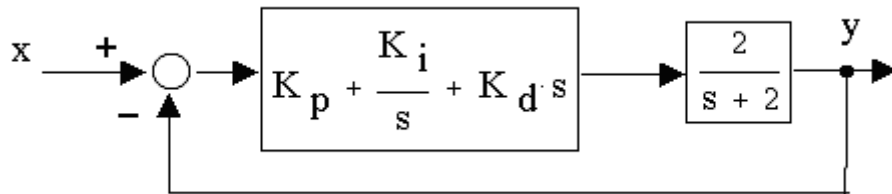


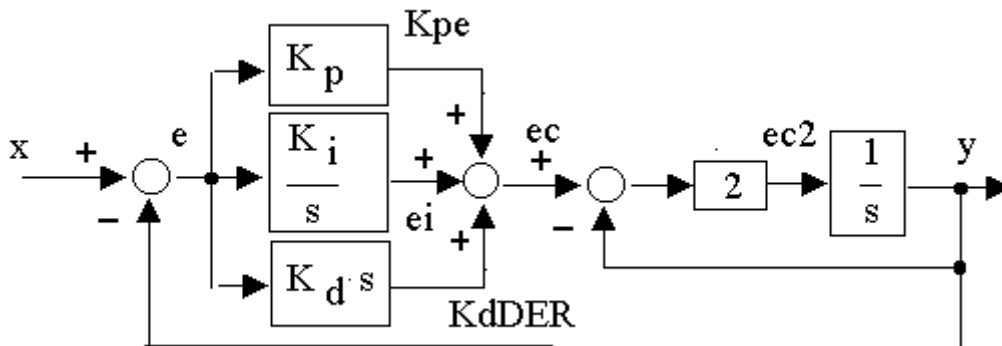
## PID 制御のシミュレーション

### 1 次遅れ系の PID 制御



まず、手始めに 1 次遅れ系の PID 制御をシミュレートしよう。

上図のようなシステムは次のように分解することで、数値の流れがよくわかり、BASIC でプログラムが書きやすくなる。



ステップ入力にたいしてシミュレートして見よう。

### プログラム

```
! PID Control of 1st Order system  
! Step input  
! pid00a.bas
```

```
set window -1,5,-1,2  
set axis color 1  
draw grid with scale (10,1)
```

```

let Kp=5
let Ki=5
let Kd=0.0
let dt=0.000051
let x=1
for Kd=0 to 1 step 0.5
  for t=0 to 2 step dt
    let e=x-y
    let ei=ei+Ki*e*dt
    let ec=Kp*e+ei+Kd*der
    let ec2=2*(ec-y)
    let y=y+ec2*dt

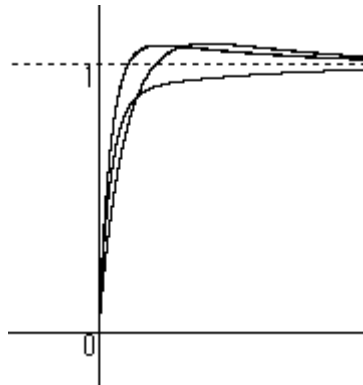
    let der=(e-ep)/dt
    if der>10 then let der=10
    if der<-10 then let der=-10

    let ep=e

    plot lines : t,y
  next t

  let y=0
next Kd
end

```



D 制御のゲインを変えてシミュレートさせて見た。 各制御のゲインを変えて試して見て下さい。

## DC モータの PID 制御

位置決め制御システムのシミュレートを行う。

( 実験とシミュレーションで学ぶモータ制御、見城、佐渡、他、日刊工業新聞社 2002年3月、p.193--- による )

パラメタを色々変えてその違いを見て下さい。



```

!PID01.bas
set window -0.1,1.2,-15,15
set axis color 1
draw grid(1,2)
let La=0.1
let Ra=2.1
let Re=100
let Jm=5*10^(-6)
let Kd=1
let Ke=0.02
let Kg=0
let Ki=0
let Kp=1
let Kt=0.02
let T1=0.01
let Amp=30
let H=0.1
let i0=0.2
let vbb=0.8
let iBo=0.2
let w0=0.01
let Dm=Ke^2/Re
let vin=7.5
let dt=0.0001
for t=0 to 1 step dt
  let epos=vin-H*w
  let ep=Kp*epos
  let ei=ei+Ki*epos*dt
  let ed=Kd*derivative
  let epid=ep+ei+ed

  let tac=tac+(60/2/pi*Kg*w-tac)/T1*dt
  let e2=epid-tac

  let vamp=Amp*e2
  if vamp>10 then let vamp=10
  if vamp<-10 then let vamp=-10

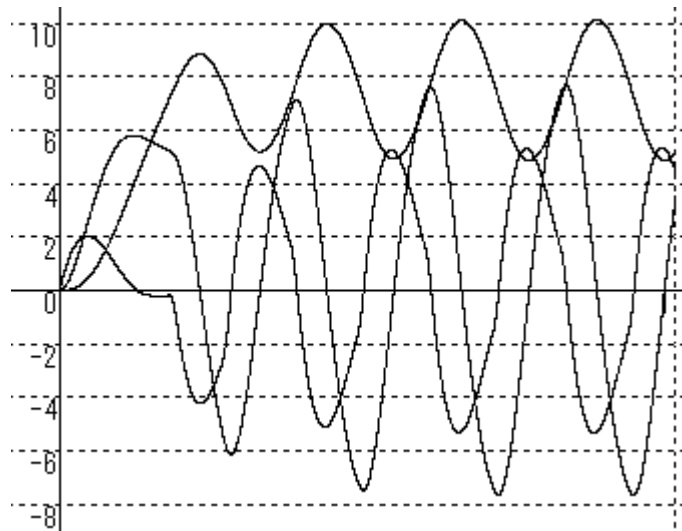
  let i=i+(vamp-vbrush-Ke*w-Ra*i)/La*dt
  let Torq=Kt*(i-ifrc)

  let w=w+(Torq-Dm*w)/Jm*dt
  let th=th+w*dt

  let ifric=i0*w/(w0+abs(w))
  let vbrush=vbb*i/(iBo+abs(i))

  let eprev=epos
  let derivative=(epos-eprev)/dt
  plot lines:t,w/100
  plot lines:t,i
  plot lines:t,th/10
next t
END

```



```

!*****position feed back
!*****P
!*****I
!*****D
!*****PID

```

```

!*****speed feed back

```

```

!*****limiter

```