＜連載＞

第 4 回　PID 制御

PID Control

須田　信英 *

大阪大学　名誉教授

Nobuhide Suda*
Professor Emeritas, Osaka University

5. ワインドアップ現象とその対策

5.1 操作部の飽和特性

これまで自動制御システムを構成する制御対象，検出部，操作部，調節部の特性は，線形であるか少なくとも線形特性で近似できるものとしてきた．しかし現実にはさまざまな非線形特性が存在し，その影響を考慮した設計を行わなければならないことが多い．そのなかでも，比較的しばしば遭遇し，しかもその影響が大きいのが操作部の飽和特性である．操作部には，弁の全開と全閉，モータの回転速度の限界など，機器自体が飽和特性を持つことがある．また，操作部の機器を保護するため，人為的にリミッタを設けて過大な電流や電圧が加わらないようにすることもあり得る．どちらの場合も，操作部は飽和特性を持つことになる．調節部が制御則に基づいて計算した操作量の指令信号がこの飽和値を超えると，実際に制御対象に加わる操作量は飽和値にとどまるから，調節部が加えているつもりの操作信号 $u(t)$ と実際に加わる操作量 $v(t)$ とに乖離を生じる．それによって制御性能が劣化するので，このような飽和特性を含む場合には，制御性能の劣化のしかたを調べ，その対策を考えておく必要がある．非線形特性を含む制御システムの理論的に厳密な取り扱いは非常に難解である．ここではごく直感的な説明だけにとどめる．

Fig. 1 に示した自動制御システムの構成のうち，調節部と操作部の部分を **Fig. 19**（a）に再掲する．なお，**Fig. 1** では操作部の入力も，出力もどちらも「操作信号」と記してあるが，これは誤植であり，正しくは，**Fig. 19**（a）の通り，操作部の入力が「操作信号」，その出力が「操作量」である．お詫びして訂正する．

以下，4.3 節で述べた微分先行型 PID 制御（PI-D 制御）を用いる．4.2 節で示したとおり，2 自由度 PID 制御の制御則は

$$u(t)=K_I\int_0^t\{r(\tau)-y(\tau)\}d\tau+c_PK_Pr(t)+c_DK_D\frac{dr(t)}{dt}-K_Py(t)-K_D\frac{dy(t)}{dt}$$

であり，微分先行型では $c_D=0, c_P=1$ であるから，操作信号 $u(t)$ は **Fig. 19**（a）に示す通り

$$u(t)=K_I\int_0^t\{r(\tau)-y(\tau)\}d\tau+K_P\{r(t)-y(t)\}-K_D\frac{dy(t)}{dt}$$

$$=K_I\int_0^te(\tau)d\tau+K_Pe(t)-K_D\frac{dy(t)}{dt}$$

となる．

ここで操作部の特性は単純な飽和特性

$$v(t)=\begin{cases}M & u(t)\geq M\\ u(t) & -M\leq u(t)\leq M\\ -M & u(t)\leq -M\end{cases}$$

であるとする．M が飽和限界である．したがって飽和特性が存在しない，あるいは飽和限界 M が十分大きい場合には，操作量 $v(t)$ は操作信号 $u(t)$ に一致する．実際には，飽和がなくても，操作部の動特性による遅れのために，操作量が操作信号と一致しないことが多いが，ここでは説明を簡潔にするため，上記の仮定をおく．

Fig. 19（a）では，操作部の出力である操作信号 $u(t)$ 以

* 〒 180–0013　武蔵野市西久保 1–44–15
TEL & FAX: 0422–54–6624
E-mail: nsuda@gakushikai.jp

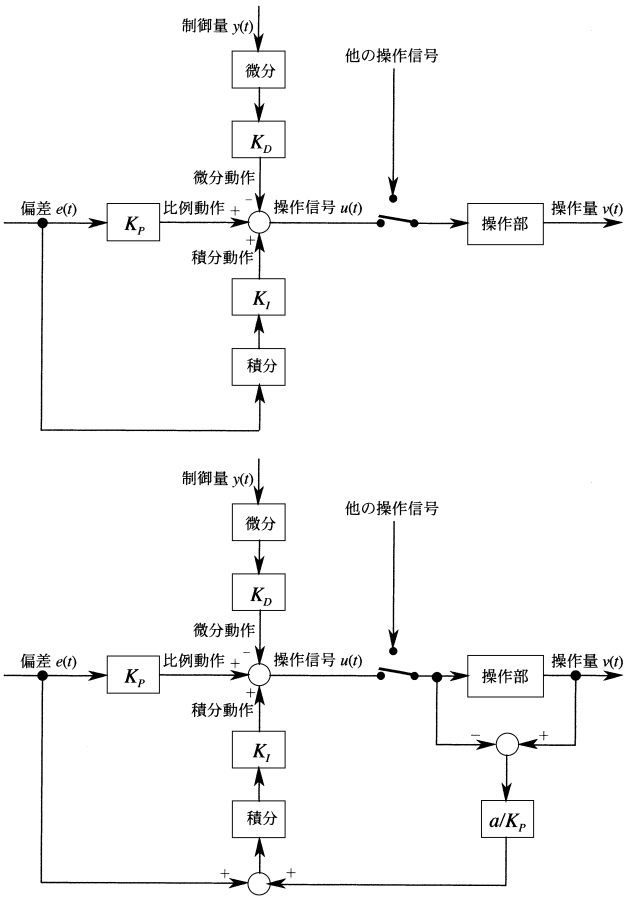


Fig. 19 Structure of Controller.
(a) Ordinary Controller.
(b) Controller with Anti-windup Scheme.

外に，他の操作信号が存在し，スイッチで切り替えられるように描いてある．これについてはのちの 5.5 節で説明する．

5.2 ワインドアップ現象

先ず例題についてシミュレーションを行ってみる．いつもの例題に対して，Chien, Hrones and Reswick の調整則によって，目標値追従を目的に調整したときおよび外乱抑制を目的に調整したときの，ステップ状目標値変化に対する応答を取り上げる．すなわち，前回の **Fig. 14** の実線および破線である．ただし，前回は PID 制御の原型を用いていたが，ここでは PI-D 制御を用いる点が異なっている．

以下，外乱抑制を目的に調整した場合，すなわち前回の **Fig. 14** の破線のケースを Case A と呼び，目標値追従を目的に調整した場合，すなわち前回の **Fig. 14** の実線のケースを Case B と呼ぶことにする．

はじめに Case A を取り上げ，ステップ状目標値変化の大きさ，つまりステップ（階段）の高さを変化させた場合の応答を **Fig. 20** に示す．**Fig. 20**（a）が制御量の応答で，破線が飽和特性のない場合の応答，実線が $M=2$ という飽和特性を想定した場合の応答である．

まず破線に注目しよう．制御量の応答は，ステップ高さに比例して拡大，縮小したものとなっている．すなわち，ステップ高さが r のときの制御量の応答を $y_r(t)$ とすれば

$$y_r(t)=ry_1(t)$$

という関係が成り立つ．これは全ての特性を線形と仮定しているから，入力である目標値変化が r 倍になれば，出力である制御量の変化も r 倍になるという，当然の結果である．そこで，制御量が目標値へ達するに要する時間，たとえば最初に目標値と等しくなる時刻は，ステップ高さ r に関係なく一定である．

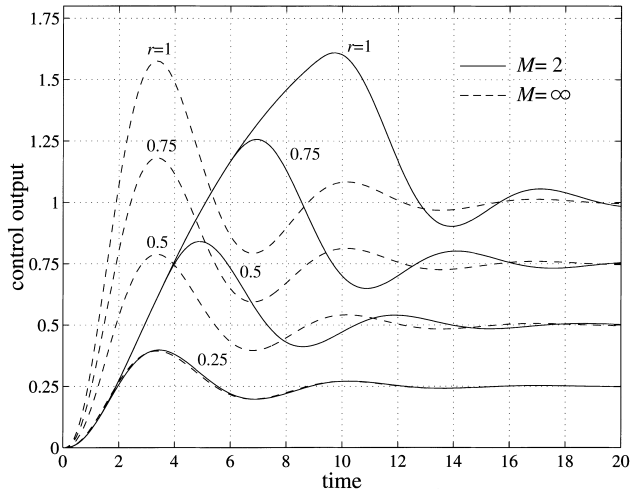
Fig. 20（b）が操作量の応答で，実線と破線の区別は上と同様である．やはり破線に注目すると，操作量の応答も，ステップ高さに比例して拡大，縮小したものとなっている．ステップの高さが大きければ，大きい操作量を加えることにより，制御量を急速に目標値へ近づけているわけで，そのおかげで制御量が目標値へ達するに要する時間がステップ高さ r に関係なく一定なのである．

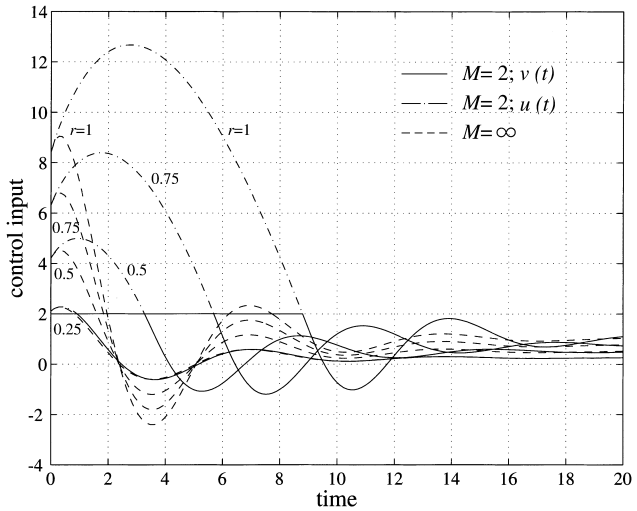
Fig. 20（b）の実線が上記の通り $M=2$ という飽和特性を想定した場合の操作量であり，鎖線がその場合の操作信号を示している．明らかに飽和特性にかかっており，したがって，ステップ高さが大きいから大きい操作量を加えて急速に目標値へ近づけようとしても，それはできない．飽和限界に等しい操作量を加えて，じわじわと目標値へ近づけるしかないわけである．

そこで **Fig. 20**（a）の実線を見ると，ステップ高さに関係なく同じ速さで立ち上がっており，高さが低いほど早く飽和からぬけて，目標値へ収束していることが読みとれる．つまり，目標値へ達するに要する時間は，ステップ高さが大きいほど長いのである．

Fig. 21 は，Case B について，同様のシミュレーションを行った結果である．

次に，Case A において，目標値のステップ高さは 1 と固定し，飽和のない場合（つまり $M=\infty$）を含む，5 種類の飽和値 M について，ステップ状目標値変化に対する応答を **Fig. 22** に示す．**Fig. 22**（a）が制御量の応答である．**Fig. 22**（b）の実線が実際に制御対象に加わる操作量 $v(t)$





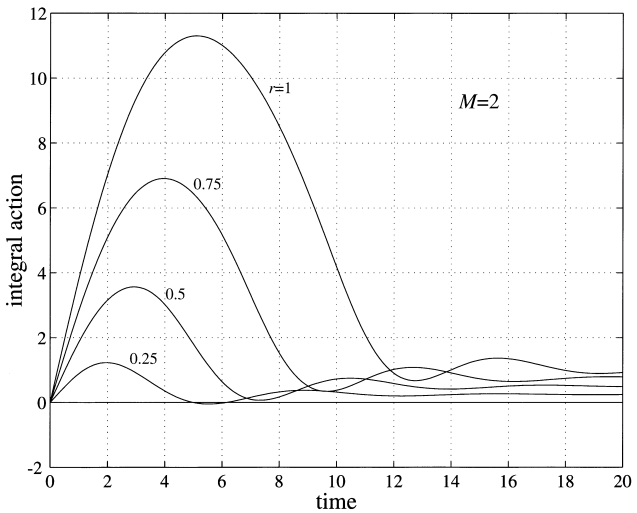
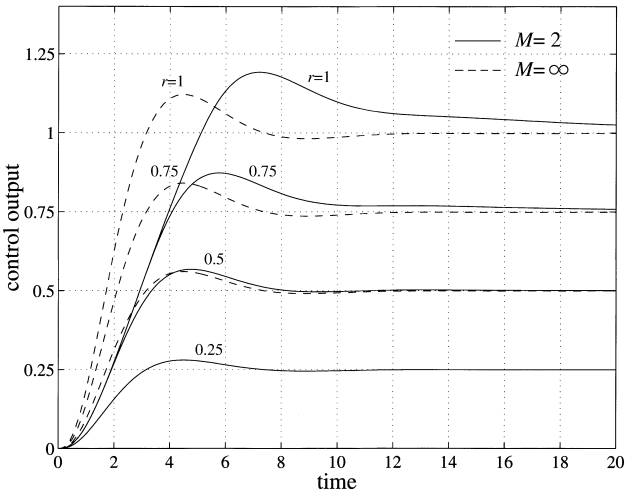
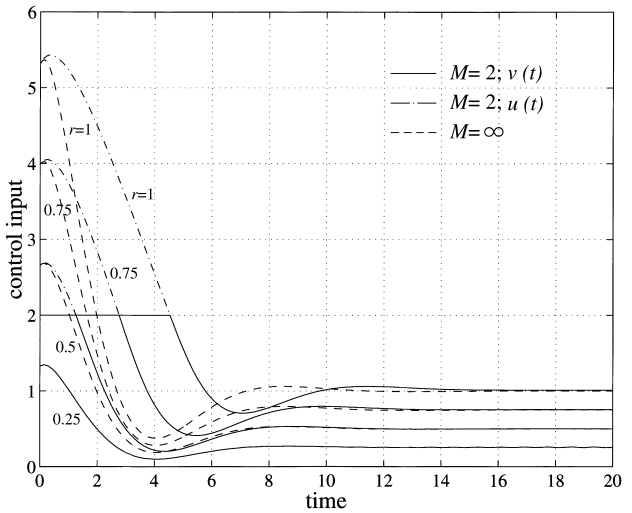


Fig. 20 Effect of Setpoint Change Magnitude: Case A.
(a) Controlled Variable. (b) Manipulated Variable.
(c) Integral Action.





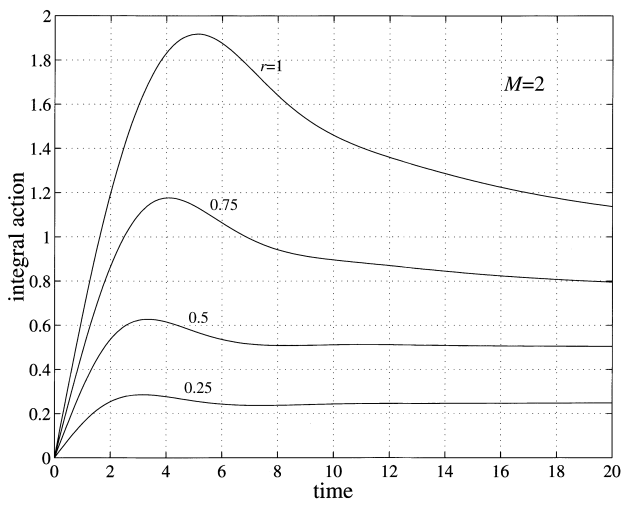


Fig. 21 Effect of Setpoint Change Magnitude: Case B.
(a) Controlled Variable. (b) Manipulated Variable.
(c) Integral Action.

で，鎖線は調節部の出力である操作信号 $u(t)$ である．はじめの方で飽和特性にかかって，$u(t)$ と $v(t)$ とが乖離しているのがわかる．**Fig. 22**（a）を見ると，M が小さい，つまり飽和特性が厳しいほど，制御量の立ち上がりが遅くなっている．これは，前述の通り，もっと大きい操作量を加えて速く制御量を目標値に近づけたいところなのに，飽和のために小さい操作量しか加わらないから，制御量の変化が遅いわけで，当然の結果である．

Fig. 22（a）でもうひとつ気がつくのは，飽和特性が厳しいほどオーバーシュートが大きいことである．そこで **Fig. 22**（b）を見ると，操作信号 $u(t)$ が大きくなっており，それに伴って操作量 $v(t)$ が飽和値未満に戻るまでに長時間を要していることがわかる．その原因は積分動作にあり，飽和のために偏差が大きくなり，それを貯め込むことによって積分動作が，**Fig. 22**（c）に示すとおり，大きくなっているのである．つまり，制御量を増加させる方向の偏差が積分によって蓄積されているため，制御量が目標値に達してそれ以上の増加は不必要になったり，オーバーシュートを生じてむしろ減少させる必要が生じても，一旦貯め込んだ偏差をある程度放出するまでは，制御量を増す方向の操作量が加わってオーバーシュートが大きくなるのだと考えられる．あえて例えれば，思い込みが強くて，状況が変わっても急には気持ちの切り替えができない人に似ているかもしれない．

これを，偏差の積分値が巻き上がり（wind up）それを巻き戻すのに時間がかかるようなものだというわけで，**リセット・ワインドアップ**とか，**積分ワインドアップ**とか呼んでいる．積分動作のことをリセット動作ともいうのである．

余談であるが，飽和値を α 倍し同時に目標値のステップ高さも α 倍すると，制御量，操作量の応答もやはり α 倍になることは容易にわかる．たとえば，**Fig. 22** の M=4，ステップ高さ 1 の場合と，**Fig. 20** の M=2，ステップ高さ 0.5 の場合とを比べると，後者は前者の 0.5 倍となっている．

次に Case B を取り上げ，飽和のない場合（つまり M=∞）を含め，やはり 5 種類の飽和値について，ステップ状目標値変化に対する応答を Fig. 23 に示す．**Fig. 23**（a）が制御量の応答である．前と同様に，**Fig. 23**（b）の実線が実際に制御対象に加わる操作量 $v(t)$ で，鎖線は調節部の出力である操作指令 $u(t)$ である．

Fig. 23（a）においても飽和特性が厳しいほど，制御量の立ち上がりが遅くなっている．しかしオーバーシュートの増大は前のケースほどには顕著でない．**Fig. 23**（c）に示す積分動作は，M=1.5 の場合を除き，飽和値未満に

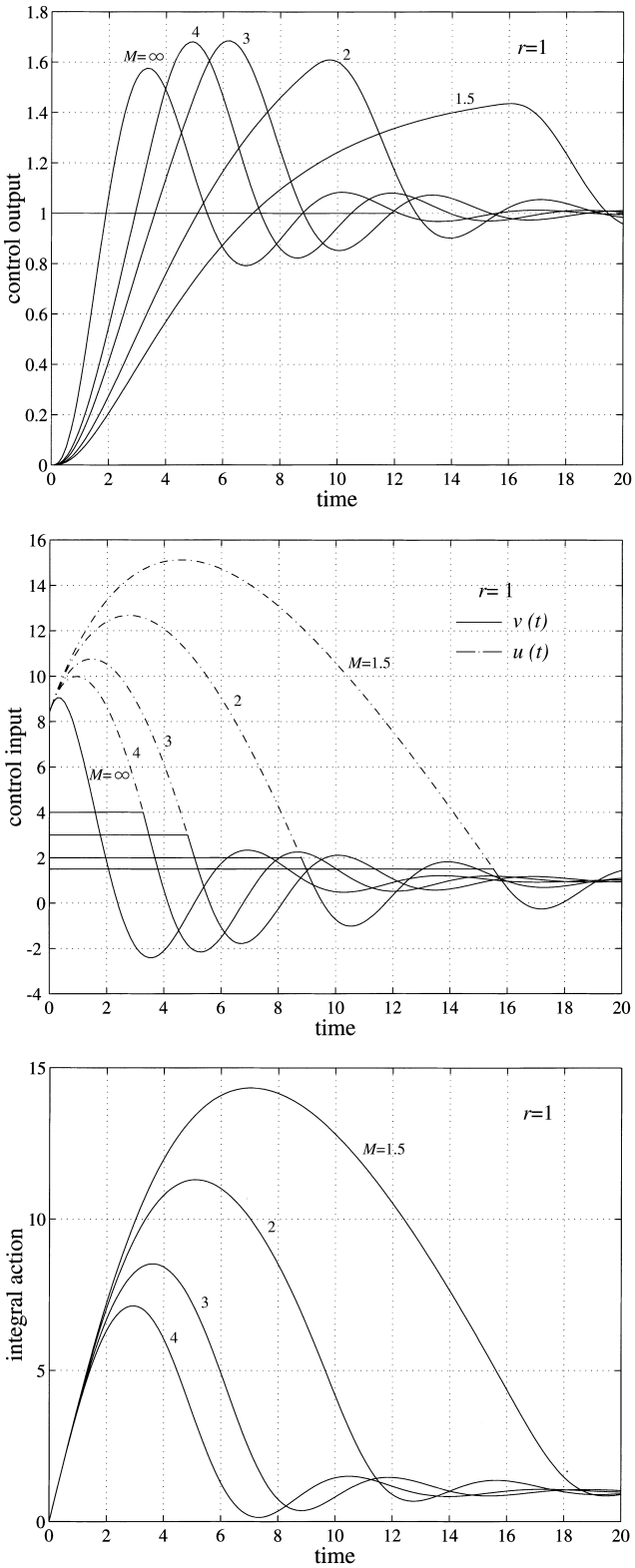


Fig. 22 Effect of Saturation Level: Case A.
(a) Controlled Variable. (b) Manipulated Variable.
(c) Integral Action.

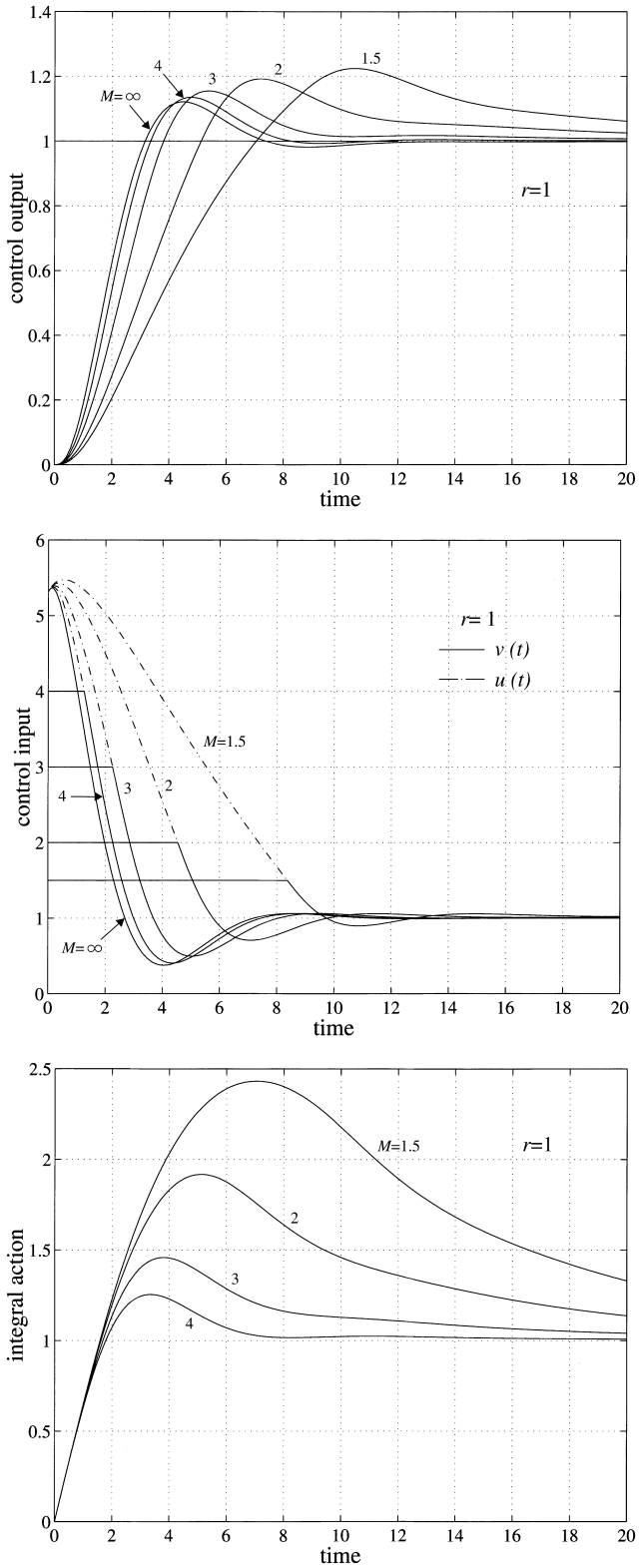


Fig. 23　Effect of Saturation Level: Case B.
(a) Controlled Variable. (b) Manipulated Variable.
(c) Integral Action.

とどまっている. つまり積分動作単独では M=1.5 の場合以外は飽和特性にかからない．飽和特性にかかるのは主として大きい偏差に比例する比例動作のためである.

そういう違いを生じるのは，積分時間 T_I が Case A に比べて大きい（Case A では 2.16，Case B では 8.05）からであると思われる．積分時間は，比例動作の係数 K_P と積分動作の係数 K_I との比，つまり

$$T_I = K_P/K_I$$

であるから，積分時間 T_I が大きいということは，比例動作に比べて積分動作のウェイトが小さいことを意味している. そこで積分時間の小さい Case A では積分動作の影響が強く，Case B ではあまり影響がないのだと考えられる．そうなると，この Case B のような場合もリセット・ワインドアップと呼ぶのは相応しくないかもしれないが，ここでは慣習に従っておく.

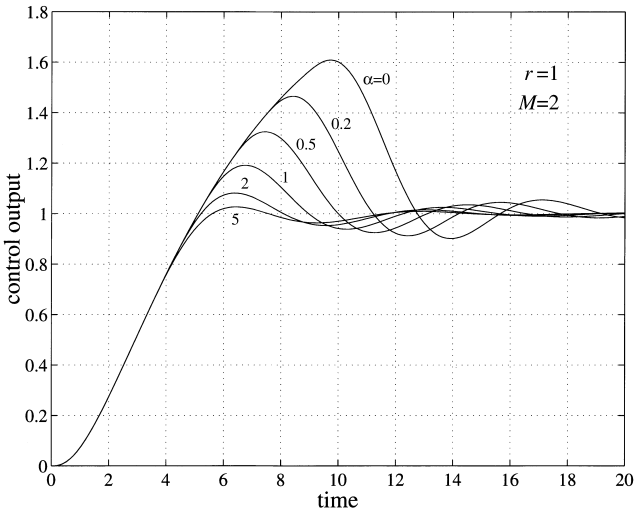
5.3　ワインドアップ対策

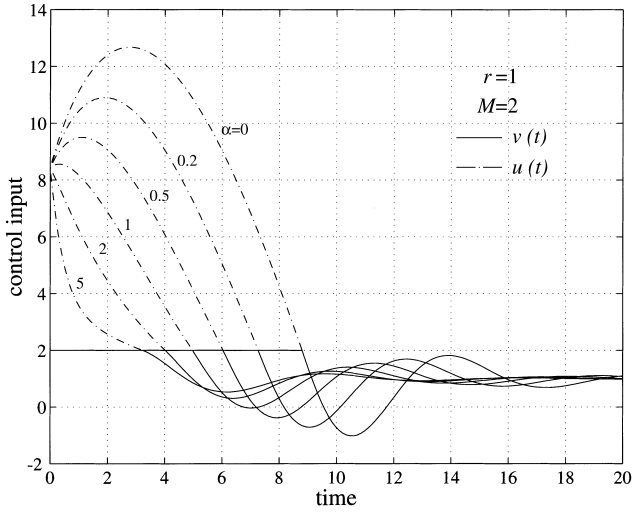
ワインドアップの影響を軽減するための対策の一つは，実際の操作量 $v(t)$ と操作指令 $u(t)$ との差 $v(t)-u(t)$ にある重み係数をかけて積分器の入力にフィードバックし，**Fig. 19**（b）の構成とすることである．重み係数を α/K_P と，比例ゲイン K_P で規格化しているのは説明の便宜のためである.

操作指令 $u(t)$ が飽和値未満であれば $v(t)$ と $u(t)$ とは一致するから，このフィードバックは作動せず，**Fig. 19**（a）の通常の PI-D 制御が行われる．**Figs. 20 〜 23** のように正の目標値変化に追従する場合，初期の偏差は正であり，したがって操作信号 $u(t)$ も正である．この $u(t)$ が飽和値を超えると $v(t)-u(t)$ は負となるから，それが積分器の入力に加算されることによって，正の偏差を一部相殺する効果をもち，ワインドアップを少なくすることが期待されるのである.

この対策の効果を見るために，飽和値は M=2 と固定し，a=0 つまりワインドアップ対策を講じない場合を含め，6 通りの係数 a についてシミュレーションを行った. **Figs. 24** が Case A，つまり **Figs. 22** に対応し，**Figs. 25** が Case B，つまり **Figs. 23** に対応している.

Fig. 24（a）と **Fig. 25**（a）が制御量の応答であるが，オーバーシュートが小さくなり，期待した効果が現れている．**Fig. 24**（b）と **Fig. 25**（b）が操作量の応答であるが，飽和値未満に戻る時期が早まる効果が確認される．そして **Fig. 24**（c）と **Fig. 25**（c）が積分動作であり，偏





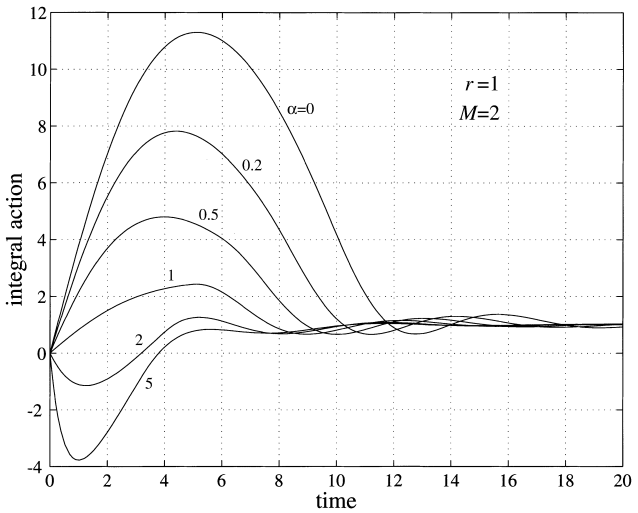
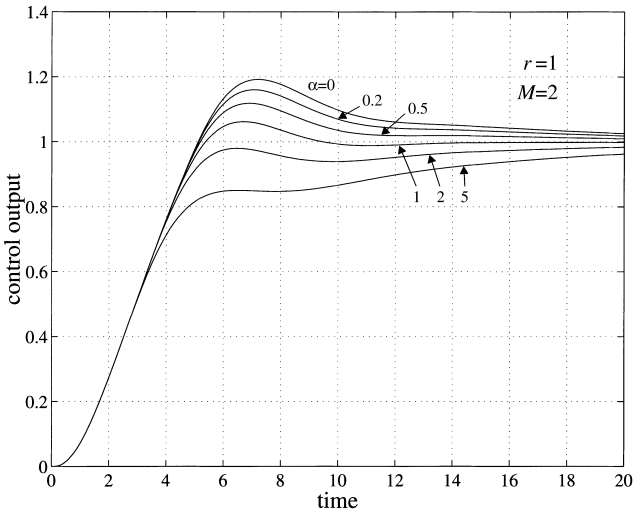
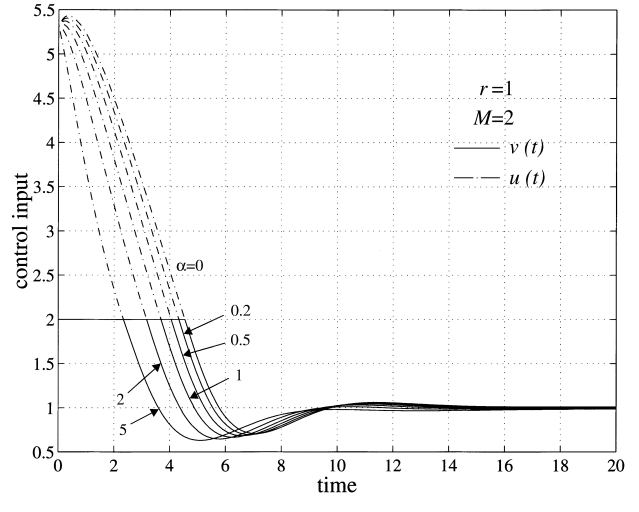


Fig. 24 Effect of Anti-windup parameter, α: Case A.
(a) Controlled Variable. (b) Manipulated Variable.
(c) Integral Action.





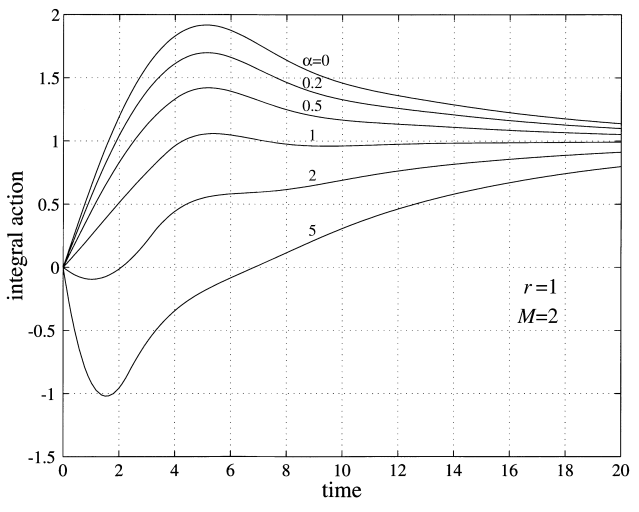


Fig. 25 Effect of Anti-windup parameter, α: Case B.
(a) Controlled Variable. (b) Manipulated Variable.
(c) Integral Action.

差の積分で巻き上がるのが抑えられている．特に係数 a を大きくすると，$v(t)-u(t)$ の負の効果が著しく，偏差は正であるにもかかわらず積分動作は負の方向に振れることさえある．

5.4 実現可能な目標値

ステップ状の急激な目標値変化に追従させようとすると，大きい修正動作が必要となり，そのために操作量が飽和限界を超えてしまう．これを回避し，操作量を飽和限界内に維持するために，最終値‘1’へ向けてゆるやかに目標値を変化させることが考えられる．4.3 節で述べたのとは目的が異なるが，やはりある意味での目標値の「緩和」である．前節のワインドアップ対策とこの緩和との関係を述べよう．

同じ制御対象，同じ制御則について，飽和特性を含まない制御系を A と呼び，飽和特性を含み，上記のワインドアップ対策を講じた制御系を B(a) と呼ぶことにする．

目標値変化 $w(t)$ を制御系 A に加えたときの制御量と操作量の応答が，目標値変化 $r(t)$ を制御系 B(a) に加えたときのそれぞれの応答に等しいとする．両者の操作量の応答が等しいということは，制御系 A においても操作量が飽和値を超えていないことを意味する．そこで $w(t)$ を操作量が飽和値を超えないで実現できる目標値，略して**実現可能な目標値**（realizable reference）と呼んでいる[10]．

若干の理論的考察から，実現可能な目標値 $w(t)$ は

$w(t)=r(t)+\{v(t)-u(t)\}$ によって決まる項

という形に表され，特に $a=1$ であれば

$$w(t)=r(t)+\{v(t)-u(t)\}/K_P$$

となることが導かれる[10]．

この関係に基づいて，ステップ状の目標値変化 $r(t)=1$ と $\{v(t)-u(t)\}$ とから計算した実現可能な目標値を**Figs. 26, 27** に示す．例によって前者が Case A，後者が Case B である．

これらの図を見ると，実現可能な目標値はどの場合でも初めは最終値‘1’へ向けてゆるやかに上昇する．そして $a<1$ の場合は一旦最終値‘1’を突き抜けて上昇し，それから戻ってくる．これは，制御量を‘1’だけ上げたいという，制御目的からすればむだな動きであり，制御量にオーバーシュートを生じる原因となっている．一方，$a>1$ の場合は最終値‘1’に達する以前に減速を始め，ゆっくりと‘1’に向かっている．これは控え目すぎるといえる．そのために **Fig. 24**（a）では制御量の目標値への接近が遅くなっている．$a=1$ の場合には，上記の式からも明らかなとおり，操作量が飽和限界内に戻った瞬間に，実現可能な目標値が本来のステップ状目標値変化と一致し，以後そのまま‘1’を保持している．これが制御目的に最も適っていると思われる．

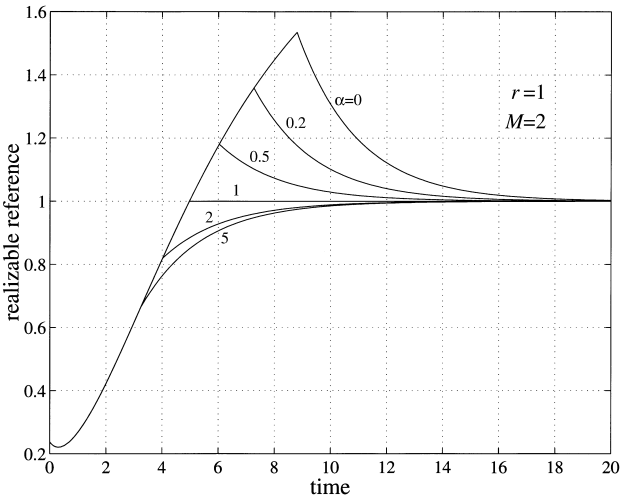


Fig. 26 Realizable Reference: Case A.

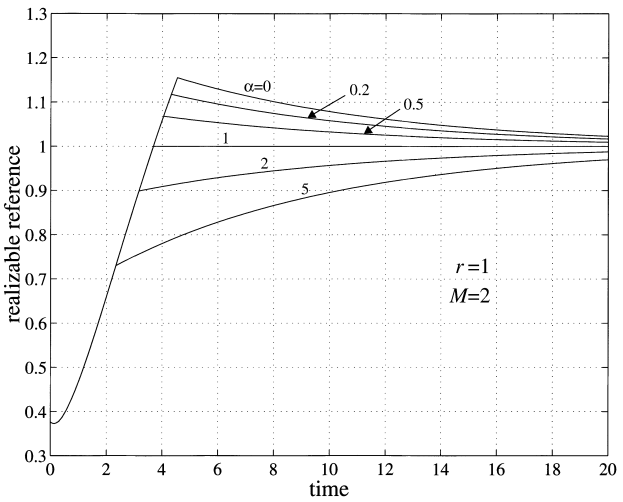


Fig. 27 Realizable Reference: Case B.

文献 11 では，**Fig. 19**（b）の構成で，$a=1$ の場合を，**自動整合制御**と呼んでいる．特に $a=1$ の場合に注目したのは上のような背景があると推察される．

なお，**Fig. 25**（a）では，**Fig. 24**（a）と異なり，$a>1$ の場合に制御量の目標値への接近が目立って遅くはなっていない．この違いも積分時間 T_I の大小と関係があるが，それについての考察は文献 12 に譲る．

5.5 ワインドアップ対策の応用

ワインドアップ対策は，これまで述べたような，操作部が飽和特性を持つ場合に，目標値の大きい変動に追従させること以外にも応用例がある．その二三を紹介しよう．

［バンプレス切り替え］

手動制御でしばらく運転し，ある時点で自動制御に切り替えるという場合がある．この場合，**Fig. 19** の他の操作信号というのが，手動制御による操作信号であり，スイッチは手動と自動の切り替えを表す．

切り替え時点における手動制御の操作信号と調節器の出力とは一般には一致しないから，切り替えに際して操作量が不連続的に変化して制御対象に大きい外乱を与えるおそれがある．そういう不連続な変化を与えないようにすることを**バンプレス切り替え**という．

ワインドアップ対策を含まない **Fig. 19**（a）の場合には，実際の操作量，すなわち手動制御による操作量と調節部の出力である操作信号との差があっても，その情報は調節部に伝わらないから，バンプレス切り替えはできない．

Fig. 19（b）に注目しよう．目標値は一定とし，手動制御によって偏差を十分長い時間にわたって零に保持したとする．比例動作，微分動作はいずれも零である．このとき，手動制御による操作量と調節部の出力とに差が生じると，その差をなくすように積分動作が働くことが示される．こうしてバンプレス切り替えができる．

この例では，操作部の飽和特性はあってもなくても考え方は同じである．

［オーバーライド制御］

たとえば，鋼板の焼鈍炉で通常は鋼板温度を制御量とするが，炉の温度が一定限度以上に上昇すると，炉の損傷を防ぐために，炉温の制御に切り替えるとか，バッファタンクで通常は流出流量を需要に合わせて制御するが，タンクの水位が上がりすぎたり，下がりすぎたときには，水位制御に切り替えるなど，平常時には本来の制御目的を果たし，異常時には本来の制御目的を一時中断して不都合な状況を回避するという制御方式がある．これは**オーバーライド制御**と呼ばれる．

水位の異常な低下を防ぐオーバーライド制御を例にとろう．水位制御系は水位をその下限に保つように，流量制御系は流出流量を需要に追従させるように，それぞれ設計される．そうして水位が下限より上にあれば流量制御系が，水位が下限に達すれば水位制御系が，その状態で流出流量が需要以上となれば再び流量制御系が，それぞれ選択される論理構成とする．

平常時で，流量制御系が選択されているとき，水位は下限とは一致しないから，水位制御系にとってみれば，水位が目標値よりも高いという偏差が継続的に加わっていることになる．これは，水位を下げる方向の偏差であり，それが積分され続けて蓄積されると，水位が下限に達して水位制御系に切り替えられても，水位がそれ以上下がらないようにするという機能がすぐには発揮されない可能性がある．

そこで水位制御系を **Fig. 19**（b）の構造とし，他の操作信号としては流量制御系の操作信号を用いる．平常時にはもちろんスイッチはこちらに閉じておく．そうすると，流量制御系と水位制御系の操作信号の差が，偏差をうち消す方向で積分器の入力に加算され，ワインドアップの軽減が期待できる．異常時おける流量制御系のワインドアップを防ぐのも同様で，流量制御系を **Fig. 19**（b）の構造とし，他の操作信号としては水位制御系の操作信号を用いればよい．

この例でも，操作部の飽和特性はあってもなくても考え方は同じである．

［バイパス弁などの制御］

たとえば，負荷の大きい変動のために圧力が高くなりすぎるのを防止するバイパス弁のように，常時偏差が持続され，特定の状況でのみ制御動作が期待される機器の制御にも応用される．

バイパス弁を例にとろう．平常運転時の圧力を P_0，バイパス弁を作動させる設定圧力を $P_1(P_1>P_0)$ とし，負荷の急変によって圧力が P_0 から $P_2(P_2>P_1)$ まで変わる場合を想定する．

圧力は負荷流量とバイパス流量によって決まる．正常の負荷流量に対して，バイパス流量が零，つまりバイパス弁が全閉のとき，圧力が P_0 となるように設計される．圧力制御系はP_1を目標値としてバイパス弁を操作するように設計される．

負荷が急減して圧力が P_1 を超えると，圧力制御系はバイパス弁を開いて圧力を P_1 に戻すように動作する．負荷が回復するにつれて，そのままでは圧力が低下するから，圧力制御系はバイパス弁を閉めて圧力を P_1 に維持する．バイパス弁が全閉になっても負荷が増加を続ければ圧力は次第に低下し，正常の負荷に対しては P_0 に戻る．

負荷流量が正常のときにも，圧力制御系はバイパス弁を閉めて圧力を P_1 に戻そうとするが，バイパス弁を全閉に

しても圧力はP_0にしかならない. つまり圧力制御系にとってみれば, 圧力が目標値よりも低いという偏差 (P_1-P_0) が継続的に加わっていることになる. これは圧力を上げる方向の偏差であり, それが積分され続けて蓄積されると, 圧力が P_1 に達しても, 圧力がそれ以上上がらないようにするという機能がすぐには発揮されない可能性がある.

圧力制御系を **Fig. 19** (b) の構造とすれば, 実際の操作量つまり全閉と操作信号との差が, 偏差をうち消す方向で積分器の入力に加算され, ワインドアップの軽減が期待できる.

［参考文献］

10) Y. Peng *et al.*: Anti-windup, bumpless, and conditioned transfer techniques for PID controllers; *Control Systems*, Vol. 16, No. 4, pp. 48–57 (1996)
11) R. Hanus *et al.*: Conditioning technique, a general anti-windup and bumpless transfer method, *Automatica*, Vol. 23, No. 6, pp. 729–739 (1987)
12) 須田信英：PID制御則について　システム／制御／情報 42 巻 1 号 pp. 2–6（1998）

追記：本解説第 3 回の Fig. 16 (a) で左上隅に「目標値 $r(t)$」とあるのは「外乱 $w(t)$」の誤です。お詫びして訂正します。