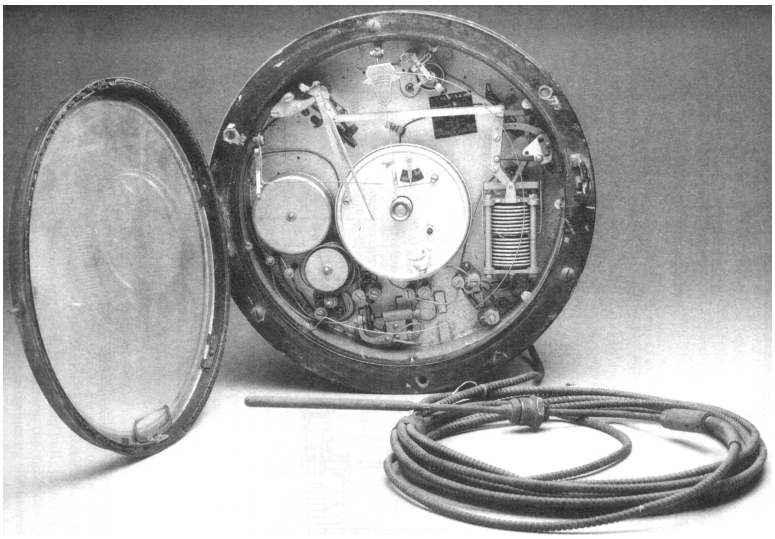
Implementación Programada de Reguladores



Técnicas de paso de manual a automático sin golpe y técnicas de antisaturación

Ramón Piedrafita Moreno

Área de Ingeniería de Sistemas y Automática

Escuela Universitaria de Ingeniería Técnica Industrial

Universidad de Zaragoza

Implementación Programada de Reguladores.

Técnicas de antisaturación de la acción integral.

---

1. Objetivos

La implementación programada de reguladores va mucho más allá que la simple expresión de un regulador en ecuaciones en diferencia. Los contenidos de este tema permiten analizar los principales aspectos prácticos (problemas prácticos) de los sistemas industriales de control por computador.

La presente lección tiene como objetivo principal introducir a los alumnos en dos de las problemáticas más importantes que se presentan en la implementación tecnológica de los Reguladores Industriales de acción PID:

- El cambio de control manual a control automático y
- El fenómeno de Windup.

Una vez descrito cada problema se procederá a detallar las técnicas existentes para corregirlo en la implementación programada, siguiendo el orden de menor a mayor que nos da la calidad de las soluciones adoptadas.

Dentro de las técnicas de cambio de control manual a control automático se estudian:

- Acción de régimen permanente predeterminada
- Seguimiento de la acción manual e
- Inicialización de la integral

El fenómeno de saturación de la acción integral supone uno de los problemas más importantes que se presentan en la implementación tecnológica de los Reguladores Industriales de acción PID. Prácticamente la totalidad de los reguladores industriales actuales disponen de alguna técnica de antisaturación de la acción integral.

Una vez se haya descrito cada problema se procederá a detallar las técnicas existentes para corregirlo en la implementación programada. Dentro de las técnicas de antisaturación se estudian:

- Banda integral
- Parada de la suma
- Resta integral y
- Algoritmo de velocidad

Cada una de estas técnicas será sometida a situaciones de control “límite” en presencia de perturbaciones, las cuales nos permitirán extraer información sobre su comportamiento y realizar una comparativa sobre su calidad. Por último se aborda el algoritmo de velocidad, el cual presenta la particularidad de que es solución de los dos problemas simultáneamente.

2. Introducción

En las instalaciones industriales es normal disponer de dos modos de funcionamiento o control: *manual y automático*.

En control *manual*, el operador mediante un mando en el panel de control, controla directamente la acción que se efectúa sobre el sistema, llevando la variable de salida hasta el régimen permanente o punto de trabajo deseado. En ese momento pasa la instalación a modo automático.

En control *automático* el sistema es controlado por el regulador PID, si el regulador no es convenientemente programado, la acción inicial del PID cuando se cambia de modo manual a automático, puede no ser la adecuada para mantener el régimen permanente, suponiendo un transitorio (golpe) sobre el sistema.

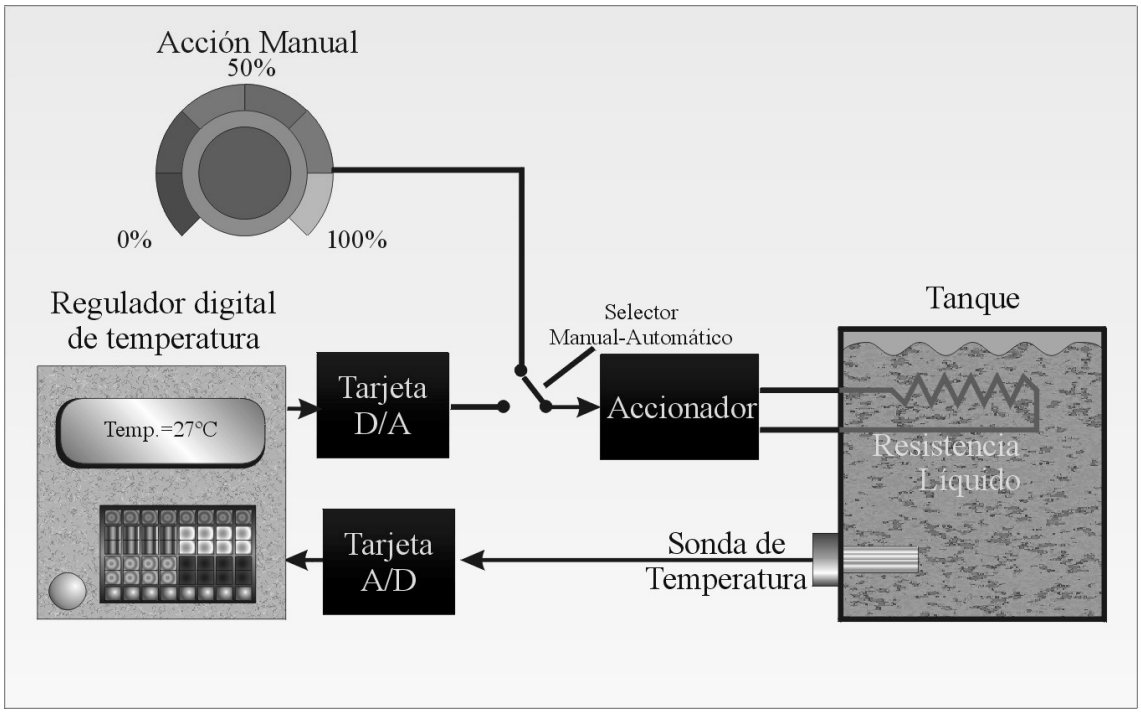


Figura 1. Control de Temperatura del Líquido de un Tanque.

El operador mediante el mando manual controla el nivel de tensión con que se alimenta las resistencias del tanque de la figura anterior, y lleva la temperatura del tanque al régimen permanente deseado.

Entonces efectúa el cambio a control automático y el regulador PID toma el control del sistema. Es deseable que en el cambio de modo de control no se produzca golpe de acción sobre el sistema, es decir, que la primera acción automática (calculada por el PID) sea igual a la última acción manual.

En los programas propuestos en la lección la técnica utilizada como método de cambio entre control manual y control automático es la inicialización de la integral.

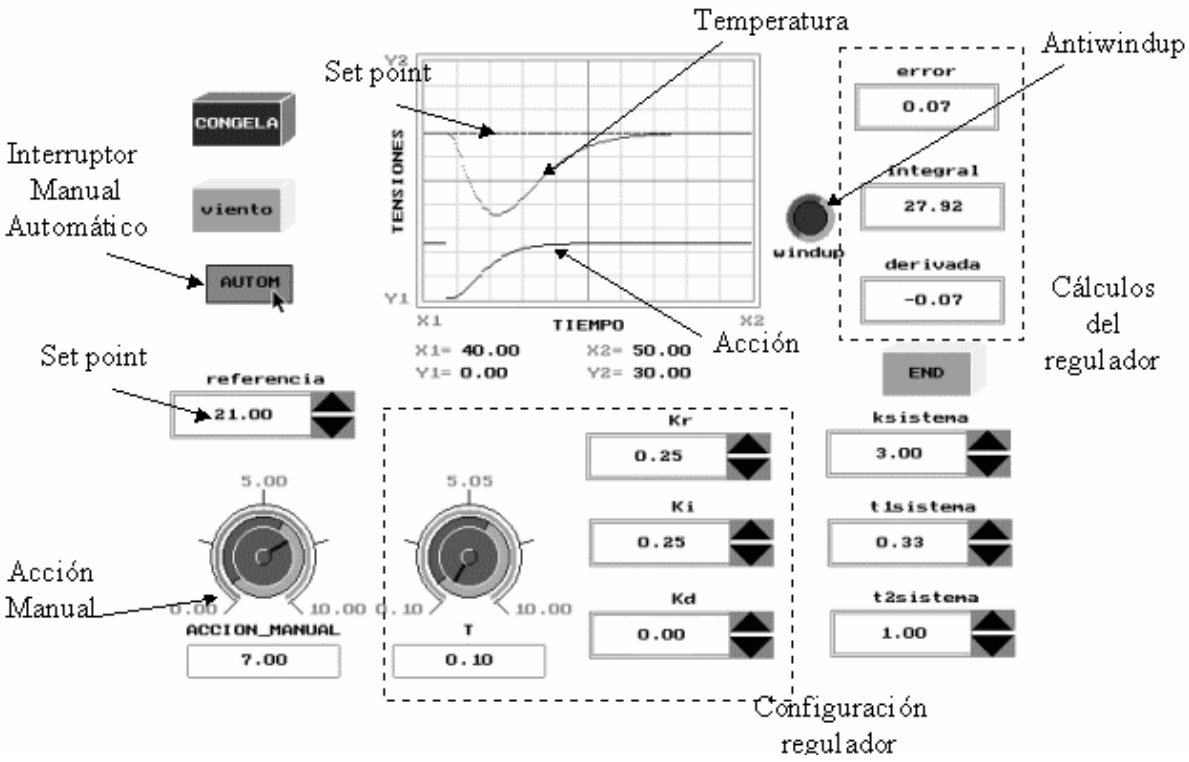


Figura 2. Panel de Control por Computador.

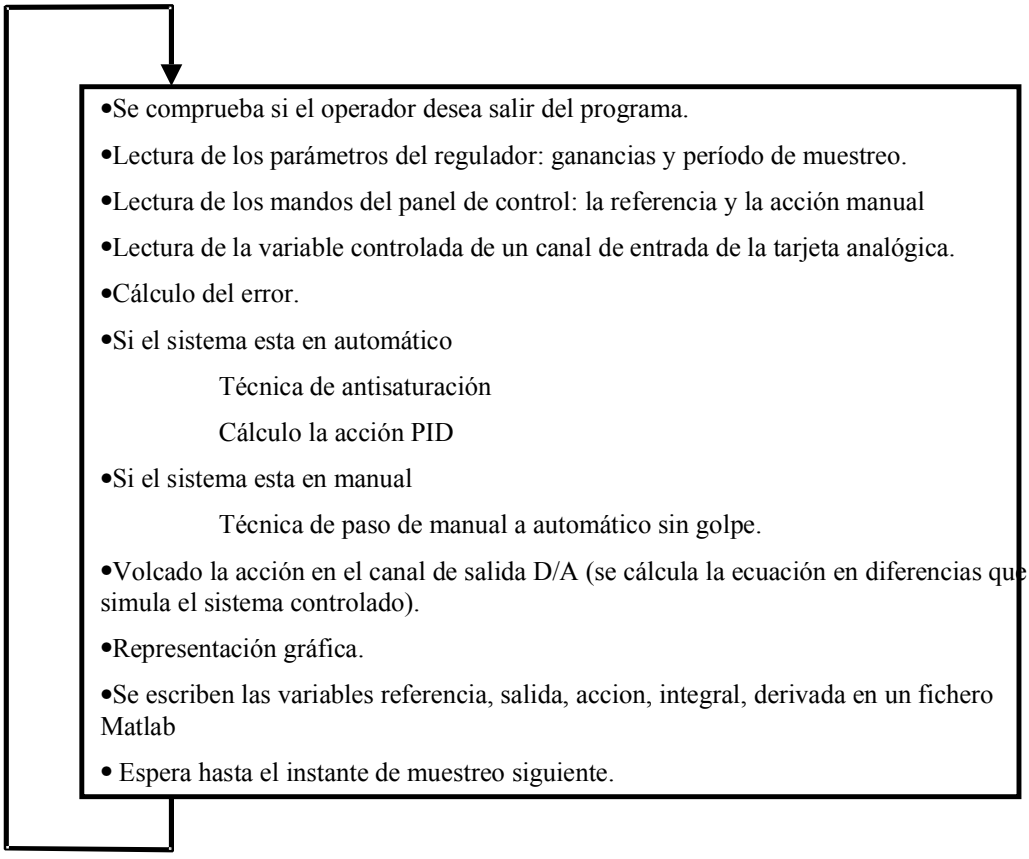


Figura 3 Bucle de Control.

```
/*lectura de la referencia automática*/
 referencia = leer_valor_ENTRADA_NUMERICA(refer);
/*lectura de la realimentación del canal A/D*/
salida= in_ADDA14(2);
/* cálculo del error*/
error = (referencia - salida);
if ( leer_estado_BOTON3D(AUTO) ==1)
{ /*si el control  esta en automático se calcula la acción automática*/
            integral=integral+T*error;
           if (T!=0)derivada=(error-errant)/T;
 /*cálculo de la acción automática*/
           accionaut=gankr*error+ganki*integral+gankd*derivada+M;
           out_ADDA14(accionaut);/*escritura de la acción en el canal D/A*/
           errant=error;
}
else /*el control está en estado manual*/
{          accionman = leer_valor_ROTATIVO(ROTA_MAN);
            out_ADDA14(accionman); /*escritura de la acción en el canal D/A*/
 /*cálculo de la inicialización de la integral. Técnica de cambio de manual a
automático*/
           if (ganki!=0) integral= (accionman-gankr*error-M)/ganki;
   }
```

Figura 4. Programación del control Automático y manual

3. Técnicas de transferencia de control manual a control automático

En las instalaciones industriales es normal disponer de dos modos de funcionamiento o control: *manual y automático*.

En control *manual*, el operador mediante un mando en el panel de control, controla directamente la acción que se efectúa sobre el sistema, llevando la variable de salida hasta el régimen permanente o punto de trabajo deseado. En ese momento pasa la instalación a modo automático.

En control *automático* el sistema es controlado por el regulador PID, si el regulador no es convenientemente programado, la acción inicial del PID cuando se cambia de modo manual a automático, puede no ser la adecuada para mantener el régimen permanente, suponiendo un transitorio (golpe) sobre el sistema.

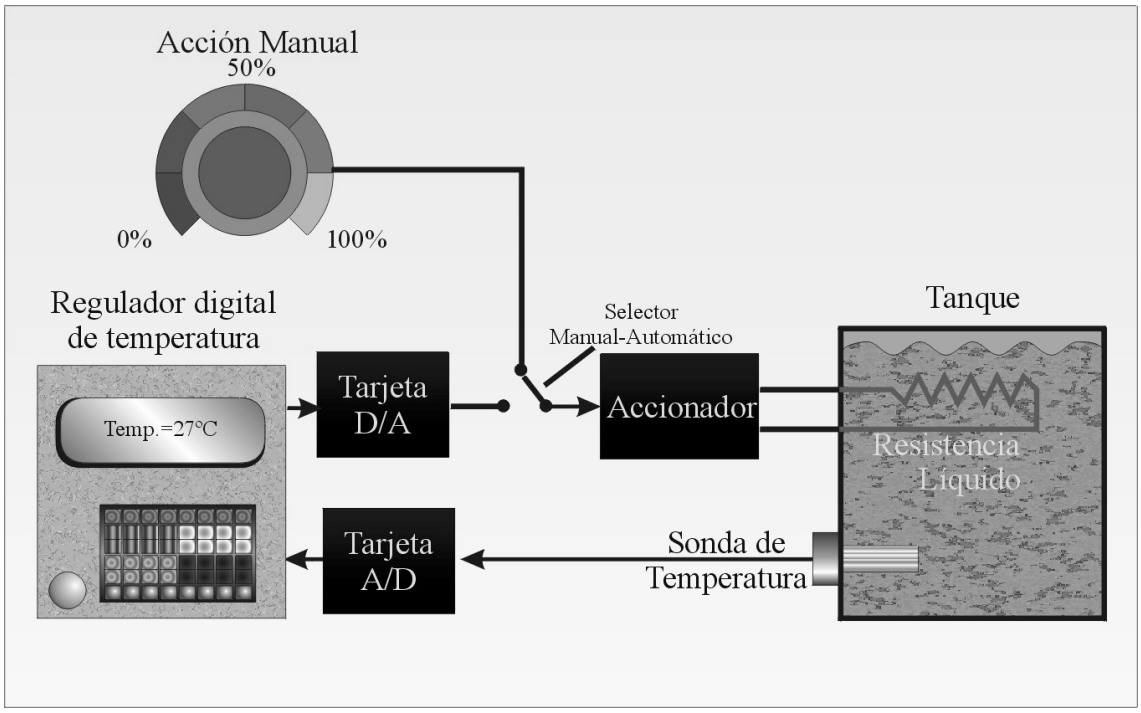


Figura 5. Control de Temperatura del Líquido de un Tanque.

El operador mediante el mando manual controla el nivel de tensión con que se alimenta las resistencias del tanque de la figura anterior, y lleva la temperatura del tanque al régimen permanente deseado.

Entonces efectúa el cambio a control automático y el regulador PID toma el control del sistema. Es deseable que en el cambio de modo de control no se produzca golpe de acción sobre el sistema, es decir, que la primera acción automática (calculada por el PID) sea igual a la última acción manual.

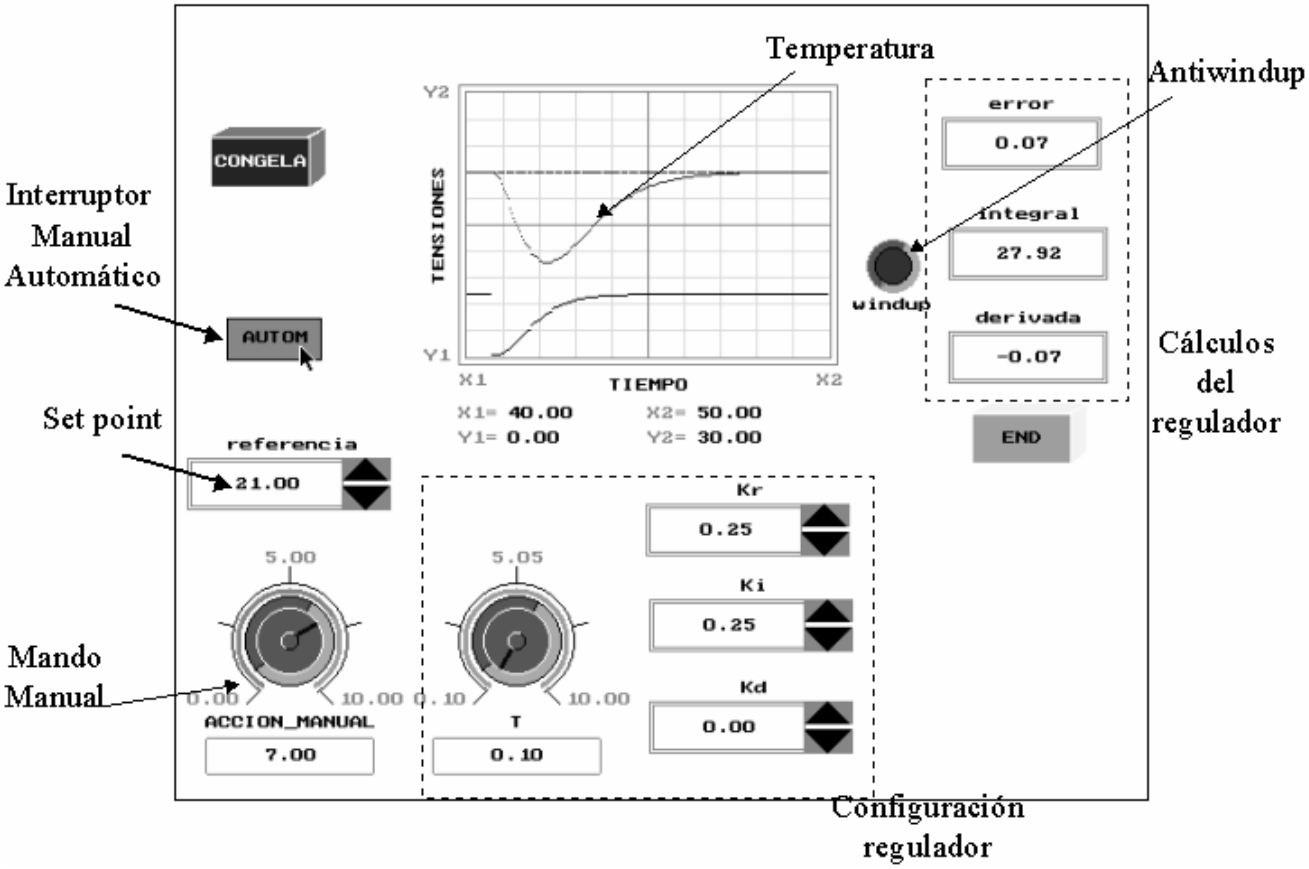


Figura 6. Panel de Control por Computador.

3.1 Golpe de acción sobre el sistema

En la figura anterior se observa un panel de control por computador programado en la pantalla de un PC. La gestión de los modos manual y automático la efectúa el propio programa de control.

- Si el sistema esta en modo manual la acción manual es escrita en el canal de salida digital/analógica.
- Si el sistema esta en modo automático la acción calculada por el regulador PID programado es escrita en el canal de salida digital/analógica.

En producción industrial, el cambio de control manual a control automático debe ser realizado sin golpes, de una manera suave.

Si el operador efectúa el cambio cuando el error del sistema es cero, es decir, cuando la variable de salida sea igual a la referencia del regulador, entonces:

```
integral=integral+T*error;
if (T>0) derivada=(error-errant)/T;
```

```
accion=gankr*error+ganki*integral+gankd*derivada;
errant=error;
out_ADDA14(accion);
```

Como el error es nulo error=errant=0, y se supone que el integrador empieza a integrar en ese momento integral=0. La acción inicial en modo automático será nula lo cual supone que la variable de salida disminuirá su valor y entonces el regulador corregirá la acción. No obstante esto supone un transitorio, un golpe de acción sobre el sistema.

```
integral=0+T*0;
if (T>0) derivada=(0-0)/T;
accion=gankr*0+ganki*0+gankd*0;
errant=error;
out_ADDA14(0);
```

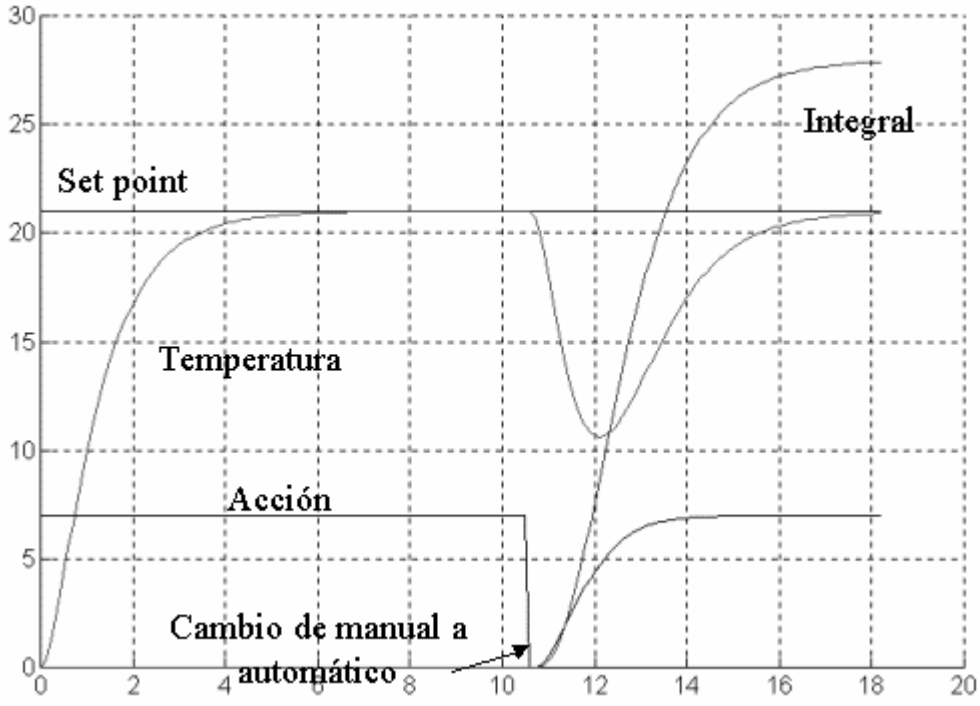


Figura 7. Golpe de acción sobre el sistema.

A continuación se describen tres métodos de corrección para conseguir una transferencia sin golpes.

3.2 Valor de M Predeterminado

En las ecuaciones en diferencias que definen el regulador PID:

```
integral=integral+T*error;
if (T>0) derivada=(error-errant)/T;
accion=gankr*error+ganki*integral+gankd*derivada;
errant=error;
out_ADDA14(accion);
```

en régimen permanente el error es cero y la acción será:

```
accion=ganki*integral;
```

Solo existe el término integral.

En algunas aplicaciones la acción de régimen permanente es cero, por ejemplo en control de posición de ejes. Pero en la mayor parte de las aplicaciones prácticas, el valor de la acción necesaria en régimen permanente no es nulo, por ejemplo en control de velocidad de ejes, control de temperatura. Para mantener la velocidad del eje de un motor a un valor de régimen permanente, la tensión aplicada al motor deberá ser distinta de cero. Para mantener la temperatura de un sistema a un valor de régimen permanente es necesario aportar o extraer una cantidad de calor del sistema. Hace falta una acción de régimen permanente distinta de cero.

Es una práctica habitual modificar la expresión del regulador PID para que recoja explícitamente el valor de la acción necesaria para mantener el régimen permanente (sin perturbaciones externas).

```
accion=gankr*error+ganki*integral+gankd*derivada+M;
```

M es el valor de la acción sobre el sistema en el punto de trabajo de régimen permanente. En los reguladores PID industriales M es introducido como un parámetro de programación. Si M no se introduce en la expresión del regulador, el término integral debe ser compensado. Pero en este caso es muy difícil conseguir un cambio suave entre el control manual y el automático. Incluso se pueden producir situaciones peligrosas, como aperturas a tope de válvulas o aceleraciones bruscas de motores.

El valor de M es calculado para una determinada carga. Es introducido como una constante en el programa o por el operador antes de producirse el cambio a modo automático. El paso a modo automático se produce cuando el valor del error es cero, la integral es cero y la acción m(n)=M.

Se remarca que también con un regulador proporcional puro puede existir error cero. Si por ejemplo M=7 y la ganancia estática del sistema controlado es 3, el régimen permanente deseado es 21.

Si el operador marca una referencia de 21 el error será cero y la acción será:

accion=gankr*0+7=7.

- Si el operador marca una consigna superior a 21 el error de posición será positivo.

$$Salida(\infty) = K * accion(\infty) = K * (gankr(referencia - Salida(\infty)) + M)$$

En la simulación K=3 (ganancia estática del sistema controlado)

Referencia=24 y M=7y gankr=0.25

Por lo cual

$$Salida(\infty) = K * accion(\infty) = 3 * (0.25(24 - Salida(\infty)) + 7)$$

$$Salida(\infty)(1 + 0.75) = 39$$

$$Salida(\infty) = 22.28$$

- Si el operador marca una consigna inferior a 21 el error de posición será negativo.

Referencia=18

Por lo cual

$$Salida(\infty) = K * accion(\infty) = 3 * (0.25(18 - Salida(\infty)) + 7)$$

$$Salida(\infty)(1 + 0.75) = 34.5$$

$$Salida(\infty) = 19.17$$

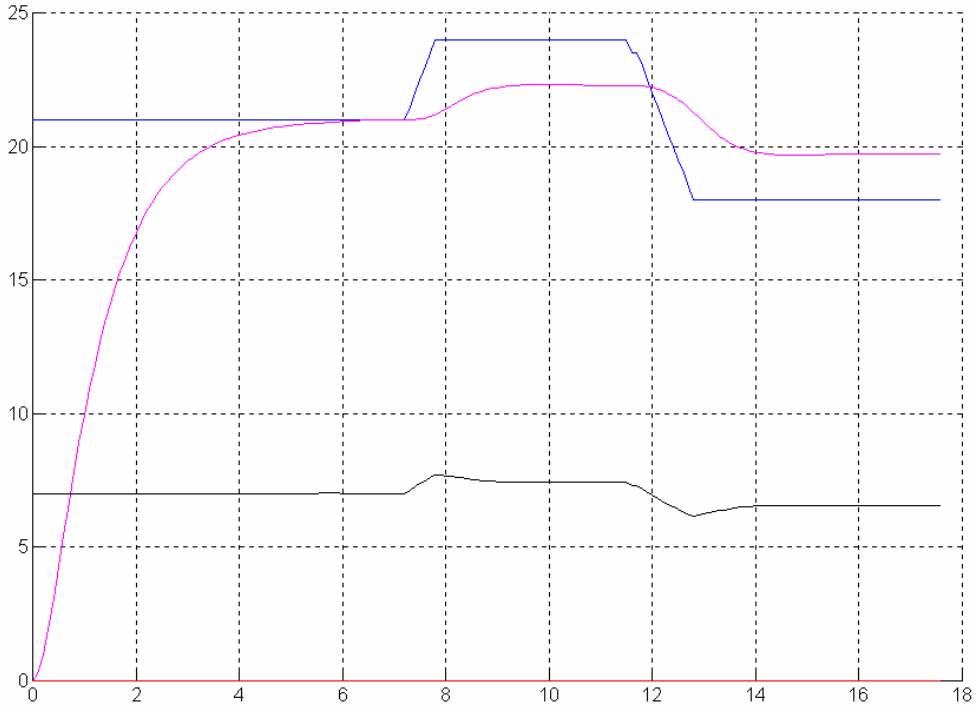


Figura 8. Errores de posición.

El problema de este método es que el valor de M es correcto para una determinado valor de la variable de salida y sin acción de perturbaciones externas. En otras

condiciones de operación puede no ser deseable que el cambio se produzca con el valor de M predeterminado.

Por ejemplo, en un control de temperatura de un horno, existen más perdidas de calor si la temperatura exterior son 5º C (Invierno) que si hay 40ºC (Verano). Es necesario suministrar más calor (más acción) en invierno que en verano.

Si el error no es cero en el cambio, se produce un golpe en la acción debido a la parte proporcional del regulador.

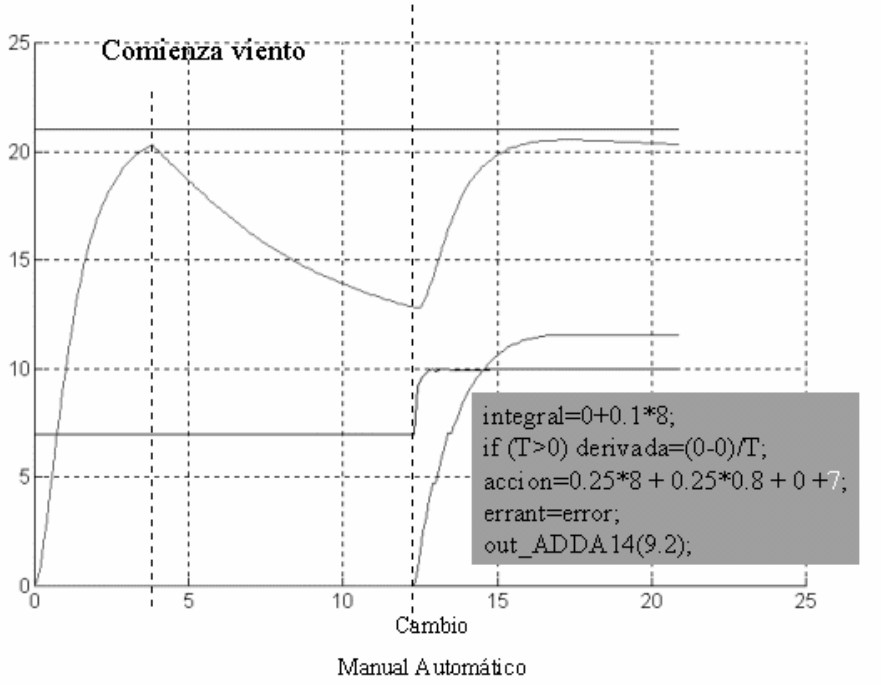


Figura 9. Cambio de manual a Automático

3.3 Seguimiento de la acción manual.

En el control manual el valor de la acción es determinado por el operador mediante el panel de control. El computador puede obtener el valor de la acción manual de diferentes formas:

- Mediante una entrada analógica del mando del panel de control.
- Leyendo la entrada del actuador del sistema.
- Si el panel de control esta implementado en el propio computador, el valor de la acción será ya una variable interna del computador.

En los dos primeros casos será necesario convertir y escalar las señales a las unidades utilizadas dentro del computador.

Suponemos que M no es una constante

M no esta predefinido y en esta variable se almacena el valor de la acción manual mientras el sistema se encuentra en modo manual.

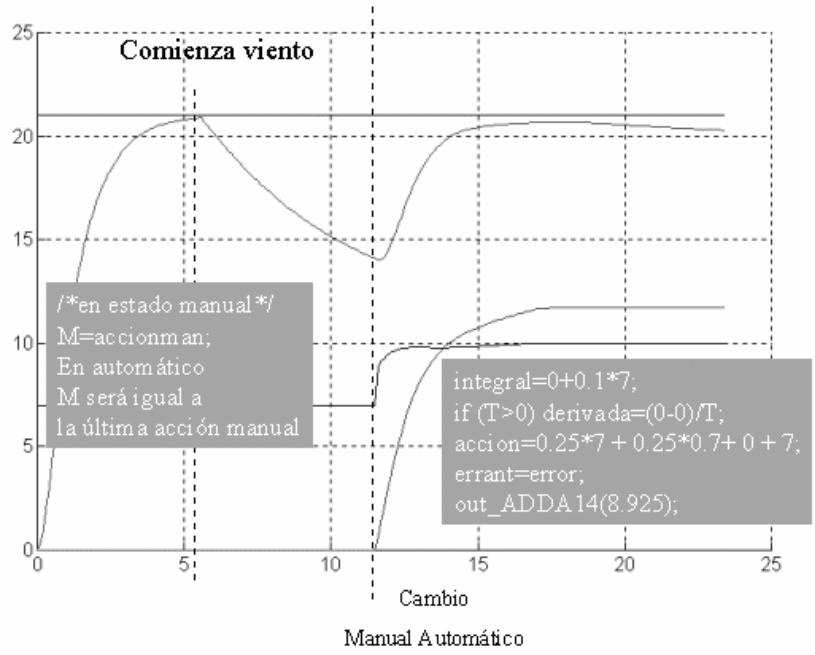


Figura 10 Seguimiento de la Acción Manual. Cambio con error distinto de cero.

3.4 Inicialización de la integral.

En este caso M esta predefinida al valor apropiado acción en régimen permanente entonces. El termino integral debe ser inicializado a un valor tal que:

$$\text{integral} = \frac{\text{accionman} - \text{gankr} * \text{error} - \text{M}}{\text{ganki}}$$

Mientras el sistema se encuentra en estado manual se ejecuta la siguiente instrucción

```
if (ganki!=0)
integral= (accionman-gankr*error-M)/ganki;
```

Cuando se produzca el cambio de manual a automático la variable integral no partirá de cero.

integral=integral+T*error;

*integral= (accionman-gankr*error-M)/ganki +T*error*

El valor de la primera acción automática calculada por el PID es:

accion=gankr*error+ganki*integral+M;

*accion=gankr*error+ganki* (accionman-gankr*error-M)/ganki + ganki + T*error+ M*

*accion= accionman+T*error*

Por loa tanto

*primera acción automática= última acción manual+T*error =7+0.1*5=7.5*

La primera acción automática es igual a la última acción manual más la primera integración del error. Por lo tanto el cambio se realiza sin golpe, a partir de ahora el regulador corregirá hasta alcanzar el régimen permanente deseado.

Se hace notar que el cambio se realiza sin golpe sea cual sea el error y la acción manual.

Bajo ninguna circunstancia de cambio, habrá golpe en la acción.

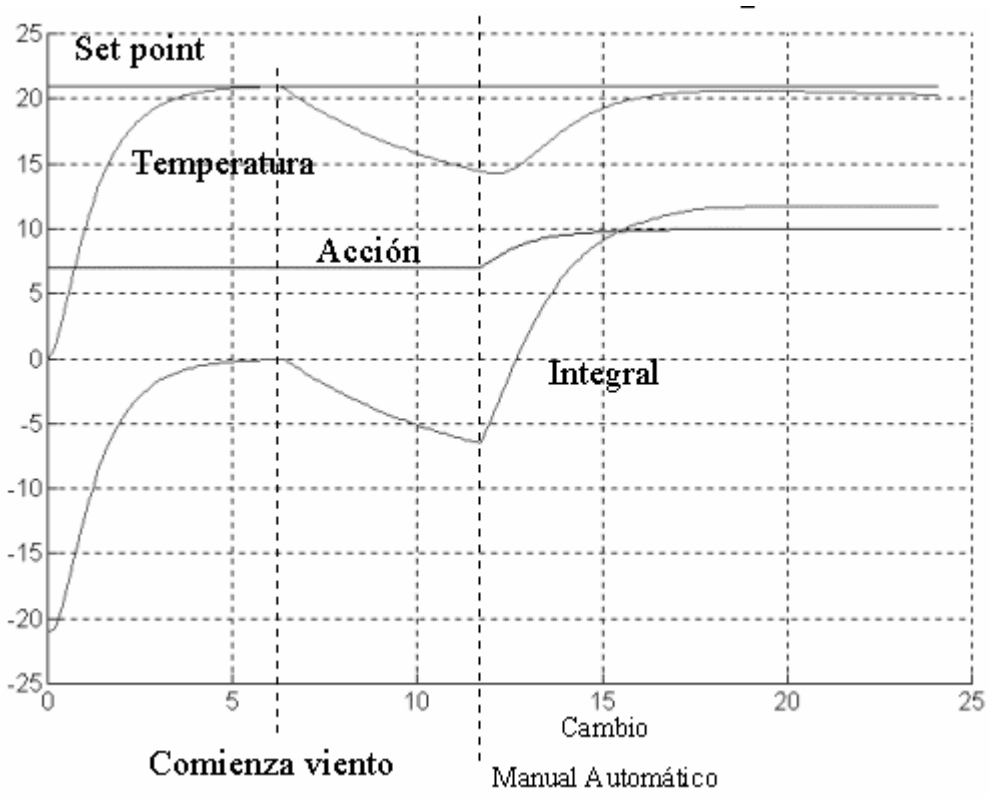


Figura 11. Inicialización de la Integral. Cambio con error distinto de cero.

4. Saturación de la acción integral.

En las aplicaciones prácticas el valor de la variable manipulada, la acción sobre el sistema esta limitada físicamente. Una válvula no puede ser abierta más que hasta cierto tope; una resistencia eléctrica que calienta un horno puede suministrar una cantidad máxima de calor por unidad de tiempo y no puede extraer calor del horno. Si el valor de la variable manipulada excede del valor máximo que el actuador puede suministrar, el control no se podrá realizar. En un buen diseño de control debería asegurarse que esto sólo ocurre en condiciones inusuales.

Un ejemplo sencillo puede ser considerar el sistema control de temperatura del líquido de un tanque. La capacidad del accionador es normalmente elegida para las condiciones de un invierno con temperaturas medias: Si coinciden temperaturas muy bajas con vientos gélidos, el sistema, aun actuando al máximo de su capacidad, no será capaz de mantener la temperatura deseada.

Con estas condiciones existirá un amplio margen de error en el control de la temperatura. Si empleamos un regulador PI, como el error es no se anula, el término integral continuará aumentando, esto es, el valor de la variable integral se incrementará en cada período de muestreo. Dicho con otras palabras la integral tiende a infinito.

Por lo tanto el valor de la acción se incrementará y la demanda de calor continuará aumentando, aunque por haber llegado al máximo no se podrá suministrar más calor. Los cambios de temperatura pueden observarse en la figura siguiente:

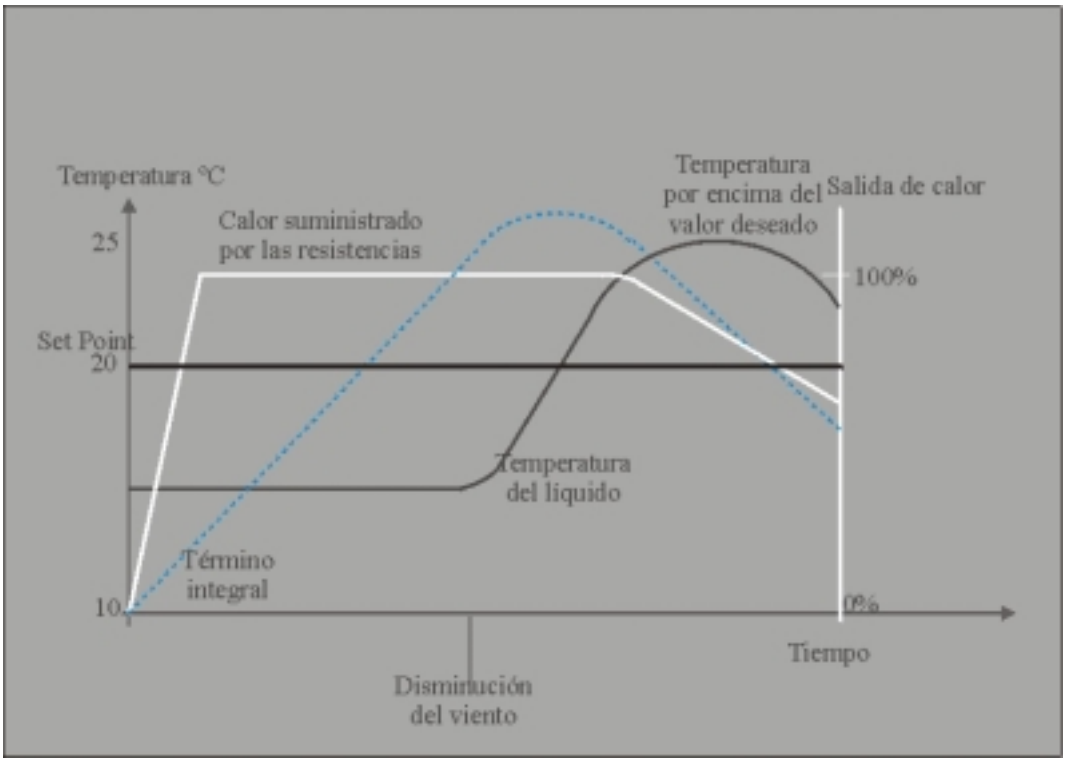


Figura 12. Control de temperatura del líquido de un tanque.

Si el viento disminuye y la temperatura exterior aumenta, la temperatura del líquido se verá incrementada y podrá llegar a la temperatura deseada. El valor del

término integral seguirá siendo demasiado grande hasta que no sea reducido mediante error negativo:

> *Cuando la temperatura del líquido sea mayor que la deseada, el error se vuelve negativo y el término integral empieza a integrar negativamente, disminuyendo su valor.*

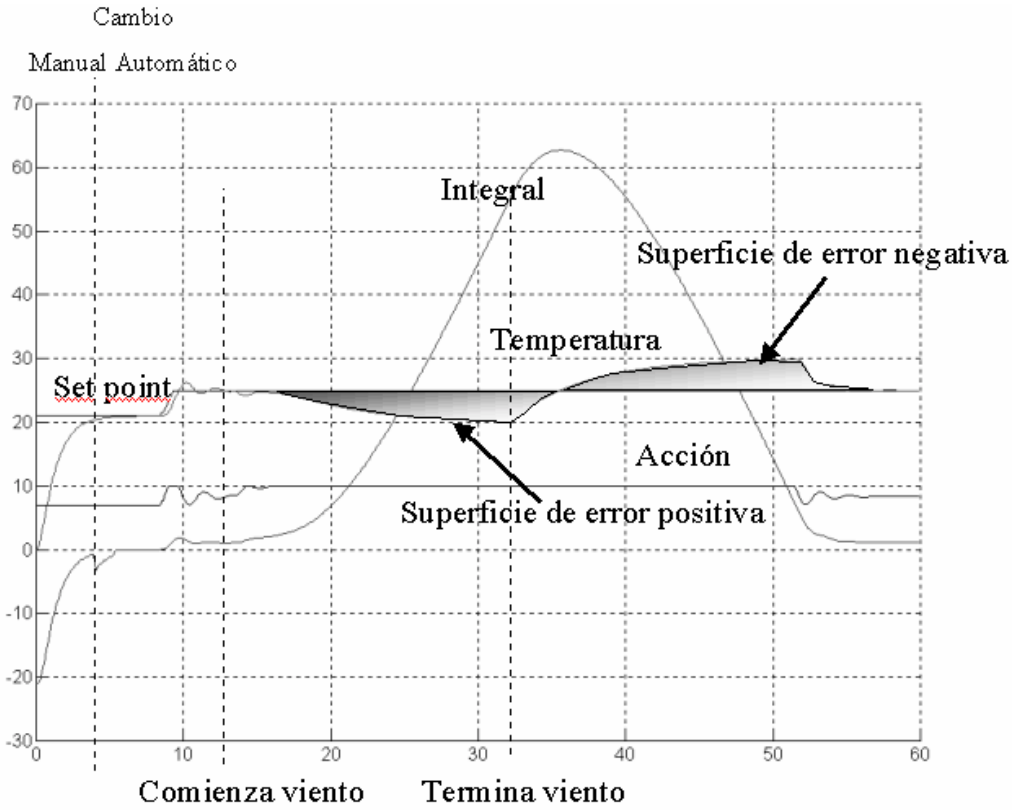


Figura 13. Windup

Por lo tanto el término integral continuará manteniendo la demanda de calor aun cuando la temperatura del líquido es ahora mayor que la deseada. Solo cuando el término integral se reduzca el sistema saldrá de saturación y se volverá a controlar el sistema.

La integral debe volver a tomar valores dentro de la "normalidad". En la simulación el Set point tiene un valor de 26. En régimen permanente la salida debe tomar ese valor:

$$Salida=26=K*Accion=K(ganki*integral+M)$$

Por lo tanto:

$$integral = \frac{Setpoint - K*M}{ganki*K} = \frac{26-3*7}{1.25*3} = 1.33$$

El efecto se denomina "saturación integral" o "Windup" y se da en los reguladores que tienen una pobre respuesta cuando llegan a una saturación extrema.

En esencia, el fenómeno windup ocurre cuando un regulador que posee acción integral encuentra un error sostenido por ejemplo en un gran cambio del punto de consigna o en el arranque de un proceso que posea un gran retardo.

Puede presentarse como consecuencia de una fuerte perturbación mantenida en la carga que impide el control del sistema dado que los accionadores, aún encontrándose al máximo no son capaces de reducir el error. El sistema de control no puede alcanzar el Set point debido a la presencia de la perturbación. En esta situación es poco deseable que la integral continúe aumentando, una vez que la acción del regulador ha alcanzado uno de los límites del actuador, ya que éste está haciendo todo lo que puede para reducir el error.

> *El fenómeno de saturación de la acción integral puede suponer la perdida de control del sistema durante un intervalo de tiempo que se puede afirmar que es del orden del tiempo de presencia de la perturbación. Esta perdida de control, supone siempre un sobrecalentamiento del sistema que podría suponer graves daños o incluso llevar a su destrucción.*

5. Técnicas de antisaturación de la acción integral

La mayoría, por no decir la totalidad, de los reguladores comerciales actuales llevan incorporados estrategias que proporcionan una desaturación del termino integral. Han sido desarrolladas varias técnicas para resolver el problema, las principales son:

- Banda integral.
- Parada de la suma al comenzar la saturación.
- Resta integral.
- Algoritmo de velocidad.

5.1 Banda Integral

En este método el regulador PI efectúa el cálculo de la acción integral si la variable controlada se encuentra en una banda en torno al Set Point. Por ejemplo, si esa banda es de 3 y el Set Point tiene un valor de 21; la integral se calcula si la variable controlada se encuentra entre 18 y 24. Definida en términos de error en la banda integral el error toma un valor máximo de 3 y un valor mínimo de –3. Fuera de la banda integral la acción integral no se aplica.

```
if((error>3) || (error<-3))
        bandaintegral=0;
else
        bandaintegral=1;
if (bandaintegral ==1)
        {integral=integral+T*error;
        accionaut=gankr*error+ganki*integral+gankd*derivada+M;}
else
        accionaut=gankr*error+gankd*derivada+M;
```

Figura 14. Programación de la banda integral

Es un método que en principio parece adecuado para resolver el problema, pero presenta en determinadas situaciones un comportamiento no muy aceptable. Un ejemplo es la situación representada en la figura siguiente.

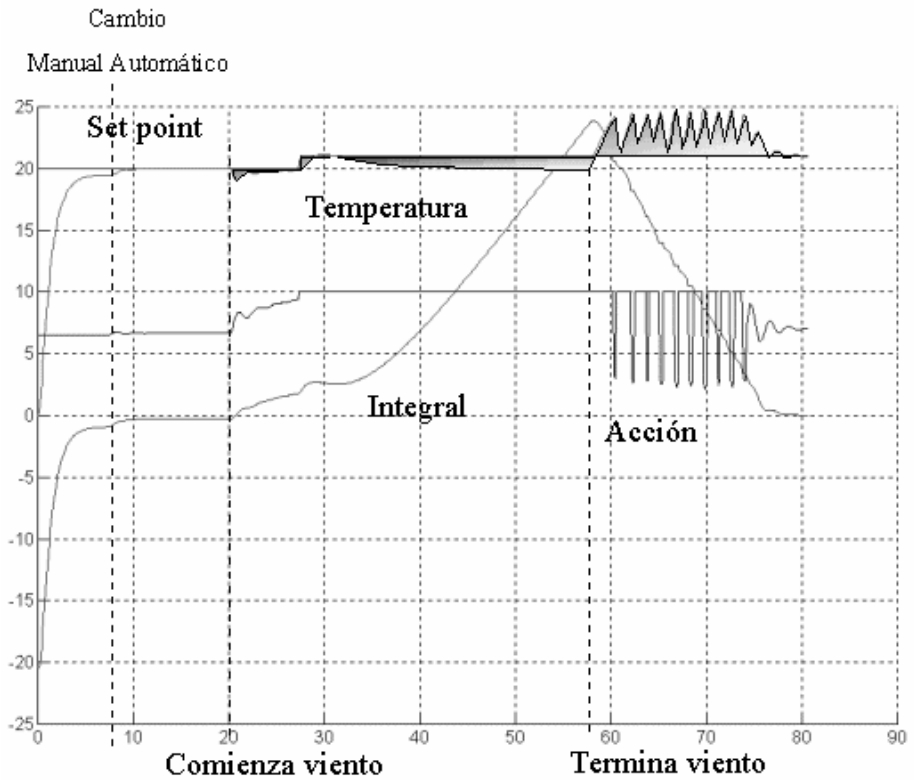


Figura 15. Banda Integral

Ante la presencia de la perturbación el regulador suministra acción máxima (10) pero no puede llegar al régimen permanente deseado y existe un error de posición de valor uno. Como la variable controlada se encuentra dentro de la banda integral, la integral se incrementa sin ningún impedimento. Cuando desaparece la perturbación, debido al alto valor de la integral, la variable controlada tiende a crecer por encima de la referencia. En el instante en que supera la banda integral (error>3) la parte integral de la acción no se aplica la acción disminuye a un valor:

$$Accion = gankr * error + M$$

$$Accion = 1.25 * (-3) + 7 = 3.25$$

La temperatura disminuye y como entra dentro de la banda integral la acción instantáneamente vuelve al valor máximo al aplicarse de nuevo la acción integral.

Se produce el efecto de dientes de sierra visualizado en la figura, sometiendo al sistema a unos fuertes y bruscos cambios en la acción

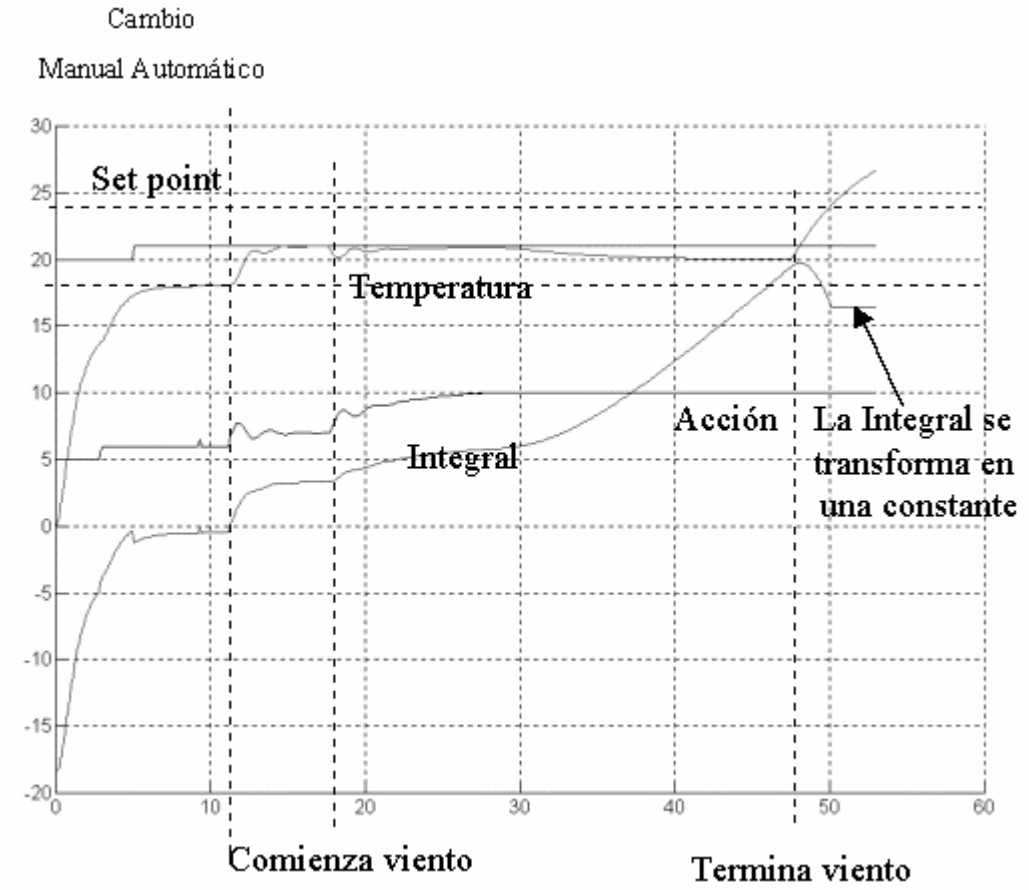


Figura 16. Perdida del control por aplicación de la acción integral fuera de la banda.

El lector podría suponer que esto es debido a que no se aplica la acción integral fuera de la banda integral. En caso de que se aplique, la respuesta es mucho peor, dado que una vez la variable controlada se encuentre fuera de la banda integral, no se disminuye la acción debido a que se sigue aplicando la acción integral y la propia integral no disminuye ante la presencia de error negativo, debido a que estamos fuera de la banda.

El regulador solo posee acción proporcional, dado que la integral se ha convertido en una constante y el sistema poseerá un error estático apreciable, superior a la banda integral.

Se puede concluir que no parece muy adecuado utilizar la banda integral que en definitiva es una ***medida de error*** *para limitar la acción que en régimen permanente cuando el error es cero solamente es* ***función de la superficie de error.***

5.2 Parada de la Suma

En este método el valor de la suma del integrador es congelado cuando el regulador se satura y el valor del integrador permanece constante mientras el regulador está en saturación. Una primera implementación de este método puede ser la congelación del término integral cuando la variable manipulada cae fuera del rango de valores $acción_{min} \div acción_{max}$.

```
if ((accionaut>10) || (accionaut<-10))
        STOPSUMA=1;
else
        STOPSUMA=0;
if (STOPSUMA==0)
        integral=integral+T*error;
if (T!=0)derivada=(error-errant)/T;
accionaut=gankr*error+ganki*integral+gankd*derivada+M;
```

Figura 17. Programación de la primera versión de la parada de la suma

El método tiene la desventaja de que el valor del término integral, cuando el sistema sale de la saturación, no se relaciona con la dinámica del sistema cuando éste funciona a pleno rendimiento. Esto se hace notar, sobre todo, en los reguladores que solo disponen de acción integral. Debido a que la acción esta producida por el término integral la acción no salé de saturación y se pierde el control tal como se observa en la figura siguiente.

```
if( (accionaut>10) ||(accionaut<-10))
        STOPSUMA=1;
else
        STOPSUMA=0;
if (STOPSUMA==0)
        integral=integral+T*error;

accionaut=ganki*integral+M;/* es constante*/
```

Figura 18. Fallo en el control.

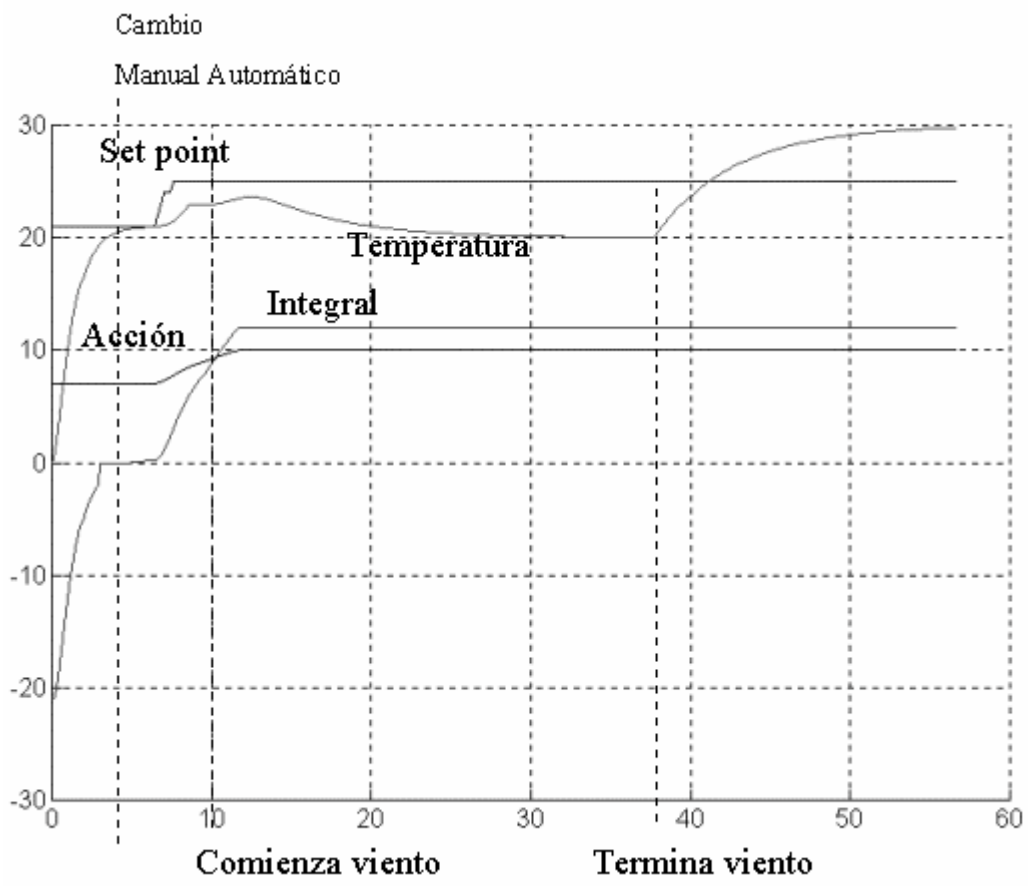


Figura 19. Perdida del control en la primera versión de la parada de la suma.

La integral está saturando el sistema. No se podrá salir de saturación dado que la parada de la suma esta activa y la integral no disminuye.

En los reguladores que poseen también acción proporcional, el fenómeno anterior no sucede debido a que en el momento en que desaparece la perturbación el error disminuye, disminuyendo por lo tanto la acción, la cual sale de saturación y la integral deja de estar parada al no estar activa la parada de la suma.

```
if(( (accionaut>10) && (error>0)) ||((accionaut<-10) && (error<0)) )
        STOPSUMA=1;
else
        STOPSUMA=0;
if (STOPSUMA==0)
        integral=integral+T*error;
if (T!=0)derivada=(error-errant)/T;
accionaut=gankr*error+ganki*integral+gankd*derivada+M;
```

Figura 20. Programación de la parada de la suma.

La técnica de la parada de la suma da una mejor respuesta si se tiene en cuenta también el signo del error. El término integral es congelado si acción esta saturada a su valor mínimo y el error es positivo o si la acción esta saturada a su valor mínimo y el error es negativo. Con esta mejora, se evitan fenómenos como el descrito anteriormente.

Actuador	Error	Suma integral
límite superior	+	parado
límite superior	-	activo
normal	+	activo
normal	-	activo
límite inferior	+	activo
límite inferior	-	parado

Figura 21. Condiciones de activación de la parada de la suma.

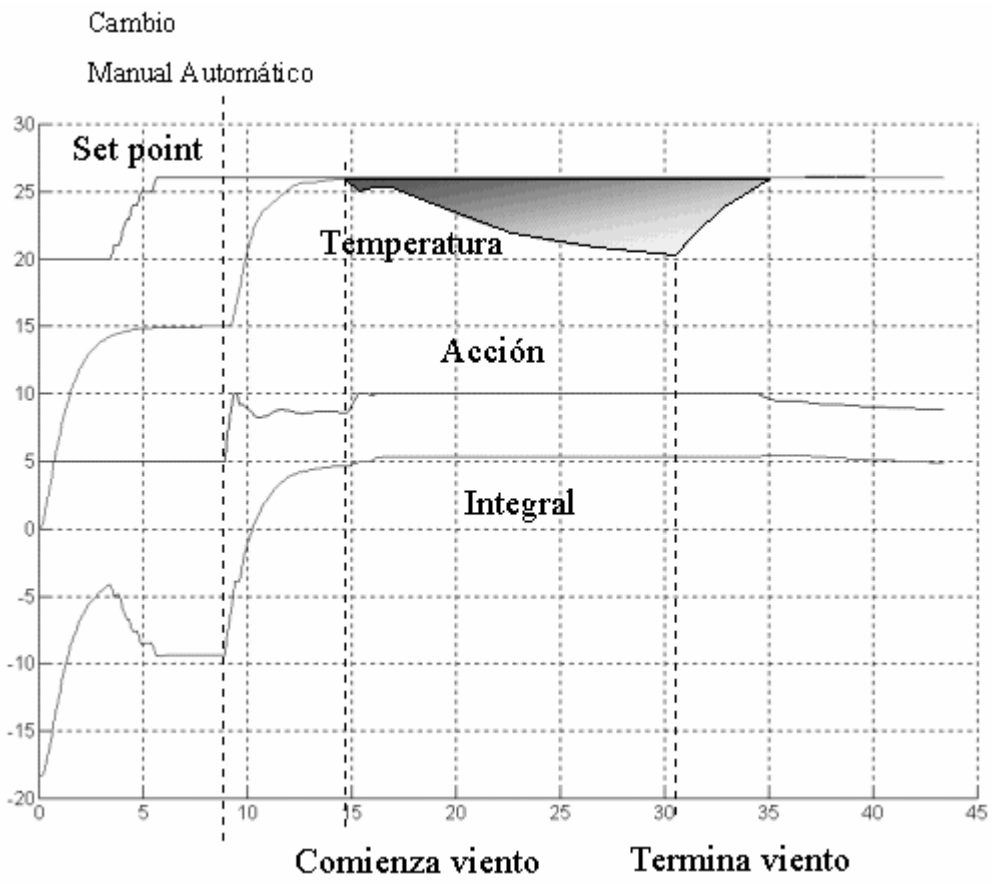


Figura 22. Parada de la suma.

En la simulación de la figura anterior el operador desea una temperatura de 26º C. Ante la presencia de la perturbación el sistema solo alcanza 20ºC. En la figura se observa una apreciable superficie de error positivo, en cambio la integral no crece.

Cuando desaparece la perturbación, y debido a que la acción esta a su valor máximo, la temperatura tiende al valor pedido por el operador. Como no se observa sobreelongación alguna, se podría afirmar que con esta técnica el problema se soluciona correctamente, pero no es del todo cierto. La respuesta observada en la figura depende del valor de los parámetros del regulador.

En la Figura 22 los valores de la ganancia proporcional y de la ganancia integral eran ambos 1.25. En la Figura 24 las ganancias valen 0.25

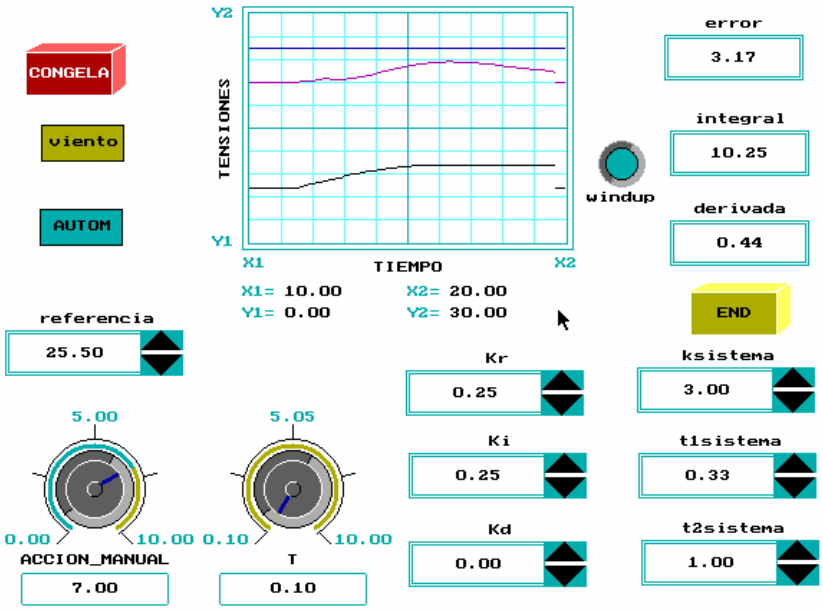


Figura 23. Condiciones de simulación

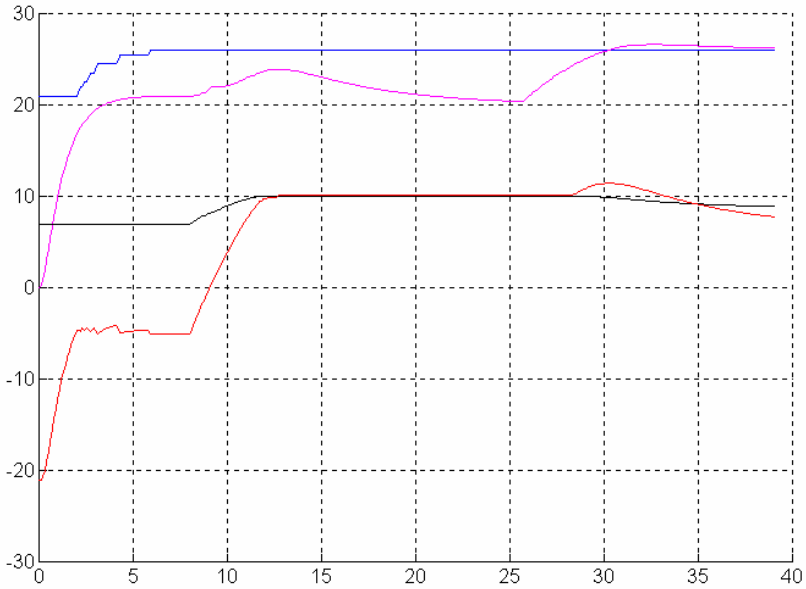


Figura 24. Sobreelongación

Cuando desaparece la perturbación, se puede producir una sobreelongación de la variable controlada sobre el Set Point. La duración de esta sobreelongación se puede afirmar que es inversamente proporcional al valor de la ganancia integral.

Cuando menor sea la ganancia de la acción integral ganki la integral puede tomar valores más altos, que luego se tienen que reducir mediante error negativo:

Sobreelongación.

En definitiva se puede afirmar que la técnica de la parada de la suma consigue solucionar el problema de la saturación de la integral, pero en su contra mantiene el valor de la integral a un valor elevado producido por la presencia de la perturbación. Lo cual retrasa la recuperación del control una vez ha desaparecido la perturbación.

5.3 Resta Integral

La idea de este método es que el valor integral es decrementado mediante una cantidad proporcional a la diferencia entre el valor calculado de la variable manipulada y el máximo valor permitido. La expresión integral de la suma:

integral=integral+T*error;

Cuando existe exceso de acción es sustituida por:

integral=integral+T*(error-kres*(accionaut-10));

La suma integral es así decrementada por el exceso de acción e incrementada por el error. El ritmo de decrecimiento depende de la elección del parámetro kres; si no es elegida adecuadamente podrá aparecer una continua oscilación saturación/desaturación.

```
if(accionaut-10>0)
  RESTAINT=1;
   else
  RESTAINT=0;
if (RESTAINT==0)
   integral=integral+T*error;
  else
   integral=integral+T*(error-kres*(accionaut-10));
accionaut=gankr*error+ganki*integral+M;
```

Figura 25. Programación de la resta integral.

En este caso el valor de la suma integral se decrementa tan pronto como el regulador entra en saturación y continúa decrementándose hasta que sale de la saturación, donde vuelve de nuevo la suma integral. Sin embargo, no hay un intento de acercar el término integral a los requerimientos del sistema y el valor de *kres* debe ser elegido por la experiencia antes que por referencia a las características del sistema.

Una mala elección de k supone que el sistema empiece a oscilar cuando se acerque a la saturación. Un ejemplo se visualiza en la figura siguiente: La simulación se ha realizado con un regulador con gankr=1.25 y ganki=1.25. El valor de kres era 60.

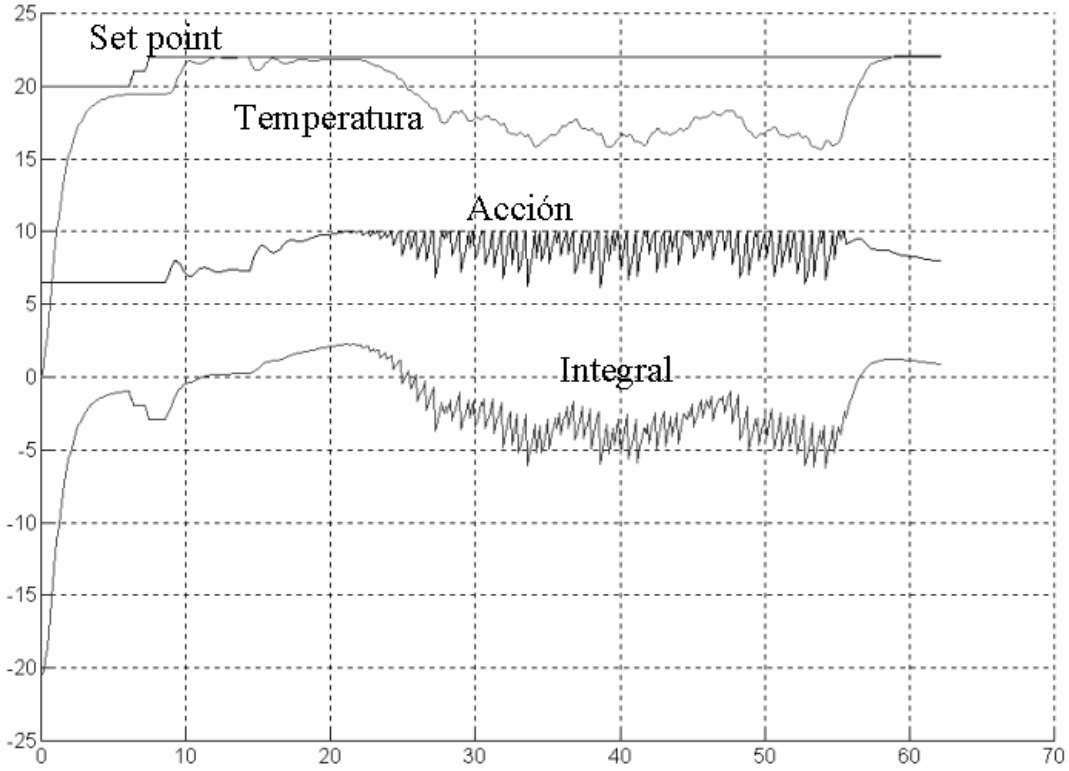


Figura 26. Oscilaciones en la Resta Integral

Se observa una fuerte oscilación en la acción, debido a los bruscos cambios de valores en la integral. En cambio se observa que al desaparecer la perturbación el sistema tiende al régimen permanente deseado sin sobreelongación. Pero esta propiedad, al igual que en la técnica de la parada de la suma es debida al alto valor de ganancia integral escogido.

En la figura siguiente se observa el control con kres=2. Se observa que la integral disminuye mientras el sistema se encuentra en saturación. La integral tenderá a un valor en el cual:

integral=integral+T*(error-kres*(accionaut-10))

Como la integral se estabiliza:

T*error-Tkres*(excesoacción)=0

$$\text{excesoaccion} = \frac{\text{error}}{\text{kres}}$$

La salida en presencia de la perturbación toma un valor máximo de 20. Como la referencia es 23, el error toma un valor de 3.

excesoaccion=1.5

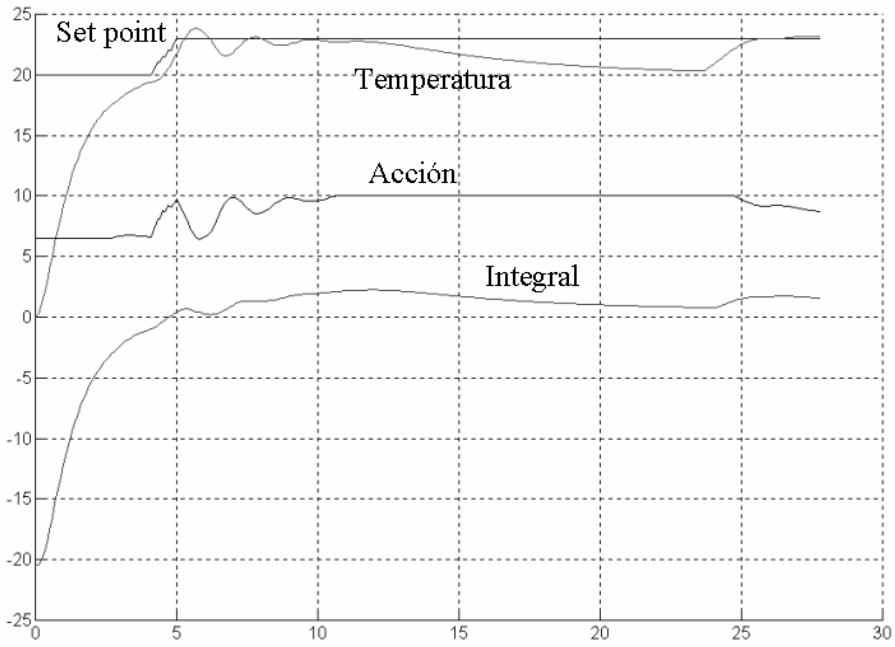


Figura 27. Resta Integral.

La acción automática toma el valor de

$$\text{accionaut= accionmax+excesoaccion}$$

$$\text{accionaut= 10+1.5=11.5}$$

Esta acción es igual a:

$$\text{accionaut=gankr*error+ganki*integral+M}$$

$$11.5=1.25*3+1.25*\text{integral}+7$$

$$\text{integral} = \frac{11.5 - 1.25 * 3 - 7}{1.25} = 0.6$$

la integral tiende al valor definido por

$$\text{integral} = \frac{\text{accionmax} + \frac{\text{errormax}}{\text{kres}} - \text{M - gankr} * \text{error}}{\text{ganki}}$$

Donde errormax es el error máximo en presencia de perturbación.

Cuanto menor sea el valor de ganki mayor será el valor de la integral, y por lo tanto mayor tiempo de sobreelongación. Cuanto mayor sea el valor de kres menor será el valor de la integral y por lo tanto menor tiempo de sobreelongación, pero también existe una mayor posibilidad de oscilaciones sobre el sistema.

El método puede modificarse para parar la suma de la parte del error durante la saturación si se dispone de una señal lógica del regulador que indique la saturación.

En este caso se consigue que el exceso de acción tienda a cero y la integral tiende a:

$$\text{integral} = \frac{\text{accionmax} - \text{M - gankr} * \text{error}}{\text{ganki}}$$

5.4 Algoritmo de velocidad.

El algoritmo PID se conoce como algoritmo de posición, dado que calcula la posición del actuador o el valor total de la acción sobre el sistema.

Existe una forma alternativa del algoritmo PID conocida como algoritmo de velocidad, que es usada para conseguir una transferencia sin golpes entre los modos manual y automático. El algoritmo de velocidad controlará la velocidad en el cambio de la acción sobre el sistema. El algoritmo de velocidad no calcula la acción sino que calcula el incremento de la acción.

Derivando respecto al tiempo la acción de un PID de posición.

$$\frac{\text{d accion}}{\text{dt}} = \text{gankr} * \frac{\text{d error}}{\text{dt}} + \text{ganki} * \text{error} + \text{gankd} * \frac{\text{d}^2\ \text{error}}{\text{dt}^2}$$

Calculando el incremento de acción entre dos instantes de muestreo

$$\Delta\,\text{accion} = \text{accion(k)} - \text{accion(k}-1)$$

$$\Delta\,\text{accion} = \text{gankr} * (\text{error(k)} - \text{error(k} - 1) + \text{ganki} * \Delta\text{t} * \text{error(k)} +$$

$$+ \frac{\text{gankd}}{\Delta t} * \left(\text{error(k)} - 2 * \text{error(k}-1) + \text{error(k}-2)\right)$$

$$\Delta\,\text{t} = \text{T}$$

$$\Delta\,\text{accion} = \text{gankr} * (\text{error(k)} - \text{error(k} - 1) + \text{ganki} * \text{T} * \text{error(k)} +$$

$$+ \frac{\text{gankd}}{\text{T}} * \left(\text{error(k)} - 2 * \text{error(k}-1) + \text{error(k}-2)\right)$$

Si el regulador no posee acción derivada el programa es:

$$\text{incraccion} = (\text{gankr} + \text{ganki} * \text{T}) * \text{error} - \text{gankr} * \text{errant}$$

El regulador mediante este algoritmo sólo controla los incrementos o decrementos de la acción sobre el sistema. Controlando que la velocidad de cambio de la acción no supere un máximo. De esta manera el cambio de automático a manual se realiza sin golpe.

Además, este algoritmo presenta la particularidad de evitar la saturación de la integral, lo cual es evidente dado que sólo calcula los incrementos de la acción.

La acción integral es obtenida en el actuador, que se encarga de sumar los incrementos de acción como por ejemplo en una servoválvula con motor paso a paso, por lo tanto esta acción esta sometida a las limitaciones físicas del propio actuador.

No existe ninguna razón que impida su aplicación para el cálculo del valor global de la acción. Solo habría que añadir las siguientes líneas al programa.

```
if(! ((accionaut>10) && (incraccion>0)))
                {accionaut=incraccion+accionaut;}
```

Si existe un error muy grande en el momento del cambio la respuesta del regulador puede ser muy lenta, sobre todo en el caso de que la ganancia integral sea muy pequeña.

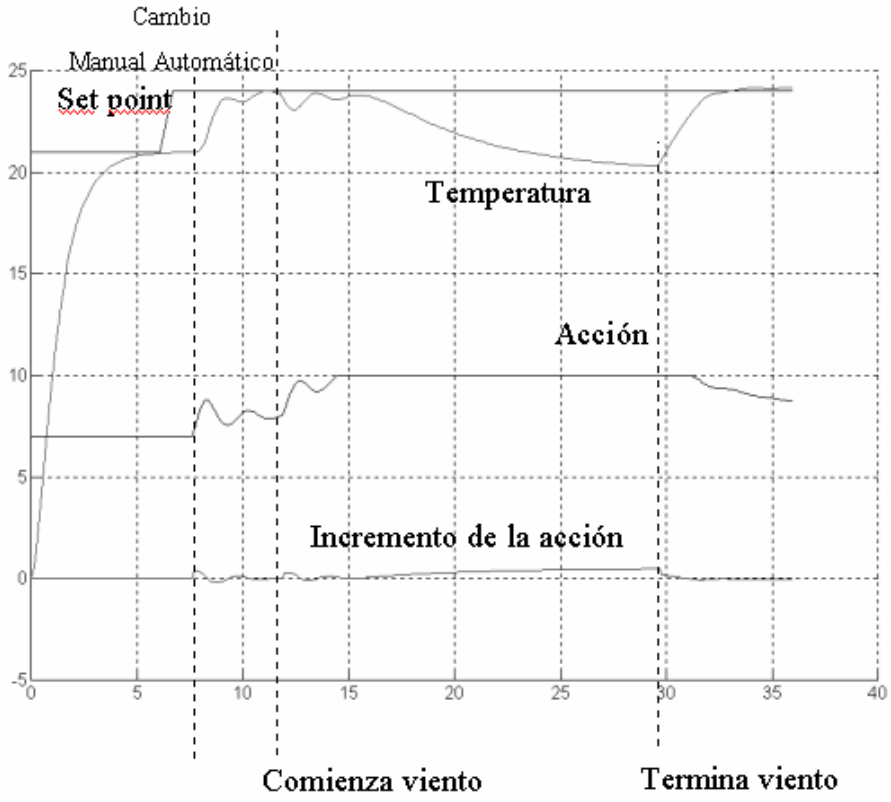


Figura 28. Algoritmo de velocidad

6. Conclusiones

El fenómeno de saturación de la acción integral puede suponer la perdida de control del sistema durante un intervalo de tiempo cuya duración es del orden del tiempo de presencia de la perturbación. Esta perdida de control, supone siempre un sobrecalentamiento del sistema que podría suponer graves daños o incluso llevar a su destrucción.

Las técnicas presentadas para evitar la saturación de la acción integral han evidenciado su calidad en situaciones límite, las que se dan en el control industrial.

La técnica de la banda integral ha evidenciado una pobre respuesta, y se afirma que no es muy adecuado utilizar como condición para parar el crecimiento de la integral una medida del error, cuando la propia integral es una medida de la superficie del error.

La técnica de parada de la suma se muestra en su segunda versión como una técnica eficaz, pero en su contra mantiene el valor de la integral al valor elevado producido por la presencia de la perturbación.

La técnica de la resta integral mejora el comportamiento de la técnica de la parada de la suma dado que integra el exceso de acción y por lo tanto disminuye la integral a un valor dentro de la *normalidad* similar al que debe tomar cuando no existe perturbación, es decir, en condiciones normales de operación. Esto permite que la recuperación del control una vez ha desaparecido la perturbación se efectúe de una manera más rápida.

En su contra, la elección del valor de kres se basa en la experiencia sobre la planta y una mala elección puede ser fatal para el sistema.

Los cambios bruscos en el valor de la acción sobre el sistema debido a que una condición cambia de valor, puede conllevar que el sistema entre en un proceso de oscilaciones no controlado. En vez de un cambio brusco debería mejorarse la técnica para que se diera un cambio gradual. Esto mismo también se podría aplicar en la técnica de la banda integral

La técnica del algoritmo de velocidad, se presenta como la más natural de todas, dado que no hay que establecer condiciones auxiliares a la ecuación que calcula la acción del controlador. El algoritmo de velocidad presenta una respuesta similar a la técnica de parada de la suma.

En su defecto, es una técnica que se suele utilizar solo en los casos en que el propio actuador realiza la función integral, por ejemplo, un motor paso a paso. No existe ninguna razón que impida su aplicación para el cálculo del valor global de la acción.

7. Bibliografía

Stuart Bennett: *Real-Time Computer Control. An introduction*. Prentice Hall International. 1994.

Sebastián Dormido, Fernando Morilla: *Autosintonía y métodos de Antiwindup en los reguladores PID*. UNED 1995.

Ramón Piedrafita, Antonio Romeo: *Control por Computador de Procesos Industriales*. Editorial Kronos. Zaragoza, 1999.

8. Simulación del sistema controlado en el programa de control.

El sistema controlado es un sistema de segundo orden sobreamortiguado con dos constantes de tiempo que pueden ser definidas por el usuario, así como su ganancia.

En la demostración realizada el sistema controlado no existe y en el ordenador no está insertada ninguna tarjeta de adquisición. Todo esto se sustituye mediante el cálculo en el ordenador de una ecuación en diferencias. Esta ecuación es el resultado de discretizar el sistema continuo: Bloqueador de orden cero- sistema controlado.

El proceso de discretización del sistema de segundo orden s expone a continuación:

$$B_o G(z) = Z[B_0(s)G(s)] = Z\left[\frac{1-e^{-Ts}}{s}G(s)\right] = \frac{z-1}{z}Z\left[\frac{G(s)}{s}\right]$$

$$B_o G(z) = \frac{z-1}{z}Z\left[\frac{K}{s(1+\tau_1 s)(1+\tau_2 s)}\right]$$

$$Z\left[\frac{K}{s(1+\tau_1 s)(1+\tau_2 s)}\right]$$

Haciendo uso de la Tabla de Transformadas Z.

$$Z\left[\frac{ab}{s(s+a)(s+b)}\right] = \frac{z}{z-1} + \frac{b}{a-b}\frac{z}{z-e^{-aT}} - \frac{a}{b-a}\frac{z}{z-e^{-bT}}$$

$$\frac{K}{s(1+\tau_1 s)(1+\tau_2 s)} = \frac{\frac{K}{\tau_1\tau_2}}{s\left(\frac{1}{\tau_1}+s\right)\left(\frac{1}{\tau_2}+s\right)}$$

$$a = \frac{1}{\tau_1}, b = \frac{1}{\tau_2}$$

$$Z\left[\frac{K}{s(1+\tau_1 s)(1+\tau_2 s)}\right]=Z\left[\frac{\frac{K}{\tau_1\tau_2}}{s\left(\frac{1}{\tau_1}+s\right)\left(\frac{1}{\tau_2}+s\right)}\right]=K\left(\frac{z}{z-1}+\frac{\frac{1}{\tau_2}}{\frac{1}{\tau_1}-\frac{1}{\tau_2}}\frac{z}{z-e^{-\frac{T}{\tau_1}}}-\frac{\frac{1}{\tau_1}}{\frac{1}{\tau_1}-\frac{1}{\tau_2}}\frac{z}{z-e^{-\frac{T}{\tau_2}}}\right)=$$

$$K\left(\frac{z}{z-1}+\frac{\tau_1}{\tau_2-\tau_1}\frac{z}{z-e^{-\frac{T}{\tau_1}}}-\frac{\tau_2}{\tau_1-\tau_2}\frac{z}{z-e^{-\frac{T}{\tau_2}}}\right)$$

$$B_oG(z)=\frac{z-1}{z}K\left(\frac{z}{z-1}+\frac{\tau_1}{\tau_2-\tau_1}\frac{z}{z-e^{-\frac{T}{\tau_1}}}-\frac{\tau_2}{\tau_2-\tau_1}\frac{z}{z-e^{-\frac{T}{\tau_2}}}\right)=K\left(1+\frac{\tau_1}{\tau_2-\tau_1}\frac{z-1}{z-e^{-\frac{T}{\tau_1}}}-\frac{\tau_2}{\tau_2-\tau_1}\frac{z-1}{z-e^{-\frac{T}{\tau_2}}}\right)$$

La entrada al sistema discretizado es la acción generada por el regulador discreto. La salida es la variable controlada.

$$Salida(z)=B_oG(z)*Accion(z)$$

$$Salida(z)=K\left(1+\frac{\tau_1}{\tau_2-\tau_1}\frac{z-1}{z-e^{-\frac{T}{\tau_1}}}-\frac{\tau_2}{\tau_2-\tau_1}\frac{z-1}{z-e^{-\frac{T}{\tau_2}}}\right)Accion(z)$$

$$Salida(z)\left(z-e^{-\frac{T}{\tau_1}}\right)\left(z-e^{-\frac{T}{\tau_2}}\right)=K\left(z-e^{-\frac{T}{\tau_1}}\right)\left(z-e^{-\frac{T}{\tau_2}}\right)+B(z-1)\left(z-e^{-\frac{T}{\tau_2}}\right)+C\left(z-e^{-\frac{T}{\tau_1}}\right)(z-1)Accion(z)$$

donde

$$B=K\frac{\tau_1}{\tau_2-\tau_1};C=K\frac{\tau_2}{\tau_1-\tau_2}$$

$$Salida(z)\left(z^2-\left(e^{-\frac{T}{\tau_1}}+e^{-\frac{T}{\tau_2}}\right)z+e^{-\frac{T}{\tau_1}}e^{-\frac{T}{\tau_2}}\right)=$$

$$K\left(\left(z^2-\left(e^{-\frac{T}{\tau_1}}+e^{-\frac{T}{\tau_2}}\right)z+e^{-\frac{T}{\tau_1}}e^{-\frac{T}{\tau_2}}\right)+B\left(z^2-\left(1+e^{-\frac{T}{\tau_2}}\right)z+e^{-\frac{T}{\tau_2}}\right)+C\left(z^2-\left(1+e^{-\frac{T}{\tau_2}}\right)z+e^{-\frac{T}{\tau_2}}\right)\right)Accion(z)$$

$$Salida(z)\left(z^2-\left(e^{-\frac{T}{\tau_1}}+e^{-\frac{T}{\tau_2}}\right)z+e^{-\frac{T}{\tau_1}}e^{-\frac{T}{\tau_2}}\right)=$$

$$\left((K+B+C)z^2-\left(K\left(e^{-\frac{T}{\tau_1}}+e^{-\frac{T}{\tau_2}}\right)+B\left(1+e^{-\frac{T}{\tau_2}}\right)+C\left(1+e^{-\frac{T}{\tau_1}}\right)\right)z+K\left(e^{-\frac{T}{\tau_1}}e^{-\frac{T}{\tau_2}}\right)+Be^{-\frac{T}{\tau_2}}+Ce^{-\frac{T}{\tau_1}}\right)Accion(z)$$

La ecuación en diferencias que simula al sistema controlado es:

$$Salida(k)=\left(e^{-\frac{T}{\tau_1}}+e^{-\frac{T}{\tau_2}}\right)Salida(k-1)-e^{-\frac{T}{\tau_1}}e^{-\frac{T}{\tau_2}}Salida(k-2)+(K+B+C)Accion(k)$$

$$-\left(K\left(e^{-\frac{T}{\tau_1}}+e^{-\frac{T}{\tau_2}}\right)+B\left(1+e^{-\frac{T}{\tau_2}}\right)+C\left(1+e^{-\frac{T}{\tau_1}}\right)\right)Accion(k-1)+\left(K\left(e^{-\frac{T}{\tau_1}}e^{-\frac{T}{\tau_2}}\right)+Be^{-\frac{T}{\tau_2}}+Ce^{-\frac{T}{\tau_1}}\right)Accion(k-2)$$

Las líneas de programa que efectúan el cálculo de la ecuación en diferencias son:

```
if (tau1!=0)      a=exp(-T/tau1);
if (tau2!=0)      b=exp(-T/tau2);
c=(tau2-tau1);
d=(tau1-tau2);}
e=a+b;
f=a*b;
g=tau1;
h=tau2;
if (tau1!=tau2)
salida1=(a+b)*salant-f*sal2ant+kso*(1+(g/c)+(h/d))*accion+kso*(-e-((1+b)*g/c)-
((1+a)*h/d))*accant+ kso*(f+g*(b/c)+ h*(a/d))*acc2ant;
        acc2ant=accant;
        accant=accion;
        sal2ant=salant;
        salant=salida1;
```

La presencia de una perturbación externa (viento) esta simulada mediante una señal de valor -10 que influye sobre la temperatura interna a través de una función de transferencia de primer orden de ganancia unidad y constante de tiempo de cinco segundos.

$$G_p(s)=\frac{K}{1+\tau s}$$

$$B_oG(z) = Z\left[B_0(s)G(s)\right] = Z\left[\frac{1-e^{-Ts}}{s}G(s)\right] = \frac{z-1}{z}Z\left[\frac{K}{s(1+\tau s)}\right] = K\frac{z-1}{z}\frac{z\left(1-e^{-\frac{T}{\tau}}\right)}{(z-1)\left(z-e^{-\frac{T}{\tau}}\right)} =$$

$$B_oG(z) = K\frac{\left(1-e^{-\frac{T}{\tau}}\right)}{\left(z-e^{-\frac{T}{\tau}}\right)}$$

$$salida_p(z) = K\frac{\left(1-e^{-\frac{T}{\tau}}\right)}{\left(z-e^{-\frac{T}{\tau}}\right)}Perturbacion(z)$$

$$salida_p(k) = e^{-\frac{T}{\tau}}salida_p(k-1) + K\left(1-e^{-\frac{T}{\tau}}\right)Perturbacion(k-1)$$

Las líneas de programa que efectúan el cálculo de la ecuación en diferencias es:

salidaper=exp(-T/5)*salidaper+1*(1-exp(-T/5))*perturbacion;

Dado que la perturbación cuando actúa es una constante no hemos considerado necesario incluir el retraso de un periodo de muestreo.

La salida será la suma total:

salida=salida1+salidaper;

8.1 LISTADO DE UN PROGRAMA DE SIMULACION

```
#include "centra2.h"
#include "..\\libreria\\adda14.h"
#include "..\\libreria\\reloj.h"
#include "..\\tc\\INCLUDE\\math.h"

/* simulacion*/
/* Técnica de antisaturación de parada de la suma*/
/*paso de manual a automático sin golpe*/
/*última técnica: inicialización del valor de la integral*/
/*el regulador calcula solo cuando la instalación esta en automático*/

void main (int argc, char *argv[])
{
double error, errant=0;
double tiempoprox,tiempo_base, tiempo, tiempo_x;  /* Variables usadas para el control del tiempo */
double gankr,ganki,gankd,referencia,integral=0,derivada=0,M,accionaut=0,accion, accant=0,acc2ant=0,accionman,T;
double salida=0,cont=0;
double salida1=0, salant=0,sal2ant=0;
double salidaper=0,perturbacion;
double kso,tau1,tau2;/*parametros sistema segundo orden*/
double a,b,c,d,e,f,g,h;
int STOPSUMA;
FILE *fichero;

CURVA c_ref, c_accion, c_salida;
inicializar_CURVA (&c_ref, YELLOW);
inicializar_CURVA (&c_accion, WHITE);
inicializar_CURVA (&c_salida, LIGHTGREEN);
```

```
/* Carga la pantalla */
INICIALIZAR_PANTALLA();
  fichero=fopen(argv[1],"w");
  fprintf(fichero,"a=[");

/* la acción de régimen permanente debe ser 7*/
/* para que la salida del sistema, cuya ganancia estática es 3, */
/*en régimen permanente y en ausencia de perturbaciones sea 21*/
M=7;
tiempo_base = reloj();
tiempo = reloj();

/* Bucle infinito de control */

for (;;)
{
/* Comprueba si se desea salir del programa */
if (leer_estado_BOTON3D (BT_FIN) ==1)
          {
          /*accion = 0.0; out_ADDA14(accion);*/
  /*se escribe las instrucciones para representar gráficas en Matlab*/
  fprintf(fichero,"];");
  fprintf(fichero,"\n");
  fprintf(fichero,"t=[0:0.1:");
  fprintf(fichero,"%f",cont-0.1);
  fprintf(fichero,"];");
  fprintf(fichero,"\n");
  fprintf(fichero,"hold on;");
  fprintf(fichero,"\n");
  fprintf(fichero,"plot(t, a(:,1), 'y')");
  fprintf(fichero,"\n");
  fprintf(fichero,"plot(t, a(:,2), 'g')");
  fprintf(fichero,"\n");
  fprintf(fichero,"plot(t, a(:,3), 'w')");
  fprintf(fichero,"\n");
```

```
fprintf(fichero,"plot(t, a(:,5), 'c')");
fprintf(fichero,"\n");
fprintf(fichero,"grid;");
fprintf(fichero,"\n");
fprintf(fichero,"hold off;");
fprintf(fichero,"\n");
/*y se cierra al fichero*/
fclose(fichero);
        return;
        }
 /*lectura de la referencia automática*/
        referencia = leer_valor_ENTRADA_NUMERICA(refer);

/*lectura de los parámetros del regulador y de T*/
        gankr = leer_valor_ENTRADA_NUMERICA(kr);
        ganki = leer_valor_ENTRADA_NUMERICA(ki);
        gankd = leer_valor_ENTRADA_NUMERICA(kd);

        T =leer_valor_ROTATIVO(sample);

/*no se lee la realimentación, la variable controlada se calcula por simulación*/
 /* salida= in_ADDA14(2);*/

 /* cálculo del error */
        error = (referencia - salida);

        if ( leer_estado_BOTON3D(AUTO) ==1)
          {
          /*si la instalación esta en automático se cálcula*/
          /*la acción automática*/

          /*Se programa la técnica de antisaturación de parada de la suma*/

          if(( (accionaut>10) && (error>0)) ||((accionaut<-10) && (error<0)) )
                STOPSUMA=1;
```

```
        else
            STOPSUMA=0;

    if (STOPSUMA==0)
            integral=integral+T*error;

    /*si el regulador no posee acción integral, la integral se inicializa a cero*/

    if (ganki==0)    integral=0;

    /*cálculo de la derivada*/
    if (T!=0)derivada=(error-errant)/T;

    /*cálculo de la acción automática*/
    accionaut=gankr*error+ganki*integral+gankd*derivada+M;
    accion=accionaut;
    errant=error;
    }
    else
    {
     accionman = leer_valor_ROTATIVO(ROTA_MAN);
    accion=accionman;

        /*cálculo de la inicialización de la integral*/
      if (ganki!=0)
            integral= (accionman-gankr*error-M)/ganki;
       }

/* limitación de la acción del regulador a -10,+10 */
if (accion > 10.0)
        accion = 10.0;
if (accion < -10.0)
         accion = -10.0;
/* ecuación en diferencias que simula el efecto de la perturbación*/
/* la perturbación actúa sobre la variable controlada como una entrada*/
```

```
/* de valor –10 a un sistema de primer orden de ganancia uno y constante*/
/* de tiempo de cinco segundos*/

if ( leer_estado_BOTON3D(perturba) ==1){
        perturbacion=-10;}
        else perturbacion=0;
salidaper=exp(-T/5)*salidaper+1*(1-exp(-T/5))*perturbacion;

/* ecuación en diferencias que simula el sistema de segundo orden*/

kso=leer_valor_ENTRADA_NUMERICA(ksistema);
tau1=leer_valor_ENTRADA_NUMERICA(t1sistema);
tau2=leer_valor_ENTRADA_NUMERICA(t2sistema);
if (tau1!=0)    a=exp(-T/tau1);
if (tau2!=0)    b=exp(-T/tau2);
c=(tau2-tau1);
d=(tau1-tau2);}
e=a+b;
f=a*b;
g=tau1;
h=tau2;
if (tau1!=tau2)
salida1=(a+b)*salant-f*sal2ant+kso*(1+(g/c)+(h/d))*accion+ kso*(-e-((1+b)*g/c)
- ((1+a)*h/d))*accant+ kso*(f+g*(b/c)+h*(a/d))*acc2ant;
        acc2ant=accant;
        accant=accion;
        sal2ant=salant;
        salant=salida1;
        salida=salida1+salidaper;


/* en cada periodo de muestreo se almacenan en el fichero los valores de las
variables */
        fprintf(fichero,"%f",referencia);
        fprintf(fichero," ");
        fprintf(fichero,"%f",salida," ");
```

```
                fprintf(fichero," ");
                fprintf(fichero,"%f",accion," ");
                fprintf(fichero," ");
                fprintf(fichero,"%f",error," ");
                fprintf(fichero," ");
                fprintf(fichero,"%f",integral," ");
                fprintf(fichero," ");
                fprintf(fichero,"%f",derivada,";");
                fprintf(fichero,"\n");


        /* Representación gráfica */
         tiempo_x = reloj() - tiempo_base;
        if(leer_estado_BOTON3D (CONGELA) == 0)
        {dibujar_coordenada_en_PANTALLA (&VISUALIZADOR1, &c_ref, tiempo_x,
referencia);
        dibujar_coordenada_en_PANTALLA (&VISUALIZADOR1, &c_salida,
tiempo_x, salida);
        dibujar_coordenada_en_PANTALLA (&VISUALIZADOR1, &c_accion,
tiempo_x, accion);
        escribir_valor_NUMERICO (&visorerror, error);
        escribir_valor_NUMERICO (&visorintegral, integral);
        escribir_valor_NUMERICO (&visorderivada, derivada);
           };
          cont=cont+0.1;


           /* Establece el periodo de muestreo a T */
          tiempo=tiempo+T;
          esperar_hasta (tiempo);


           }


        }
```

8.2 LISTADO DE UN PROGRAMA DE CONTROL

```
#include "centra2.h"
#include "..\\libreria\\adda14.h"
#include "..\\libreria\\reloj.h"

/* técnica de antisaturación de la parada de la suma*/
/* paso de manual a automático sin golpe*/
/* última técnica: inicialización del valor de la integral*/
/* el regulador calcula solo cuando la instalación esta en automático*/

void main
{
double error, errant=0;
double tiempoprox,tiempo_base, tiempo, tiempo_x;  /* Variables usadas para el control del tiempo */
double gankr,ganki,gankd,referencia,integral=0,derivada=0,M,accionaut=0,accion, accionman,T;
double salida,cont=0;

int STOPSUMA;

CURVA c_ref, c_accion, c_salida;
inicializar_CURVA (&c_ref, YELLOW);
inicializar_CURVA (&c_accion, WHITE);
inicializar_CURVA (&c_salida, LIGHTGREEN);

/* Carga la pantalla */
INICIALIZAR_PANTALLA();
M=3; /* la acción de regimen permanente debe ser 3*/

tiempo_base = reloj();
```

```
tiempo = reloj();

/* bucle infinito de control */

for (;;)
{
/* Comprueba si se desea salir del programa */
if (leer_estado_BOTON3D (BT_FIN) ==1)
        {
  /* se pone a cero la acción */
  accion = 0.0;
  out_ADDA14(accion);
  return;
        }
   /*lectura de la referencia automática*/
        referencia = leer_valor_ENTRADA_NUMERICA(refer);
   /*lectura de los parámetros del regulador y del período de muesteo*/
        gankr = leer_valor_ENTRADA_NUMERICA(kr);
        ganki = leer_valor_ENTRADA_NUMERICA(ki);
        gankd = leer_valor_ENTRADA_NUMERICA(kd);

        T =leer_valor_ROTATIVO(sample);

  /* lectura de la realimentación */
  /* se lee el canal 2 analógico/digital */
  salida= in_ADDA14(2);

  /* cálculo del error */
        error = (referencia - salida);

        if ( leer_estado_BOTON3D(AUTO) ==1)
          {
          /*Sí la instalación esta en automático se cálcula*/
          /*la acción automática*/
          /*Se programa la técnica de parada de la suma*/
```

```
if(( (accionaut>10) && (error>0)) ||((accionaut<-10) && (error<0)) )
        STOPSUMA=1;
   else
        STOPSUMA=0;

if (STOPSUMA==0)
        integral=integral+T*error;

/*si el regulador no posee acción integral, la integral se pone a cero*/
if (ganki==0)    integral=0;

/*cálculo de la derivada*/
if (T!=0)derivada=(error-errant)/T;

/*cálculo de la acción automática*/
accionaut=gankr*error+ganki*integral+gankd*derivada+M;
accion=accionaut;
errant=error;
}
else
{
 accionman = leer_valor_ROTATIVO(ROTA_MAN);
accion=accionman;

  /*cálculo de la inicialización de la integral*/
   if (ganki!=0)
         integral= (accionman-gankr*error-M)/ganki;
    }
  /*escritura de la acción en el canal de salida digital/analógico*/
out_ADDA14(accion);

/* Representación gráfica */
tiempo_x = reloj() - tiempo_base;
if(leer_estado_BOTON3D (CONGELA) == 0)
```

```
{dibujar_coordenada_en_PANTALLA (&VISUALIZADOR1, &c_ref, tiempo_x,
referencia);
dibujar_coordenada_en_PANTALLA (&VISUALIZADOR1, &c_salida,
tiempo_x, salida);
dibujar_coordenada_en_PANTALLA (&VISUALIZADOR1, &c_accion,
tiempo_x, accion);
escribir_valor_NUMERICO (&visorerror, error);
escribir_valor_NUMERICO (&visorintegral, integral);
escribir_valor_NUMERICO (&visorderivada, derivada);
   };
        /* Establece el periodo de muestreo a T */
  tiempo=tiempo+T;
  esperar_hasta (tiempo);

   }

}
```