすることも多い。

第 4 章

外界との干渉を陽に考慮した制御系

4.1 拘束のあるシステムの運動方程式

今一般化ベクトル $q \in R^n$ を用いて、拘束の式が

$$c_i(q) = 0, \quad (i = 1, \dots, m) \quad (4.1)$$

の用に表されているとする。これを簡単にベクトル表現して

$$C(q) = 0, C \in R^m \quad (4.2)$$

更に、幾何学的な拘束を受けないシステムの運動方程式が

$$D(q)\dot{q} + h(q) = \tau \quad (4.3)$$

によって表現されているとしよう。そうすると、拘束条件の式 (4.2) を微分して、

$$\dot{C}(q) = J_c \dot{q} = 0 \quad (4.4)$$

ただし、

$$J_c := \frac{\partial C}{\partial q} \quad (4.5)$$

ここで、 J_c の各行ベクトルは \dot{q} に直交していることを示している。これは、各行が q の拘束面の法線を表していることを示している。

(4.3) では拘束の条件を無視しているが、実際には拘束からくる力が右辺に加わり、その一般化力を τ_c とすると、(4.3) は

$$D(q)\dot{q} + h(q) = \tau + \tau_c \quad (4.6)$$

となる。しかし、この拘束力は仕事をしないので、

$$\tau_c^T \dot{q} = 0 \quad (4.7)$$

でなければならない。ところが、 \dot{q} と直交するベクトルは (4.5) によって与えられているので、結局拘束力は

$$\tau_c = -J_c^T \lambda, \quad \lambda \in R^m \quad (4.8)$$

によって与えられる事になる。ここで与えられた拘束力は (4.5) をもう一度時間で微分して、

$$J_c \ddot{q} + \dot{J}_c \dot{q} = 0 \quad (4.9)$$

を得、これと (4.6) を連立して、

$$\begin{bmatrix} D(q) & J_c^T(q) \\ J_c(q) & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ \lambda \end{bmatrix} = \begin{bmatrix} \tau - h(q, \dot{q}) \\ -\dot{J}_c \dot{q} \end{bmatrix} \quad (4.10)$$

によって与えられる。これより、 \ddot{q} , λ が一意に定まるためには、(4.10) の左辺の行列が正則である必要があることが分かる。

4.2 力と位置のハイブリッド制御

いま、ロボットにヤスリを持たせて、金属表面を磨くような作業を考えよう。この時、ヤスリは表面に対して、適当な押しつけ力を保ちながら、適当な速度で表面に沿って移動しなければならない。このように、ある方向には位置や速度、それ以外の方向に力を制御する方法を力と位置のハイブリッド制御と呼ぶ。

今、拘束したい超平面の方程式がデカルト座標 x を用いて次の様にと与えられているとする。

$$C(x) = 0 \quad (4.11)$$

これより、 x の接平面の方程式は

$$\frac{\partial C}{\partial x} \dot{x} = 0 \quad (4.12)$$

となる。ここで、各行は平面に対する法線ベクトルとなっており、発生できる力の基底ベクトルとなっていることに注意する。更に、ここでは E_f の各行はその長さが 1 に規格化されているとする。これを明示的に表現するために、

$$E_f = \frac{\partial C}{\partial x} \in R^{m \times n} \quad (4.13)$$

とする。ここで、 $C(x)$ には冗長な拘束がないように、 E_f は行フルランクであるとする。一方、位置あるいは、速度を制御できるのはこの補空間になるので、その規格化された基底ベクトルを横に並べたものを E_p とする。ただし、

$$\text{rank} \begin{bmatrix} E_p \\ E_f \end{bmatrix} = n \quad (4.14)$$

となるように E_p を選ぶ。

今、一般化座標 q とデカルト座標 x が

$$x = f(q) \quad (4.15)$$

また、速度関係が

$$\dot{x} = J\dot{q}, \quad J := \frac{\partial f}{\partial q} \quad (4.16)$$

として与えられているとする。今、ロボットが拘束を満たして運動しているとする、

$$\dot{C}(x) = 0 \quad \rightarrow \quad E_f J \dot{q} = 0 \quad (4.17)$$

となり、前節の議論よりロボットの運動方程式は

$$D(q)\ddot{q} + h(q, \dot{q}) = \tau - J^T E_f^T \lambda \quad (4.18)$$

となる。

一方、 E_p の各行は規格化された基底ベクトルであるから、 \dot{x} と各行の内積はその基底の成分、つまりその方向の速度となる。つまり、移動方向の速度ベクトルを \dot{x}_p とすると、

$$\dot{x}_p = E_p \dot{x} \quad (4.19)$$

と表現できる。これと拘束の式を合わせると、

$$E\dot{x} = \begin{bmatrix} \dot{x}_p \\ 0 \end{bmatrix}, \quad E := \begin{bmatrix} E_p \\ E_f \end{bmatrix} \quad (4.20)$$

を得る。更に、この式を微分して、

$$E\ddot{x} + \dot{E}\dot{x} = \begin{bmatrix} I \\ 0 \end{bmatrix} \ddot{x}_p \quad (4.21)$$

を得、(4.16) を微分して

$$\ddot{x} = J\ddot{q} + \dot{J}\dot{q} \quad (4.22)$$

となる。この関係と運動方程式の関係を用いると、(4.21) は

$$\begin{bmatrix} I \\ 0 \end{bmatrix} \ddot{x}_p = EJ^{-1}(D^{-1}(\tau - h - J^T E_f^T \lambda) - \dot{J}\dot{q}) + \dot{E}J\dot{q} \quad (4.23)$$

ここで、ハイブリッド制御系を構成するのに、力と位置が見かけ上別々のループで非干渉な制御ができることが望ましい。このために、加速度と力に対応する新しい入力 u_1 と u_2 を考えて、

$$\ddot{x}_p = u_1 \quad (4.24)$$

$$\lambda = u_2 \quad (4.25)$$

となるように非干渉線形化することを考える。このための必要条件として、この関係を (4.23) に代入して、

$$\begin{bmatrix} I \\ 0 \end{bmatrix} u_1 = EJ^{-1}(D^{-1}(\tau - h - J^T E_f^T u_2) - \dot{J}\dot{q}) + \dot{E}J\dot{q} \quad (4.26)$$

を得る。この式は τ について一意に解けて、

$$\tau = D \left\{ JE^{-1} \left(\begin{bmatrix} I \\ 0 \end{bmatrix} u_1 - \dot{E}J\dot{q} \right) + h + J^T E_f^T u_2 \right\} \quad (4.27)$$

となる。

次に、この入力 u_1 が十分であることを示す。この入力を (4.23) に代入して、整理すると

$$\begin{bmatrix} I \\ 0 \end{bmatrix} \ddot{x}_p = \begin{bmatrix} I \\ 0 \end{bmatrix} u_1 + EJD^{-1}J^T E_f^T (u_2 - \lambda) \quad (4.28)$$

を得る。この式に左から、 $\begin{bmatrix} 0 & I \end{bmatrix}$ を掛けると、

$$0 = E_f J D^{-1} J^T E_f^T (u_2 - \lambda) \quad (4.29)$$

となり、 E_f が行フルランクであるから、 $E_f J D^{-1} J^T E_f^T$ は正則となり、結局 $u_2 = \lambda$ となる。さらに、この関係を (4.28) に代入すると、 $\ddot{x}_p = u_1$ を得る。従って、位置 x_p の目標値を x_d 、その偏差を $e_p := x_p - x_d$ とし、力 λ の目標値を λ_d 、その偏差を $e_f := \lambda - \lambda_d$ とすると、

$$u_1 = \ddot{x}_d - K_1 e_p - K_2 \dot{e}_p, \quad K_1 > 0, \quad K_2 > 0 \quad (4.30)$$

$$u_2 = \lambda_d - K_3 \int_0^t e_f(\tau) d\tau, \quad K_3 > 0 \quad (4.31)$$

とすれば良いことが分かる。

4.2.1 メカニカル・インピーダンス制御

力と位置のハイブリッド制御は、環境の幾何学的パラメータが正確に分っている時には、非常に良い制御結果を与えるが、環境に不確定性がある場合には環境を破壊したり、自分自身を壊してしまう恐れがある。これは、軌道を指定している方向に障害物が有り、積分制御によりオフセットのない制御を使っていた場合には、障害物を無理矢理突き抜けて進もうとするために無理な力を発生する可能性があるためである。これを避けるには、障害物に接触した時に、障害物に大きな力を加えることなく、柔軟に軌道を変更して回避動作または軌道を停止する機能が必要となる。また、人間と直接接するようなサービスをするロボットなどは、この機能が必須の条件となる。このような機能を実現する制御法の一つがメカニカル・インピーダンス制御である。電気回路のインピーダンス特性は、与えられる電圧に対して流れる電流がどのようになるかの特性を与えるが、機械系ではこのアナロジーとして、与えられる力に対して機械的な速度がどのようになるかを表すものがメカニカル・インピーダンスと呼ばれている。メカニカル・インピーダンス制御は、与えられたメカニカル・インピーダンス特性を、与えられた座標系で制御対象が持つように制御するモデルマッチング制御、モデル追従制御具体例であると言える。

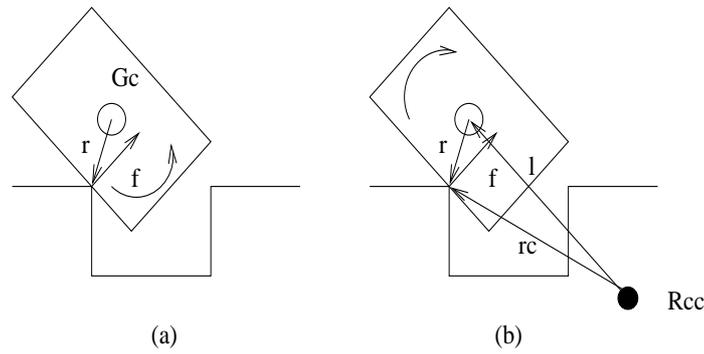


図 4.1: RCC の原理

モデルマッチングに基づく制御則

今、外力を作業座標系で F_e として表し、その外力が働く点までのヤコビアンを J とすと、運動方程式は

$$D(q)\dot{q} + h(q) = \tau - J^T F_e \quad (4.32)$$

となる。又、この運動方程式を作業座標系に座標変換すると

$$J^{-T}D(q)J^{-1}\ddot{x} + J^{-T}(C(q, \dot{q}) - J^{-T}D(q)\dot{J})J^{-1}\dot{x} + J^{-T}G(q) = J^{-T}\tau - F_e \quad (4.33)$$

となる。今、 F_e に対して、 x, \dot{x} の運動が次の運動方程式 (メカニカル・インピーダンス) により表されているとする。

$$M_m\ddot{e} + D_m\dot{e} + K_m e = F_e \quad (4.34)$$

ただし、 $e = x - x_d$ で x_d は最初に与えられている目標軌道である。今、 F_e が観測出来るとして、分解加速度制御が使えるとする。式 (4.34) が成り立っているとすると、加速度 \ddot{q} がどのような式を満たさないといけないかを計算すると、

$$\ddot{x} = M_m^{-1}(\ddot{x}_d - D_m\dot{e} - K_m e + F_e) \quad (4.35)$$

となる。したがって、今決定しなければならないのは制御入力 τ であるので、この関係を式 (4.33) に代入して τ を求めると、

$$\tau = J \left(J^{-T}D(q)J^{-1}M_m^{-1}(\ddot{x}_d - D_m\dot{e} - K_m e + F_e) + J^{-T}(C(q, \dot{q}) - J^{-T}D(q)\dot{J})J^{-1}\dot{x} + J^{-T}G(q) + F_e \right) \quad (4.36)$$

となる。これは、外力 F_e を入力で一旦キャンセルし、分解加速度制御の望ましい加速度を $M_m^{-1}(\ddot{x}_d - D_m\dot{e} - K_m e + F_e)$ としたものに等しいことが分る。しかし、この構成法は対象のプラントのダイナミクスを望ましいものに完全に一致させるように制御するもの (完全モデルマッチング制御) であるので、パラメータに誤差がある場合には、モデルの応答と実際のものとの間にオフセットを生じてしまう可能性がある。

4.2.2 遠隔コンプライアン中心

今、図 4.1(a) のように、棒を穴に挿入しようとする作業をロボットにさせることを考える。この際、棒の側面が穴のエッジに接触すると、図のような回転力が生じてしまい、うまく棒を穴に挿入することができない。もし、重心を仮想的に (b) の Rcc の点に移動することができれば、自然と姿勢を立て直し簡単に棒を穴に挿入することができる。このように仮想的に考えた重心のことを遠隔コンプライアン中心 (RCC: Remote Compliance Center) という。前節のインピーダンスモデルでは、運動の中心をどのように設定するかは言及しなかったが、インピーダンスの中心は

このようにタスクに応じて設定する必要がある。ここでは、力センサーがあるとして、制御によって RCC を実現する方法について説明する。

図 (a) の元の棒の運動方程式は、棒を剛体とみなすと次の方程式で与えられる。

$$M\dot{v} + G = f_c + f \quad (4.37)$$

$$I\dot{\omega} + \omega \times I\omega = \tau_c + r \times f \quad (4.38)$$

ただし、 v, ω は重心の並進速度、重心周りの回転速度であり、 f_c, τ_c は制御できる力及びトルクである。

次に、Rcc 周りでの望ましい運動方程式を次の式で与える。

$$M_d\dot{v}_d = f \quad (4.39)$$

$$I_d\dot{\omega}_d + \omega_d \times I_d\omega_d = r_c \times f \quad (4.40)$$

ただし、 v_d, ω_d は Rcc の望ましい並進速度、回転速度である。この関係が成り立つように、 f_c, τ_c を決めるのが今の問題である。モデルマッチングの考え方に従って、式 (4.38) と式 (4.40) を比べると、 τ_c は

$$\tau_c = -r \times f + r_c \times f = l \times f \quad (4.41)$$

とすれば良いことが分かる。一方、並進速度に関しては、

$$v = v_d + \omega_d \times l \quad (4.42)$$

$$\dot{v} = \dot{v}_d + \dot{\omega}_d \times l + \omega_d \times \omega_d \times l \quad (4.43)$$

の関係があるので、

$$\dot{v} = \dot{v}_d = M_d^{-1} f \quad (4.44)$$

となるように、 f_c を逆算するば良い。具体的には、

$$M^{-1}(f_c + f - G) = M_d^{-1} f + \dot{\omega}_d \times l + \omega_d \times \omega_d \times l \quad (4.45)$$

であるので、

$$f_c = M(M_d^{-1} f + \dot{\omega}_d \times l + \omega_d \times \omega_d \times l) - f + G \quad (4.46)$$

となる。ただし、

$$\dot{\omega}_d = I_d^{-1}(r_c \times f - \omega \times I_d\omega) \quad (4.47)$$

である。

4.2.3 モデル追従制御に基づく制御則

上で述べた実現の方法は、パラメータが完全に分かっているとして、必要な加速度を実現するように入力を決定するものであった。このような制御の場合、制御に用いているパラメータと、実際のパラメータの間に誤差が合った場合、制御系が安定になったとしても主に重力項の補償誤差によって定常的な偏差を避けることができない。そこで、状態までも含めた完全な追従を諦めて、漸近的にシステムの挙動が目標のそれに一致する制御系を考える。これがモデル追従制御系である。特に、ここでは定値外乱が存在するばあにも有効なモデル追従制御系を考える。この制御系を特に仮想内部モデル (VIM) 追従制御と呼ぶこともある。

全節のように、望みのインピーダンス特性を実現した場合、その加速度は (4.35) を満たさなければならない。そこで、この式の右辺を改めて望ましい仮想の目標値 (Virtual Reference) と考えて \ddot{x}_r と置く。更に、分解加速度制御系を構成した場合、新しい入力 v に対して

$$\ddot{x} = v \quad (4.48)$$

とすることができる。今、実際の位置 x と仮想目標値の偏差を

$$e := x - x_r \quad (4.49)$$

とすると、

$$\ddot{e} = v - \ddot{x}_r \quad (4.50)$$

となる。ここで $u := v - \ddot{x}_r$ と置き、定値外乱に対する定常偏差を零にするように次の拡大系を考える。

$$\frac{d}{dt} \begin{bmatrix} e \\ \dot{e} \\ \ddot{e} \end{bmatrix} = \begin{bmatrix} 0 & I & 0 \\ 0 & 0 & I \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \\ \ddot{e} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix} \dot{u} \quad (4.51)$$

このシステムに対して $x^T = [e, \dot{e}^T, \ddot{e}^T]$ と \dot{u} に対して、閉ループ系が内部安定になるように、次の評価関数

$$J = \int_0^{\infty} x^T Q x + \dot{u}^T R \dot{u} dt \quad (4.52)$$

を最小にする 2 次形式最適制御系を設計すると

$$\dot{u} = -F x' \quad (4.53)$$

なる制御則が得られる。これを $x(0) = 0$ と仮定して両辺を積分すると

$$u(t) = -F \int x \tau = -F \begin{bmatrix} \int e d\tau \\ e \\ \dot{e} \end{bmatrix} \quad (4.54)$$

となる。この補償器には積分補償器を含むので、定置外乱に対しては偏差 e は零になる。(または、上の自由応答系の漸近収束性よりも明らか。)

実際の入力は

$$v = \ddot{x}_r - F \begin{bmatrix} \int e d\tau \\ e \\ \dot{e} \end{bmatrix} \quad (4.55)$$

となる。

第 5 章

遠隔操作系の設計

宇宙探査、深海探査など人間が現場へ直接行くことが危険である場合や、特殊な専門技能を持つ医師が遠隔地にいる患者の手術をするなど、遠隔にあるものを操作性よく扱うことが要求されるようになってきている。これを実現する一つの方法として、遠隔地におかれたロボットシステムと機構的等価なものを操作者がいる側にも用意し、それを操作することによって遠隔のシステムを操作するものをマスタースレーブシステムと呼ぶ。この際、操作者（オペレータ）が操作するものをマスタシステム、遠隔地に置かれたものをスレーブシステムと呼ぶ。このシステムでは、基本的にマスターシステムの動作をスレーブ側のシステムが忠実に再現することを基本にしている。このマスタースレーブシステムは、システム間の通信のやり取りの仕方によって

- ユニラテラル・マスタスレーブシステム
- バイラテラル・マスタスレーブシステム

に大別される。ユニラテラル・マスタスレーブシステムはマスタからスレーブ側への一方向の通信のみが行われ、通常マスタの姿勢/速度の情報が送られる。このシステムでは、スレーブシステムが何らかの障害物に衝突したような場合でも、その情報はマスタ側へは伝えられない。このため、操作性は一般に良くない。この欠点を克服するように、スレーブ側からも信号をマスタ側へ返すようにしたものがバイラテラル・マスタスレーブシステムである。通常のバイラテラル・マスタスレーブシステムでは、スレーブが外界から感じた力情報がマスタ側に通信される。しかし、スレーブとマスタでの双方向通信を実現するために、システム全体としてフィードバック経路が形成されてしまい、通信によるむだ時間が無視できないような場合には不安定なシステムとなってしまうことがある。本節では、この不安定性をどのように回避するかを説明する。

5.1 バイラテラル系の等価表現と伝送遅れ補償

5.1.1 ハイブリッド行列とスカタリング行列

次の 2 端子対網に関して、

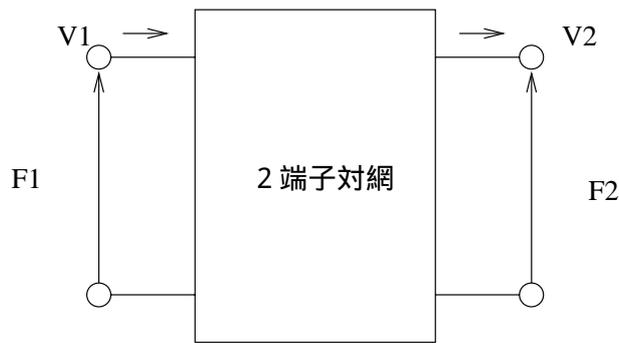
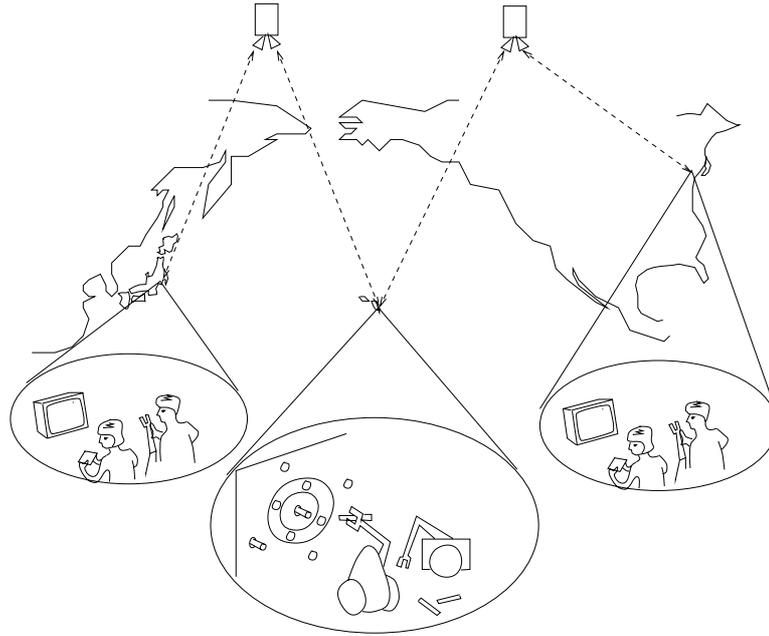
$$\begin{bmatrix} F_1(s) \\ -v_2(s) \end{bmatrix} = H(s) \begin{bmatrix} v_1(s) \\ F_2(s) \end{bmatrix} \quad (5.1)$$

を満たす行列 $H(s)$ をハイブリッド行列という。一方、

$$\begin{bmatrix} F_1(s) - v_1(s) \\ F_2(s) + v_2(s) \end{bmatrix} = S(s) \begin{bmatrix} F_1(s) + v_1(s) \\ F_2(s) - v_2(s) \end{bmatrix} \quad (5.2)$$

を満たす行列 $S(s)$ をスカタリング行列という。これら 2 つの関係より $H(s)$ と $S(s)$ は次の関係がある。

$$S(s) = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} (H(s) - I)(H(s) + I)^{-1} \quad (5.3)$$



5.1.2 受動性とスカタリング行列

n- 端子対網が受動的であるとは、

$$\int_0^T F^T(t)v(t)dt \geq -\beta, \quad \beta \geq 0 \quad \text{for } \forall T \tag{5.4}$$

を満たすときである。

この条件はスカタリング行列を用いてノルム (H_∞ ノルム) の条件に置き換えることができる。

[定理]

n- 端子対網が受動的であるための必要かつ十分条件は

$$\|S(s)\|_\infty \leq 1 \tag{5.5}$$

ただし、

$$\begin{aligned} \|S(s)\|_\infty &= \sup_\omega \sqrt{\lambda_{max}(S^*(\omega)S(j\omega))} = \sup \frac{\|Sx\|_2}{\|x\|_2} \\ S^*(j\omega) &= S^T(-j\omega), \quad \|x\|^2 = \int_0^\infty x^T(t)x(t)dt \end{aligned}$$

で、 $\|x\|_2$ は L_2 ノルム。

[証明](十分性のみ)

$$\|S(s)\| \leq 1$$

とすると、

$$\|F + v\|^2 \geq \|F - v\|^2 \quad (5.6)$$

$$4 \langle F, v \rangle \geq 0, \quad \langle F, v \rangle := \int_0^\infty F^T(t)v(t)dt \quad (5.7)$$

となり受動的であることがわかる。

5.1.3 信遅れと受動性の崩壊

今2端子対網に伝達遅れが存在して次のような関係があるとする。

$$v_2(t) = v_1(t - T) \quad (5.8)$$

$$F_1(t) = F_2(t - T) \quad (5.9)$$

ただし、 T はむだ時間。これより、

$$\begin{bmatrix} F_1(s) \\ -v_2(s) \end{bmatrix} = \begin{bmatrix} 0 & e^{-Ts} \\ -e^{-Ts} & 0 \end{bmatrix} \begin{bmatrix} v_1(s) \\ F_2(s) \end{bmatrix} \quad (5.10)$$

となるので、ハイブリッド行列は

$$H(s) = \begin{bmatrix} 0 & e^{-Ts} \\ -e^{-Ts} & 0 \end{bmatrix} \quad (5.11)$$

となる。これに対応して、スキャタリング行列は

$$S(s) = \begin{bmatrix} -\tanh(Ts) & \operatorname{sech}(Ts) \\ \operatorname{sech}(Ts) & \tanh(Ts) \end{bmatrix} \quad (5.12)$$

この S に対して H_∞ ノルムを計算してみると、

$$S^T(j\omega)S(j\omega) = \begin{bmatrix} \tan^2(\omega T) + \sec^2(\omega T) & 2j\tan(\omega T)\sec(\omega T) \\ -2j\tan(\omega T)\sec(\omega T) & \tan^2(\omega T) + \sec^2(\omega T) \end{bmatrix} \quad (5.13)$$

であるので、各 ω に対する固有値 λ は

$$\lambda = (|\tan(\omega T)| + |\sec(\omega T)|)^2, (|\tan(\omega T)| - |\sec(\omega T)|)^2 \quad (5.14)$$

となる。従って、

$$\|S(s)\|_\infty = \sup_\omega \sqrt{\lambda_{\max}(S^T(-j\omega)S(j\omega))} = \sup_\omega (|\tan(\omega T)| + |\sec(\omega T)|) = \infty \quad (5.15)$$

となり、この回路は受動的でないことがわかる。

5.1.4 重力補償を施したロボットアームの受動性

ロボットの関節に摩擦のない運動方程式は次のようになる。

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau \quad (5.16)$$

ただし、

- q : 一般化座標ベクトル (全ての関節が回転関節なら $q^T = [\theta_1, \dots, \theta_n]$)
- τ : 一般化入力ベクトル
- $D(q)$: 慣性テンソル
- $C(q, \dot{q})\dot{q}$: 中心力、コリオリ力項
- $G(q)$: 重力項

である。

ここで注意する点は、

$$D(\dot{q}) - 2C(q, \dot{q}) \quad (5.17)$$

が歪み対称になるような行列 $C(q, \dot{q})$ (一意ではない) が存在することである。ただし、行列 P が歪み対称行列である条件は、

$$P^T = -P \quad (5.18)$$

が成り立つことである。このような行列では、2次形式を作るとどんな x に対しても

$$x^T P x = 0 \quad (5.19)$$

となる。

今制御入力として、重力項を完全に補償するように

$$\tau = G(q) + F \quad (5.20)$$

とする。ただし、 F は新しい入力である。こうすると、 F から見た運動方程式は

$$D(q)\dot{v} + C(q, v)v = F \quad (5.21)$$

となる。ただし、 $\dot{q} := v$ とした。このように表現すると、 F から v は受動的となる。実際、今関数 $V(q, v)$ (運動エネルギー) を

$$V(q, v) = \frac{1}{2} v^T D(q) v \quad (5.22)$$

で定義すると、慣性テンソルは正定対称行列であるので

$$V(q, v) \geq 0 \quad (5.23)$$

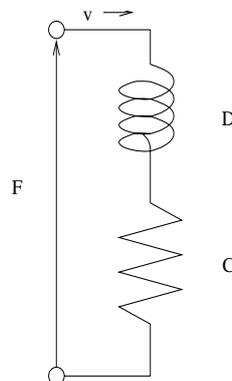
である。今 V を時間で微分すると、

$$\begin{aligned} \dot{V} &= v^T (D\dot{v} + \frac{1}{2} D(\dot{q})v) \\ &= v^T (F - C(q, v)v + \frac{1}{2} D(\dot{q})v) \\ &= v^T F + \frac{1}{2} v^T (D(\dot{q}) - 2C(q, v))v \\ &= v^T F \end{aligned}$$

従って、上式の両辺を 0 から T まで時間で積分すると

$$\int_0^T v^T F dt = V(T) - V(0) \geq -V(0) \quad (5.24)$$

となり、受動的であることがわかる。式 (5.21) のシステムを電気回路とのアナロジで次のように表現する。



5.1.5 基本的なバイラテラルマスタ・スレーブ系の構成

今、式(5.21で表されるようなロボットを2台用意して、力帰還型のマスタ・スレーブシステムを構成することを考える。2台のロボットの重力を補償した後の運動方程式を次のように表す。

$$D_m(q_m)\dot{v}_m + C_m(q_m, v_m)v_m = -F_{cm} + F_o \quad (5.25)$$

$$D_s(q_s)\dot{v}_s + C_s(q_s, v_s)v_s = F_{cs} - F_e \quad (5.26)$$

ただし、 F_o : 操作者がマスタアームに加える操作力
 F_e : 外界がスレーブアームに加える外力

今考えているバイラテラルマスタ・スレーブ系の目的は

1. v_s を v_m に一致させる。(マスタの動きにスレーブを一致させる。)
2. F_o を F_e に一致させる。(操作者がスレーブの外力を正確に感じる。)

である。このために次の最も簡単な制御系を考える。

スレーブへの制御入力 F_{cs} : $F_{cs} = K_v e + K_p \int e dt$, $e := v_m - v_s$

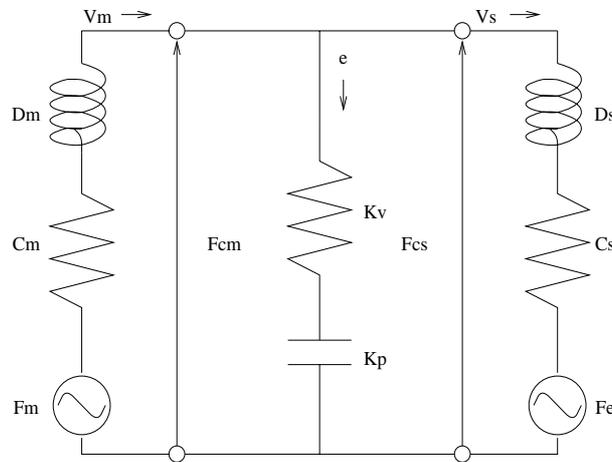
マスタへの制御入力 F_{cm} : $F_{cm} = F_{cs}$

勿論、この制御則では操作者がマスタアームの慣性を直接感じてしまうためあまり良い制御ではないが、ここでは簡単のためにその点は無視する。スレーブの制御は位置に関してはPD制御系になっている。また、 $v_s = v_m = 0$ の場合、運動方程式より

$$F_{cs} = F_e$$

$$F_m = -F_{cm}$$

となるので、 $F_{cm} = F_{cs}$ とすると平衡状態では $F_m = F_e$ となる。力の方も、アームが運動している場合には、慣性、制御則の影響を受けて正確な情報は伝わらないが、その点もここでは無視する。この制御系を適用した場合のシステム全体の等価回路を次に示す。この回路より、 F_m, F_e が零(短絡した場合)には回路は受動素子のみからなり、



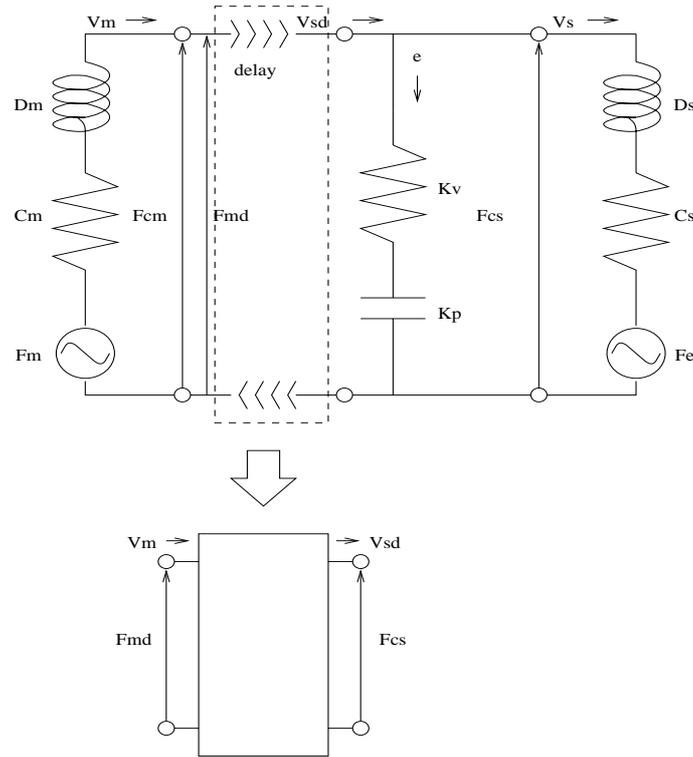
回路全体としても受動的であることがわかり不安定になることはない。

次にマスタシステムとスレーブシステムが遠隔に離れて置かれ、通信に時間がかかる場合を考える。つまり、 v_m と F_c が時間遅れてを伴ってそれぞれ相手のシステムに送られる場合を考える。今、

$$v_{sd}(t) = v_m(t - T)$$

$$F_{md}(t) = F_{cs}(t - T)$$

とすると、伝送路は次の図のように2端子対網となるが、これは上の解析より受動的ではない。したがって、この伝送路を組み込まれた回路は受動的でなくなる。(受動的ではないというだけで、必ずしも不安定になるわけではない。)しかし、実際にこのような構成のシステムを作成すると、小さな伝送遅れで簡単に系全体が不安定になってしまう。



5.1.6 伝送路の受動化

上で見たように伝送路は受動的でないので、これを受動化するために

$$\begin{bmatrix} F_{md}(s) - v_m(s) \\ F_{cs}(s) + v_{sd}(s) \end{bmatrix} = S(s) \begin{bmatrix} F_{md}(s) + v_m(s) \\ F_{cs}(s) - v_{sd}(s) \end{bmatrix} \quad (5.27)$$

ただし、

$$S(s) = \begin{bmatrix} 0 & e^{-sT} \\ e^{-sT} & 0 \end{bmatrix}$$

を考える。この時 $S^*(j\omega)S(j\omega) = I$ となるので、 $\|S(s)\|_\infty = 1$ となりこの 2 端子対網は受動的になることがわかる。信号の時間領域での表現は

$$\begin{aligned} F_{md}(t) &= v_m(t) + (F_{cs} - v_{sd})(t - T) \\ v_{sd}(t) &= -F_{cs}(t) + (F_{md} + v_m)(t - T) \end{aligned}$$

となり入手可能な信号から生成できることがわかる。また、伝達遅れ T が零の時には $F_{md} = F_{cs}, v_{sd} = v_m$ となり、通信回路がない場合に一致することに注意する。この変換をブロック線図で表すと次のようになる。

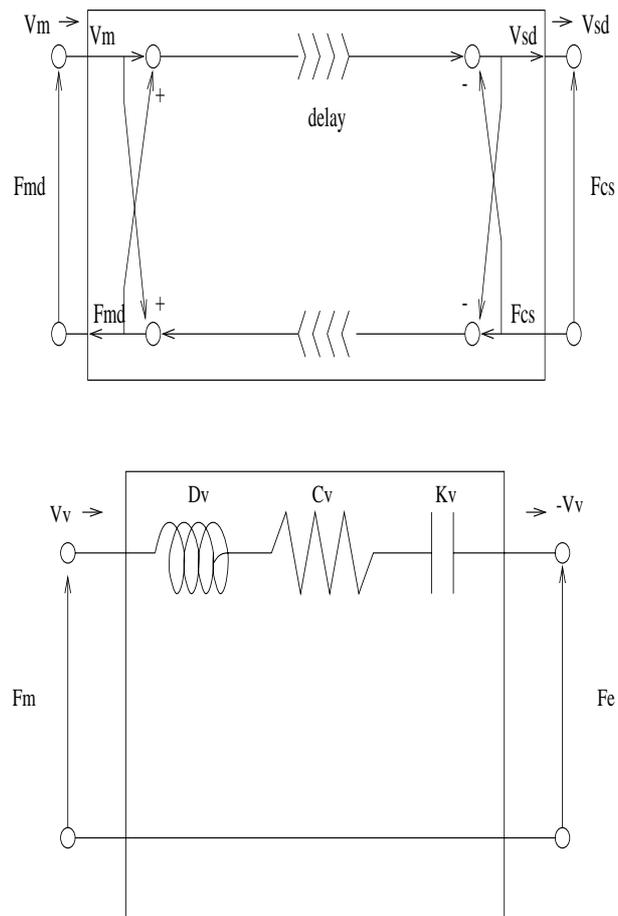
この手法の特徴は、マスタシステム、スレーブシステムがそれぞれ受動性の概念に基づいて設計されていれば、通信遅れが存在してもそのパラメータを変更しなくとも、 T の大きさに関係なく安定性が保存される点である。(勿論、性能は一般的に T が大きくなると悪くなる。)

5.1.7 動的システムの仮想実現

今、離れた地点に置かれた人が同じ動的システムを共有しているようなシステムを作ること考える。それぞれの人は、 $F_m, -F_e$ を加えることができるとする。この時、共有する仮想のシステムの運動方程式を

$$D_v(q_v)\dot{v}_v + C_v(q_v, v_v)v_v + K_v q_v = F_m - F_e \quad (5.28)$$

で与えられると考える。ここで、 K_v はバネによる復元力の項である。この方程式で広いクラスの機械システムは表現することができる。この運動方程式と等価な回路表現は次のようになる。



この回路も明らかに受動的である。この動きを2つの地点に置かれた機械的なシステムでその動作を模倣するための最も簡単な制御系はマスタ・スレーブシステムと同様に次のようになる。

