



Sistemas de Control 2

Apunte de Cátedra: Controladores PID predictivo Acción Feedforward

Docente: Ing. Gabriel Yonatan Aguirre

Contenido

Introducción	1
Diseño de controladores PID predictivos	1
Implementación de controladores PID predictivos	2
Conclusiones	8
Referencias bibliográficas	8

Introducción

Esta documentación refleja algunos aspectos importantes a la hora de implementar un controlador predictivo del tipo PID. De manera que se deja en evidencia el diseño y como llevar a cabo mediante un ejemplo práctico.

Por otro lado, con la implementación del PD predictivo se presenta la opción de agregar una acción *Feedforward*.

Diseño de controladores PID predictivos

Los controladores predictivos se basan en emplear una predicción del error actual e implementar los controladores PID. Como punto fuerte establece que la implementación de este tipo de controladores se evita la demora del cálculo y actualización de la acción de control en el mismo periodo de muestreo.

Vamos a tomar el caso del PI predictivo para el análisis de implementación.

Arrancamos con la idea de la predicción del error dado por:

$$e_p(k) = e(k-1) + [e(k-1) - e(k-2)] \quad (1)$$

Mientras que la ecuación recursiva de un controlador PI se puede definir como:

$$u(k) = k_p e(k) + ak_p e(k-1) + u(k-1) \quad (2)$$

Donde la función de transferencia está dada por:

$$G_{PI}(z) = \frac{k_p(z+a)}{z-1} \quad (3)$$

El modelo establece que se reemplace el error actual de la ecuación (2) por el error predictivo de la expresión (1), con lo cual se obtiene:

$$u(k) = (2k_p + ak_p)e(k-1) - k_p e(k-2) + u(k-1) \quad (4)$$

$$u(k) = K_1 e(k-1) + K_2 e(k-2) + u(k-1) \quad (5)$$

Con esta ecuación recursiva se puede simplificar el sistema y obtener la FT necesaria para hacer el diseño del controlador:

$$G_{pIp}(z) = \frac{U(z)}{E(z)} = \frac{K_1 z + K_2}{z(z-1)} \quad (6)$$

Expresión idéntica al PI pero con un atraso incluido. Entonces con esta expresión se debe diseñar el controlador predictivo empleando LGR u otro método que se desee.

Mediante la expresión (6) se realiza el diseño por LGR o el método deseado.

Implementación de controladores PID predictivos

Para programar la ecuación de implementación de este controlador se busca llevar a cabo respetando el siguiente esquema.

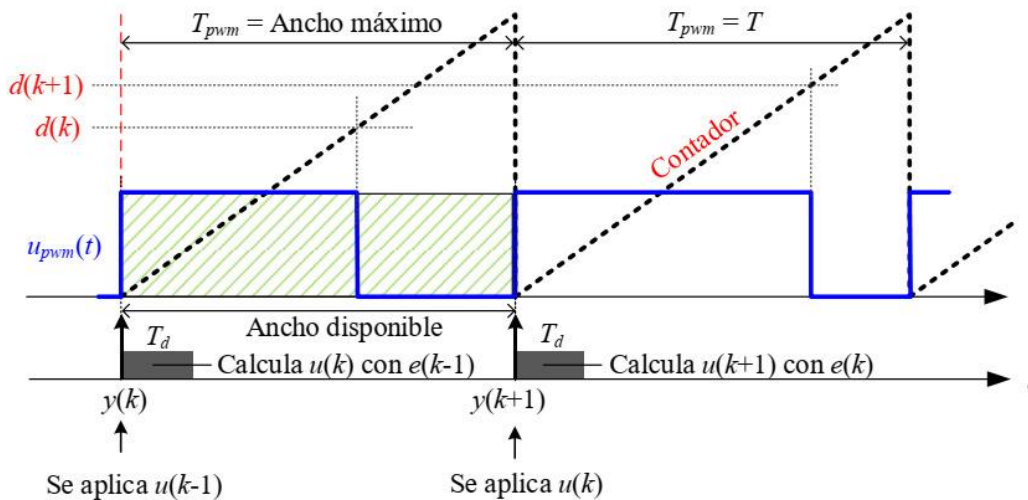


Figura 1: Esquema de implementación del controlador PID predictivo.

Al instante actual aplico primeramente la acción de control calculada anteriormente, es decir, $u(k-1)$ y luego calculo la acción de control a implementar en el siguiente periodo. De esta manera aprovecho al máximo el ancho del PWM.

Para el caso del PI predictivo plantemos el siguiente ejemplo:

$$G_p(s) = \frac{6,8571e^8}{(s + 2,795e^4)(s + 6816)} \quad (7)$$

Empleando un $T = 25\mu s$ obtengo la siguiente FT en tiempo discreto.

$$G_{pd}(z) = \frac{0,1621z + 0,1214}{z^2 - 1,341z + 0,4194} \quad (8)$$

Empleando el LGR con las especificaciones de $M_p = 1\%$ y $t_s = 0,5ms$ se obtiene el siguiente controlador.

$$G_{pIp}(z) = \frac{U(z)}{E(z)} = \frac{0,2517z - 0,2103}{z(z-1)} \quad (9)$$

La respuesta correspondiente se presenta en la siguiente figura.

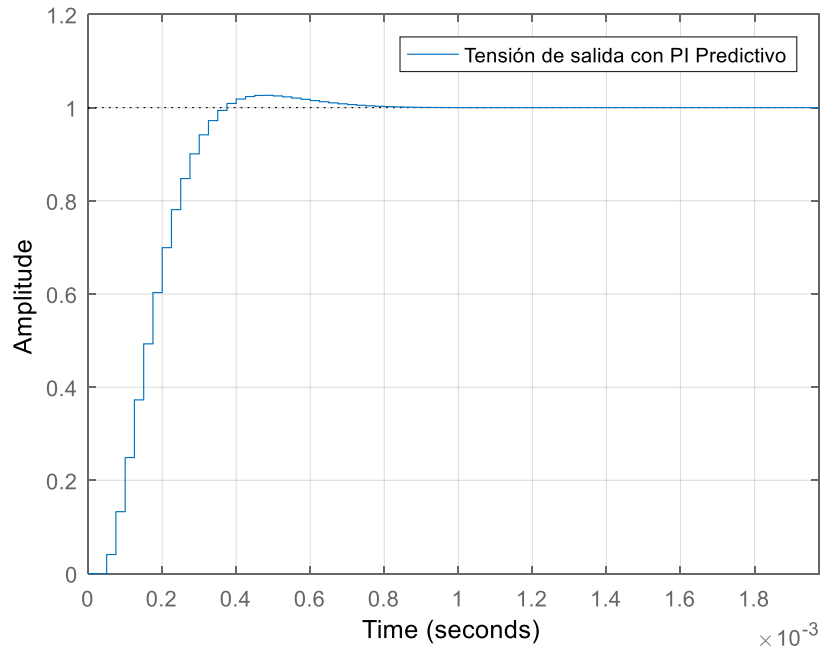


Figura 2: Respuesta al escalón unitario del sistema con el control PI pred.

Y la acción de control correspondiente será.

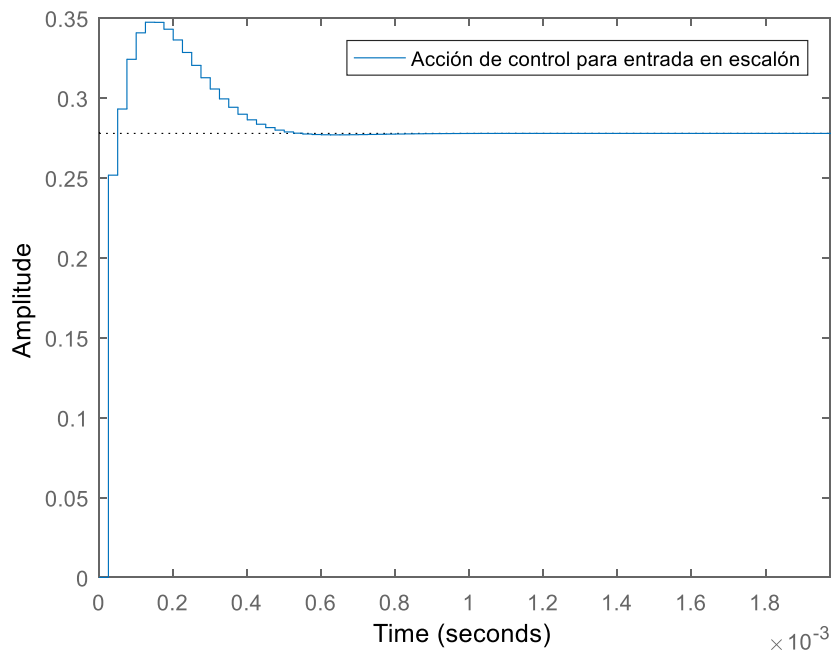


Figura 3: Acción de control del controlador PI pred sobre la planta.

Se observa que el sistema tarda un periodo en aplicar la acción de control, y 2 periodos en comenzar a variar la salida.

Ahora lo llevamos a la forma de implementar en el PSIM mediante DLL. El código utilizado en la DLL se presenta a continuación.

```
// Coeficientes del controlador PI pred:
#define K1 0.2517
#define K2 -0.2103
#define Kp -K2 //0.2103
#define a (K1-2*Kp)/Kp //-0.80314
#define OFF 0 // Apagado.
#define ON 1 // Encendido.
//****Función PSIM 2022/24**** Compile GCC 64 bit
void SimulationStep(double t, double delt, double *in, double *out){
    static double y=0, ref=0, ek=0, ek1=0;
    static double ekm1=0, ekm2=0;
    static double uk=0, ukm1=0;
    static double Ts;
    static int contador=0, Ncont, ComparReg;
    static char FLAG_calculo, FLAG_muestreo = ON;
    Ts = 0.000025;
    Ncont=Ts/delt;
    // Se realiza el muestreo de las variables controladas
    if(FLAG_muestreo == ON){
        FLAG_muestreo = OFF;
        y = in[1];
        ref = 1;
        FLAG_calculo = ON;
    } //termina el muestreo
    // Calculo de la estrategia de control
    if(FLAG_calculo == ON){
        FLAG_calculo = OFF;
        out[2] = uk; //Se actualiza uk empleando el valor anterior
        ek = ref - y;
        // ***** Método usando ec originales *****
        //ek1=2*ekm1-ekm2;
        //uk=Kp*ek1+a*Kp*ekm1+ukm1; //ec del PI tradicional (ec 2)
        // Modificadas para implementación (usar muestra actual)
        //ek1=2*ek-ekm1; // (ec 2)
        //uk=Kp*ek1+a*Kp*ek+uk; //ec del PI tradicional ant
        //*****Usando la ec rec del PIpred*****
        // uk=K1*ekm1 + K2*ekm2 + ukm1 // (ec 5)
        // Modificada para ser implementada en el siguiente periodo
        uk=K1*ek + K2*ekm1 + uk; //pisa el valor de uk
        // Almacena variables anteriores para próximo periodo.
        ukm1 = uk; ekm2 = ekm1; ekm1 = ek;
    } //termina el cálculo del control
    contador = contador + 1; // Incrementamos el contador
    if (contador == 0){
        FLAG_muestreo = ON;}
    if (contador == Ncont - 1){
        contador = -1;}
    // Se actualizan los valores de las salidas
    out[3] = ref;
    out[4] = y;
    out[5] = ek;
}
```

Básicamente sucede que el atraso incluido en el PI se implementa por hardware, por lo tanto es necesario quitar dicho atraso de la ecuación recursiva. Similar a como se incluía el atraso en la planta.

Las respuestas temporales del sistema y de la acción de control se presentan a seguir.

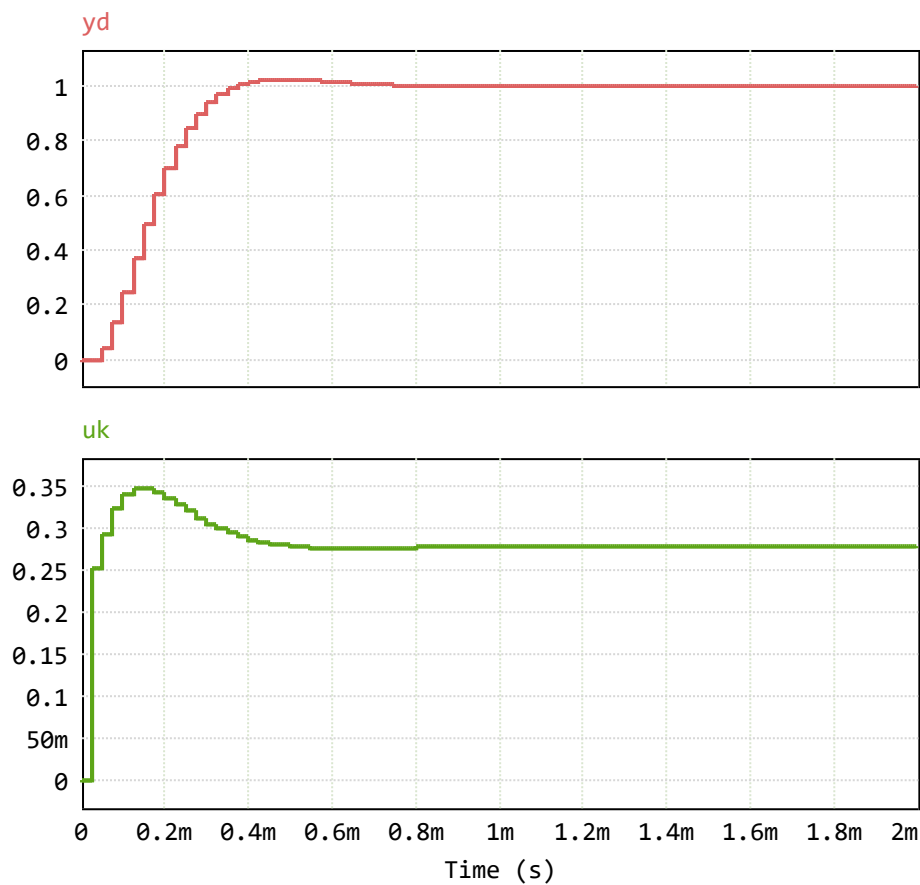


Figura 4: Respuesta del controlador PI pred empleando PSIM y la dll.

Otra manera de implementar el cálculo sería la siguiente:

```
ek = ref - y;
uk = ukm1 + Kc*ekm1 - aKc*ekm2;
ukm1 = uk; ekm2 = ekm1; ekm1 = ek;
out[2] = ukm1; // Se realiza la actualización de la
// acción de control con la acción de control anterior
```

Donde se aplica la acción anterior sobre el sistema en el momento actual.

Ahora vamos a ver el caso de un controlador PD predictivo, para ellos empleamos la planta presentada en la ecuación (10).

$$G_p(s) = \frac{1,667e^7}{(s^2 + 8333s + 1,667e^7)} \quad (10)$$

Empleando un $f = 10\text{kHz}$ obtengo la siguiente FT en tiempo discreto.

$$G_{pd}(z) = \frac{0,01816z + 0,0158}{z^2 - 1,625z + 0,6592} \quad (11)$$

De alguna manera se obtiene el controlador PD predictivo que se describe a continuación.

$$G_{PDp}(z) = \frac{U(z)}{E(z)} = \frac{3,755z - 3,425}{z^2} \quad (12)$$

La respuesta correspondiente se presenta en la siguiente figura.

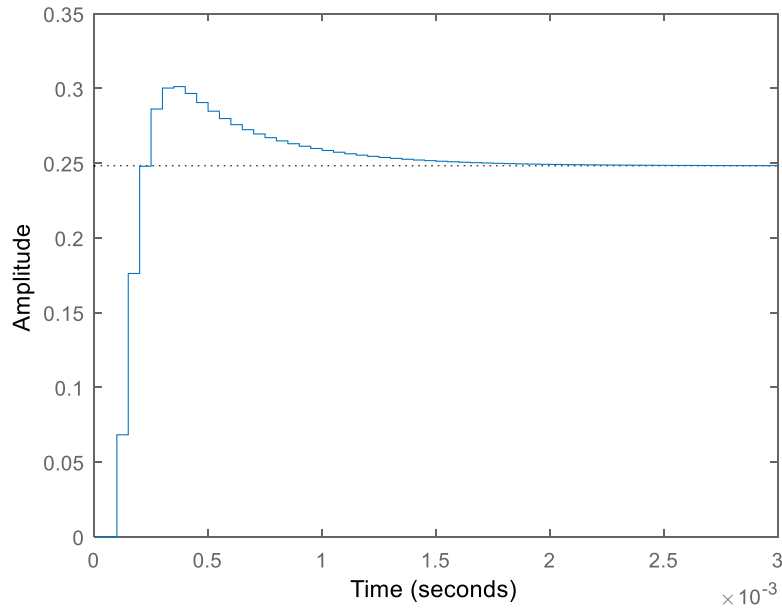


Figura 5: Respuesta del sistema controlado por un PD pred.

En cuanto a la acción de control es:

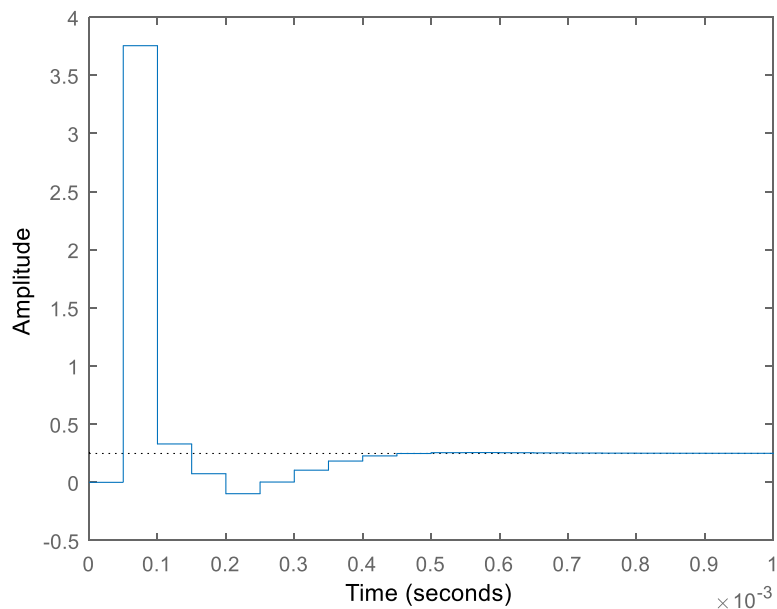


Figura 6: Acción de control del controlador PD pred sobre la planta.

Para la implementación tomamos la ecuación 12 y obtenemos la ecuación recursiva

$$u(k) = K_1 e(k-1) + K_2 e(k-2) \quad (13)$$

Donde $K_1 = 3,755$ y $K_2 = 3,425$.

Modificando el código de la DLL para implementarlo en el PSIM se obtiene.

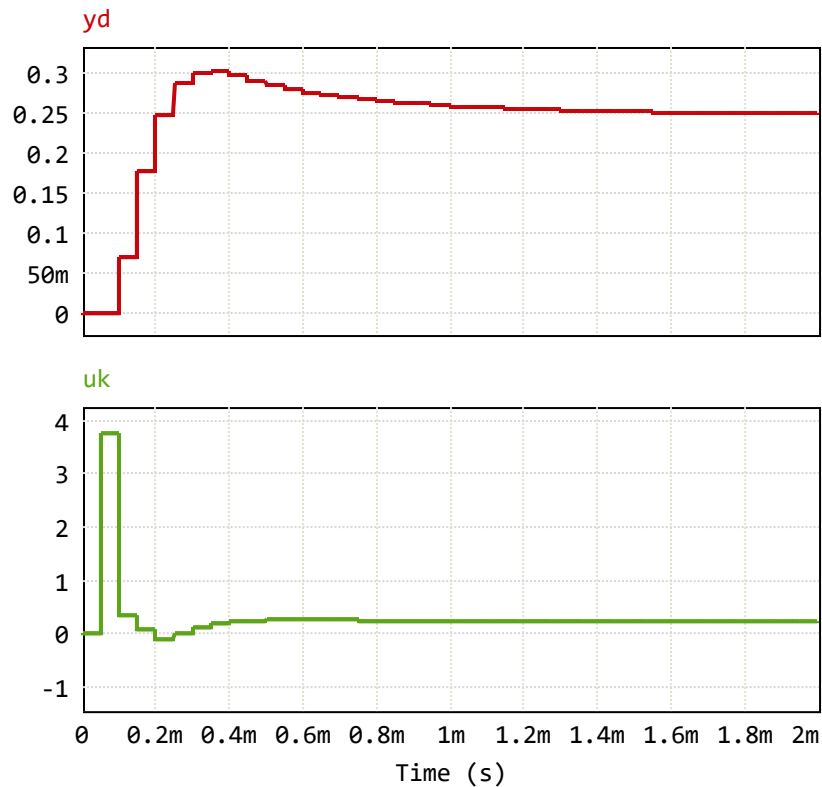


Figura 7: Respuesta del controlador PD pred empleando PSIM y la dll.

Cabe resaltar que para este tipo de controladores, donde no se posee un integrador, se puede optar por emplear una acción *feedforward*, que consiste en sumar a la acción de control la referencia actual.

El diagrama de bloques que refleja esta posibilidad está dado por:

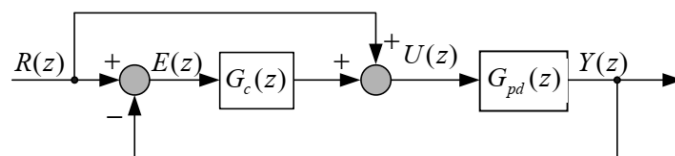


Figura 8: Diagrama de bloques de un controlador *feedforward*.

Observando el esquema podemos obtener la siguiente expresión:

$$G_{LCcc}(z) = \frac{Y(z)}{R(z)} = \frac{G_{pd}(z)[1+G_c(z)]}{1+G_c(z)G_{pd}(z)} \quad (14)$$

Mientras que la acción de control será:

$$\frac{U(z)}{R(z)} = \frac{1+G_c(z)}{1+G_c(z)G_{pd}(z)} \quad (15)$$

Por lo tanto la ecuación recursiva del controlador PD predictivo más la acción *feedforward* será

$$u(k) = K_1 e(k-1) + K_2 e(k-2) + r(k) \quad (16)$$

Y consecuentemente la respuesta temporal será:

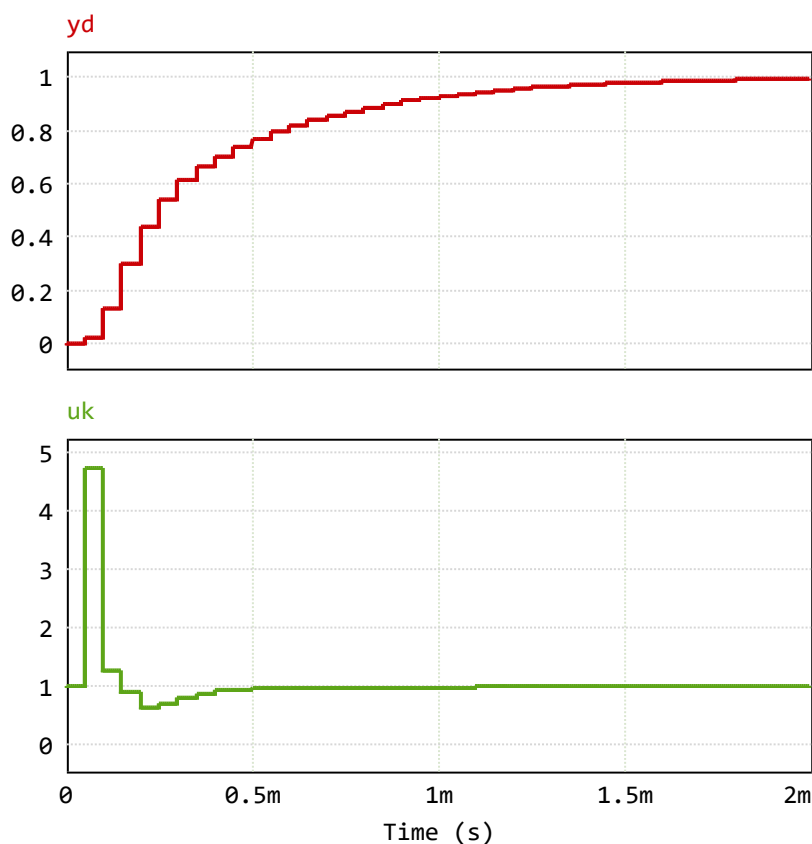


Figura 9: Respuesta del controlador PD pred + *feedforward* empleando PSIM y la dll.

El código empleado en la DLL para el cálculo de la acción de control quedaría como

```
out[2] = uk + ref; //Se actualiza uk con el valor anterior
ek = ref - y;
uk=K1*ek + K2*ekm1; //pisa el valor de uk
// Almacena variables anteriores para próximo periodo.
ukm1 = uk; ekm2 = ekm1; ekm1 = ek;
```

Conclusiones

En el escrito se expone una posibilidad de implementar controladores PID predictivos, incluyendo una acción *feedforward*. Se ponen de ejemplo dos casos de estudio, donde se resuelven empleando el Matlab y el PSIM con su DLL correspondiente. Los resultados son iguales en ambas resoluciones.

Referencias bibliográficas

- [1] J. H. Aylor, R. L. Ramey and G. Cook, "Design and application of a microprocessor PID predictor controller", *IEEE Trans. on Industrial Electronics and Control Instrumentation*, vol. 27, n. 3, pp. 133 – 137, Aug. 1980.