El PID es un método básico de control, en el que se busca mejorar la respuesta de la máquina de tres formas diferentes, por medio del método proporcional, el integral y el derivativo. Para regular el sobre impulso, el número y tiempo de las oscilaciones, y el error en estado estable, es una técnica muy sencilla que permite una gran mejora, de forma tradicional es necesario conocer la ecuación que caracteriza a la planta, pero aún sin conocerla se puede implementar el control.

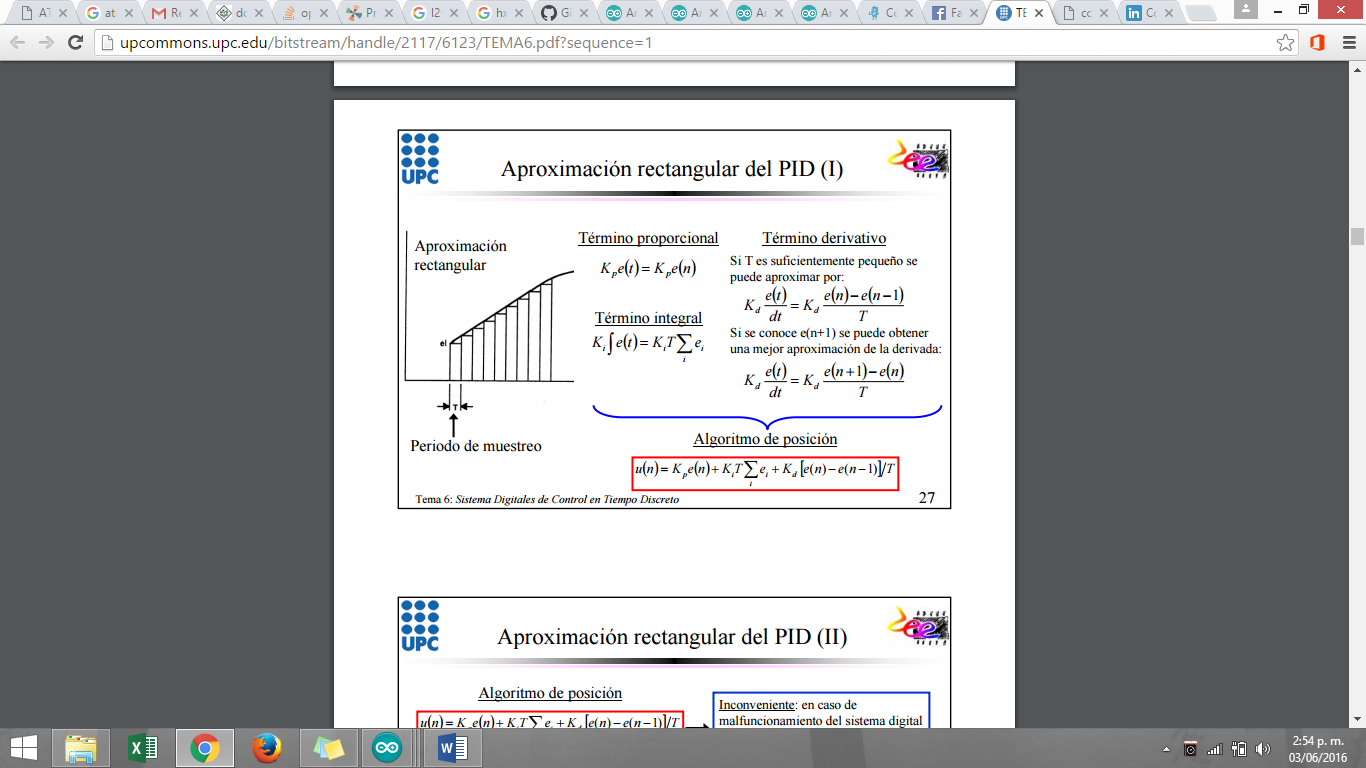
Al responder unas preguntas sencillas se comprende de mejor forma el funcionamiento de este.

¿El control se aplica en sistemas analógicos y digitales? Si, originalmente se aplicó a sistemas analógicos, pero con el crecimiento de la electrónica digital se implementó en está, ¿cómo es posible? Debido al funcionamiento original, porque las operaciones que utiliza se pueden cambiar del tiempo continuo al discreto, conservando el método proporcional como una multiplicación, remplazando el integral por una sumatorio, y el derivativo por su definición como pendiente en un punto. Estos cambios se aplican en cada instante de tiempo obteniendo un resultado aproximado al analógico.

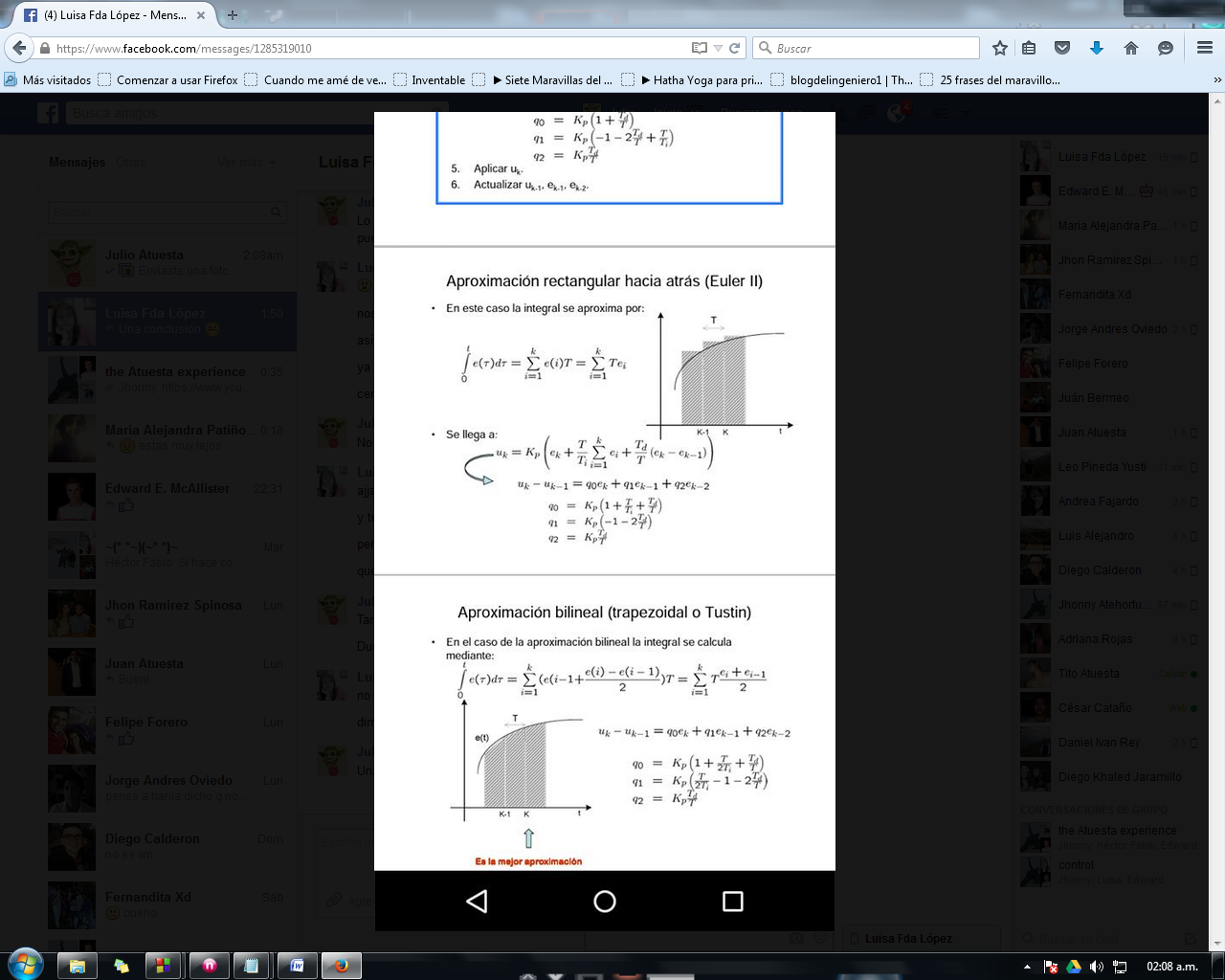
¿Funciona mejor el control PID de forma digital o analógica?, depende de la aplicación, pero en un sistema con una alta frecuencia de muestreo en comparación a la frecuencia con la que se desarrolla el evento un PID digital puede tener mejores resultados que el analógico debido a que los sistemas digitales tienen menos problemas de pérdidas y acoples de impedancias al ser desarrollada la técnica en software y no de forma mecánica o electrónica en el hardware.

¿Exactamente cómo funciona? El control se cimienta sobre cinco valores claves, el primero es la referencia deseada en la salida del sistema, el comportamiento que deseamos que cumpla. El segundo es el error que se tiene, la diferencia entre el estado deseado y el estado real. Las otras tres corresponden a la constante proporcional Kp, está se multiplica con el error (el cual tiene signo) para afectar directamente a la salida sumando o restando el valor obtenido, la constante integral Ki que se multiplica con una sumatoria de los errores pasados, ya que este trabaja después del sobre impulso de la señal, sobre la oscilación del valor deseado siempre tendrá en su sumatoria componentes positivas negativas del error, afectando a la salida en forma que la magnitud y por tanto el número de las oscilaciones disminuya, finalmente la constante derivativa Kd multiplicada con la diferencia de estados anteriores busca predecir el estado futuro para disminuir el error en el estado estable.

De forma matemática el control PID discreto se representa así:



Como se mencionó la parte proporcional del control se mantiene, mientras que la integral se aproxima a una sumatoria, es este caso se usa la aproximación trapezoidal, una forma sencilla explicada en la siguiente imagen.



La parte derivativa del control se expresa como siendo que el sistema se mueve en una sola dimensión se tendrá donde *x* representa la parte proporcional del control.

Escrito en el código se obtine lo siguiente:

----declaración de variables de interes.

uint8\_t kp, ki, kd; //constantes PID, podrían ser de 16 bit para más precisión

uint8\_t proporcional, prop\_anterior, integrativo, derivativo;

uint8\_t sensor, error, error\_anterior, referencia;

--------- se activa el conversor análogo digital, para obtener mediante ensayo y error el valor de las constantes de control que proporcionen el mejor resultado

ADCSRA=(1<<ADEN)|(1<<ADSC);// se habilita el ADC e inicia la conversión

kp=(ADCL); //toma 8 menos significativos retornados en al conversión pin ADC0

ADMUX=(1<<MUX0); // lee pin 1 PIN ADC1

ADCSRA|=(1<<ADSC);

ki=(ADCL); //físicamente como se conecta el aref?

ADMUX&=0x04;// PIN ADC4

ADCSRA|=(1<<ADSC);

kd=(ADCL);

---------- se escribe en líneas de código la solución lógica a los problemas del sistema, y se implementa el PID en la variable que se desea afectar

while (sensor==0x06); //curva derecha

{

proporcional=kp\*error;

integrativo=ki\*(proporcional +((error-error\_anterior)/2));

derivativo=kd\*(proporcional-prop\_anterior);

prop\_anterior=proporcional; // se guarda el estado actual como pasado para que sea usado en un estado futuro

error\_anterior=error;

OCR0B=255;//motor izquiedo

OCR0A=proporcional+integrativo+derivativo; // motor derecho, con control aplicado para manejar el cambio de la salida que se crea en la curva

}