

# Capítulo

# 12

## PWM

# Pulse Width Modulation

Comumente conhecida por sua sigla em inglês PWM (Pulse-Width Modulation), a modulação por largura de pulso de um sinal, ou em fontes de alimentação, envolve a modulação de sua razão cíclica (duty cycle) para transportar qualquer informação sobre um canal de comunicação ou controlar o valor da potência entregue a uma carga.

### 12.1 Introdução

A maneira mais simples de se controlar a potência aplicada a uma carga é através da associação de um reostato em série, conforme mostra a figura 1.

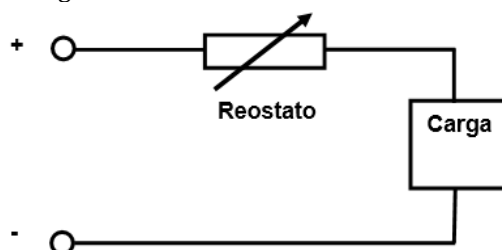


Figura 1 – Controle de potência em uma carga com o uso de reostato.

Variando-se a resistência do reostato, pode-se modificar a corrente na carga e, portanto, a potência aplicada a ela. Este tipo de controle ainda é encontrado nas lâmpadas de painéis de alguns veículos antigos.

A grande desvantagem deste tipo de controle, denominado “linear”, é que a queda de tensão no reostato multiplicada pela corrente que ele controla representa uma grande quantidade de energia dissipada em forma de calor. O controle passa a dissipar (e pedir) mais potência que a aplicada na própria carga em determinadas posições do ajuste. Além dessa perda ser inadmissível, ela exige que o componente usado no controle seja capaz de dissipar elevadas potências, o tornando caro e grande (normalmente são usados reostatos ou potenciômetros de fio, mesmo para potências relativamente baixas).

Uma outra opção de controle de potência seria a substituição do reostato por transistores, ou circuitos integrados, em um controle mais elaborado, polarizados adequadamente de forma que ainda varie linearmente a potência aplicada pelo controle direto da corrente. Mesmo assim, a potência dissipada pelo dispositivo que controla a corrente principal (o transistor) é elevada. Esta potência depende da corrente e da queda de tensão no dispositivo e, da mesma forma, em certas posições do ajuste, pode ser maior que a própria potência aplicada à carga.

Na eletrônica moderna, é fundamental que se busque um rendimento com pequenas perdas e a ausência de grandes dissipadores que ocupam muito espaço, principalmente quando circuitos de alta potência estão sendo controlados. Desta forma, este tipo de controle de potência linear não é conveniente, sendo necessárias outras configurações de maior rendimento, como as que fazem uso da tecnologia PWM.

Os controles de potência em inversores de frequência, conversores para servomotor, fontes chaveadas e muitos outros circuitos utilizam o conceito PWM (Pulse Width Modulation), ou Modulação por Largura de Pulso, como base de seu funcionamento.

Para que se entenda como funciona esta tecnologia no controle de potência, partimos do circuito mostrado na figura 2, formado por um interruptor eletrônico de ação muito rápida e uma carga.

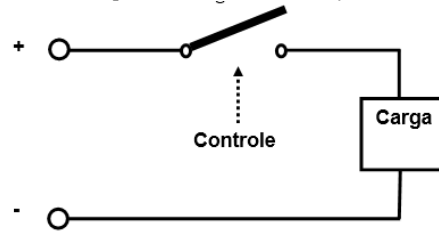


Figura 2 – Controle de potência em uma carga com o uso de interruptor eletrônico.

Quando o interruptor está aberto não há corrente na carga e a potência aplicada é nula. No instante em que o interruptor é fechado, a carga recebe a tensão total da fonte e a potência aplicada é máxima.

Como fazer para obter uma potência intermediária, digamos 50%, aplicada à carga? Uma ideia é fazer com que a chave seja aberta e fechada rapidamente de modo a ficar 50% do tempo aberta e 50% do tempo fechada. Isso significa que, *em média*, teremos metade do tempo com corrente e metade do tempo sem corrente, conforme mostra a figura 3.

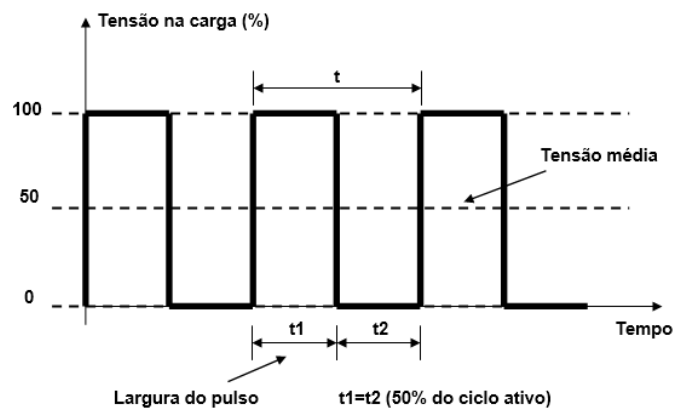


Figura 3– Princípio de funcionamento do PWM.

A tensão média aplicada à carga, é, neste caso, 50% da tensão de entrada. Veja que o interruptor fechado define uma largura de pulso pelo tempo em que ele fica nesta condição ( $t_1$ ), e um intervalo entre pulsos pelo tempo em que ele fica aberto ( $t_2$ ). Os dois tempos juntos definem um período ( $t$ ) e, portanto, uma frequência de controle. A relação entre o tempo de duração do pulso ( $t_1$ ) e a duração de um ciclo completo de operação do interruptor ( $t$ ) nos define ainda o ciclo ativo (*Duty Cycle*), que neste caso é de 50%.

Variando-se a largura do pulso e mantendo o período  $t$  constante, teremos ciclos ativos diferentes e, com isso, podemos controlar a tensão média e a potência média aplicada à carga. Assim, quando a largura do pulso varia de zero até o máximo, a potência também varia. Este princípio é usado justamente no controle PWM: modulamos a largura do pulso de modo a controlar o ciclo ativo do sinal aplicado a uma carga e, com isso, a potência aplicada a ela.

A figura 4 apresenta sinais PWM para diferentes larguras de pulso e diferentes valores médios de tensão ( $V_{average}$ ) aplicada à carga.

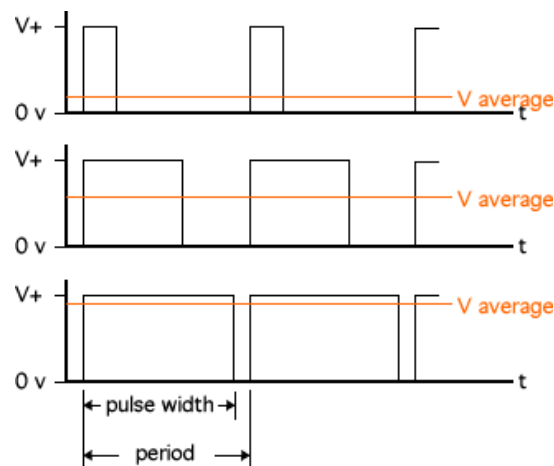


Figura 4 – Sinal PWM com diferentes larguras de pulso.

Na prática, o interruptor é substituído por algum dispositivo de estado sólido que possa abrir e fechar o circuito rapidamente como, por exemplo, um transistor bipolar, um FET de potência, um IGBT ou um TRIAC.

Na operação de um controle por PWM, existem diversas vantagens a serem consideradas e alguns pontos para os quais o projetista deve ficar atento para não desperdiçar estas vantagens. Na condição de aberto, nenhuma corrente circula pelo dispositivo de controle e, portanto, sua dissipação é nula. Na condição de fechado, teoricamente, se ele apresenta uma resistência nula, a queda de tensão é nula, e ele também não dissipa nenhuma potência. Isso significa que, na teoria, os controles PWM não dissipam potência alguma e, portanto, consistem em soluções ideais para este tipo de aplicação.

Na prática, entretanto, isso não ocorre. Em primeiro lugar, os dispositivos usados no controle não são capazes de abrir e fechar o circuito num tempo infinitamente pequeno. Eles precisam de um tempo para mudar de estado e, neste intervalo de tempo, sua resistência sobe de um valor muito pequeno até um valor muito alto, ou o contrário. Neste intervalo de tempo a queda de tensão e a corrente através do dispositivo não são nulas, e uma boa quantidade de calor poderá ser gerada, de acordo com a carga controlada.

Dependendo da frequência de controle e da resposta do dispositivo usado, uma boa quantidade de energia poderá ser perdida neste processo de comutação. Entretanto, mesmo com este problema, a potência dissipada num controle PWM ainda é muito menor do que num circuito de controle linear com potência equivalente (a eficiência de controle de potência com PWM pode chegar a 98%). Transistores de comutação rápida, FETs de potência, e outros componentes de chaveamento podem ser suficientemente rápidos para permitir que projetos de controles de potências elevadas sejam implementados sem a necessidade de grandes dissipadores de calor conectados nesses dispositivos.

O segundo problema que poderá surgir vem justamente do fato de que os transistores de efeito de campo, ou os bipolares, usados em comutação não se comportam como resistências nulas quando saturados (fechados). Os transistores bipolares podem apresentar uma queda de tensão de alguns volts quando saturados, o mesmo ocorrendo com os FETs. Deve-se observar, em especial, o caso dos FETs de potência que são, às vezes, considerados comutadores perfeitos, com resistências de fração de ohm entre o dreno e a fonte quando saturados ( $R_{ds(on)}$ ) mas, na prática, não é isso que ocorre.

A baixíssima resistência de um FET de potência quando saturado (resistência entre dreno e fonte no estado *on*) só é válida para uma excitação de *gate* feita com uma tensão relativamente alta. Assim, dependendo da aplicação, principalmente nos circuitos de baixa tensão, os transistores de potência bipolares ou mesmo os IGBTs podem ser ainda melhores que os FETs de potência.

O controle de potência por PWM possui diversas aplicações práticas, tais como: controle de velocidade de motores, controle de luminosidade de lâmpadas e LEDs, controle do posicionamento de servo motores, controle de temperatura em processos térmicos, em sistemas de telecomunicações analógicos, onde a informação modula a largura do pulso, conversão D/A, entre outros.

## 12.2 PWM no Arduino Mega 2560

A placa Arduino Mega 2560 possui pinos específicos que podem ser usados como saídas PWM. Os pinos digitais de 2-13 e de 44-46 podem fornecer um sinal PWM a partir do uso da seguinte função interna:

```
analogWrite(pino, duty),
```

onde *pino* é um dos pinos disponíveis e *duty* corresponde ao *duty cycle*. O parâmetro *duty* é um número que varia de 0 a 255, onde 0 corresponde a um *duty cycle* de 0% e 255 corresponde a 100%. Após a chamada dessa função, o pino passa a fornecer uma onda quadrada de frequência fixa e com *duty cycle* conforme o valor passado pela função. Não é necessário declarar o pino como OUTPUT, pois internamente a função já se encarrega de configurá-lo, embora seja uma forma de deixar claro ao programador que o pino é uma saída. A frequência do sinal PWM, na maioria dos pinos, é aproximadamente 490 Hz, porém, os pinos 4 e 13 operam em aproximadamente 975 Hz.

O programa do quadro abaixo mostra um exemplo de aplicação do PWM. O programa fornece o sinal PWM ao LED conectado ao pino 13, onde o *duty cycle* é controlado pela tensão analógica presente no pino A0.

```
void setup() {  
  pinMode(13, OUTPUT);  
}  
  
void loop() {  
  unsigned int val = analogRead(A0);  
  analogWrite(13, val/4);  
}
```

O programa simplesmente configura o pino 13 como saída digital na função `setup( )`. A função `loop( )` declara a variável `val`, do tipo inteiro sem sinal, que recebe o valor lido da porta analógica A0. A variável é então passada como o *duty cycle* da função `analogWrite( )`. O valor é dividido por 4 porque a leitura analógica retorna um número que varia de 0 a 1023, e o parâmetro `duty` deve variar apenas de 0 a 255.

## 12.3 PWM – Acesso aos registradores

A função `analogWrite( )` não está relacionada com as entradas analógicas e nem com a função `analogRead( )`. Ela apenas fornece um modo simples para se trabalhar com sinais PWM, porém não possibilita nenhum controle sobre a frequência do sinal PWM fornecido no pino.

Em alguns casos, a frequência do sinal PWM é muito importante para o sistema, como por exemplo a frequência de acionamento de uma bobina de um motor. Para modificar a frequência, deve-se acessar os registradores de controle da geração dos sinais PWM. Os timers são os responsáveis pela geração dos sinais PWM e a sua relação com os pinos controlados se dá como segue:

TIMER0 → pinos 4, 13  
TIMER1 → pinos 11, 12  
TIMER2 → pinos 9, 10  
TIMER3 → pinos 2, 3, 5  
TIMER4 → pinos 6, 7, 8  
TIMER5 → pinos 44, 45, 46

Como pode ser verificado, um dado timer controla o sinal PWM gerado em mais de um pino. Qualquer configuração feita para um timer específico, afetará todos os pinos relacionados a ele. Deve-se enfatizar que alterações na configuração do TIMER0 também afetam as funções de tempo do Arduino, como `delay( )` e `millis( )`.

Por exemplo, o bloco de código mostrado abaixo configura o TIMER1 para geração de sinais PWM nos pinos 11 e 12 com uma frequência de 2 kHz, aumentando ainda a resolução do *duty cycle*, que é limitada em 8 bits na função `analogWrite( )`, para 10 bits. A configuração é feita escrevendo-se nos registradores TCCR1A e TCCR1B. Isso faz com que o timer 1 seja configurado para contar continuamente de 0 a 1023, levando o pino de saída para o nível lógico alto quando o timer estiver em 0, e resetando o pino quando o conteúdo do timer se igualar ao registrador OCR1.

```
TCCR1A = TCCR1A | 0b10100011;  
TCCR1A = TCCR1A & 0b10101111;  
TCCR1B = TCCR1B | 0b00001010;  
TCCR1B = TCCR1B & 0b11101010;
```

Para ajustar o *duty cycle* com a resolução de 10 bits, deve-se escrever diretamente o valor *duty* no registrador de comparação de saída OCR1, como em:

```
OCR1A = duty;  
OCR1B = duty;
```

## 12.3 Implementação de controle PID em sistemas microcontrolados

Processos contínuos têm sido controlados por sistemas de controle em malha fechada (com realimentação) desde o final de 1700 (com o uso de um dispositivo mecânico para controlar a velocidade da máquina a vapor de James Watt, em 1788). Embora o controle em malha fechada tenha percorrido um longo caminho desde James Watt, a abordagem básica e os elementos do sistema não mudaram. Para fins de discussão, vamos usar um sistema de controle de temperatura como modelo nas descrições que seguem.

Os elementos que compõem um sistema de controle em malha fechada são:

*Planta* – As partes que produzem aquecimento ou resfriamento no sistema;

*Sensores* - Os dispositivos de medição que medem as variáveis dentro da planta;

*Setpoint* (ponto de ajuste) - O valor no qual se deseja que o sistema opere. É a temperatura desejada;

*Sinal de erro* - Diferença entre temperatura atual da planta e a temperatura desejada;

*Distúrbios* - Uma perturbação, por exemplo, seria uma porta aberta que permite uma rajada de ar frio sobre o ambiente controlado, modificando rapidamente a temperatura;

**Controlador** - Intencionalmente deixado por último, este é o elemento mais importante de um sistema de controle. O controlador é responsável por várias tarefas e é o elo que liga todos os elementos físicos e não físicos do processo. Ele mede o sinal de saída de sensores da planta, processa o sinal e, em seguida, calcula um erro com base na medição e no ponto de ajuste. Uma vez que os dados dos sensores foram medidos e tratados, o resultado deve ser usado para encontrar um valor de excitação (uma tensão, por exemplo) que, em seguida, deve ser enviado para a planta para que o erro possa ser corrigido. A taxa em que tudo isto acontece depende do poder de processamento do controlador, que pode ou não ser um problema, dependendo da característica de resposta da planta. Um sistema de controle de temperatura exige menos poder de processamento do que um sistema de controle de velocidade de um motor, por exemplo. Isso está diretamente relacionado com o tempo de resposta da planta quando o controlador envia um sinal de excitação (sistemas térmicos são mais lentos que sistemas mecânicos).

A figura 5 mostra um diagrama de blocos básico de um sistema de controle em malha fechada, obtida pela realimentação do sinal de saída do sistema, mediante a leitura por um sensor. Na figura, o Setpoint está representado pela entrada *Reference*.

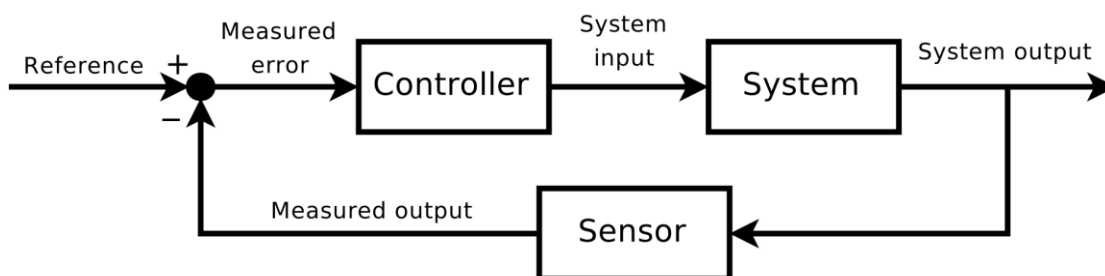


Figura 5 – Sistema de controle em malha fechada.

Existem muitos métodos para se controlar uma variável em um processo automatizado. Como exemplo de métodos de controle, alguns deles muito difundidos e pesquisados em todo mundo acadêmico, podemos citar o controle *On-Off*, passando pelo PID, lógica difusa (controle *Fuzzy*) até as recentes redes neurais.

### Controle On-Off

Em se tratando de controle de temperatura por resfriamento, uma grande parcela dos sistemas encontrados no mercado utiliza ainda o método *On-Off*, que consiste em se determinar uma temperatura de trabalho onde o sistema de refrigeração irá operar em sua máxima potência, quando detectada temperatura acima do *setpoint*, permanecendo assim até que a temperatura se torne menor que o valor do *setpoint*. Neste método de controle, o sistema atua através de um termostato, conforme a figura 6.

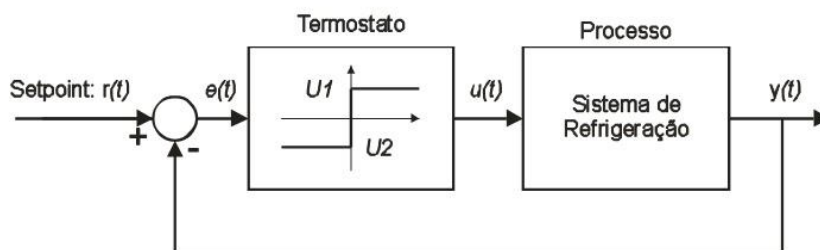


Figura 6 – Sistema de controle On-Off.

O sinal de controle  $u(t)$  só pode assumir dois valores, ligado ou desligado, conforme o valor do erro apresentado seja positivo ou negativo. Matematicamente, podemos expressar  $u(t)$  como:

$$u(t) = \begin{cases} U1, e(t) > 0 \\ U2, e(t) < 0 \end{cases} \quad (1)$$

Podemos notar que não existe valor de  $u(t)$  quando  $e(t) = 0$ . Isso causa uma grande variação na temperatura do ambiente controlado, que em muitos casos pode aceitar esta variação.

Um grande problema deste sistema ocorre quando a temperatura  $y(t)$  está muito próxima ao valor do *setpoint*  $r(t)$ . Neste caso,  $u(t)$  poderá entrar em oscilação devido à proximidade destes valores. Para resolver este problema, podemos utilizar o método de controle *On-Off* com histerese, conforme figura 7.

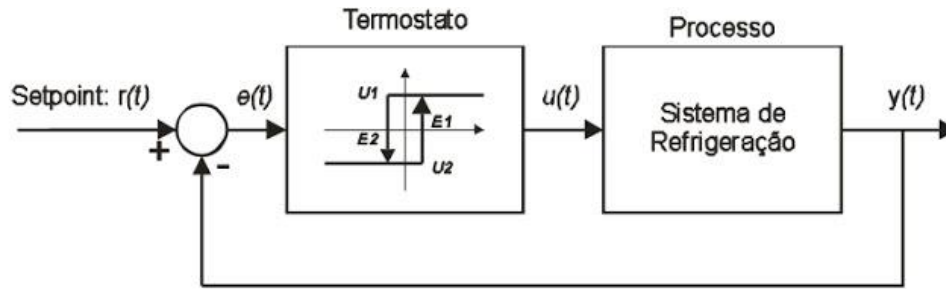


Figura 7 – Sistema de controle On-Off com histerese.

Desta forma, cria-se uma banda morta entre o *setpoint* que acionará o sistema de refrigeração e o *setpoint* que desligará este mesmo sistema. Isso acaba com o problema de oscilações no acionamento do sistema, mas, por outro lado, aumenta ainda mais a variação da variável controlada  $y(t)$ . Portanto, quando precisamos que o controle seja mais eficaz, ao ponto de manter essa variação de temperatura do ambiente em níveis muito pequenos, faz-se necessário a aplicação de métodos de controle mais elaborados e eficientes.

### Controle Proporcional (P)

Depois do controle *On-Off*, em termos de simplicidade, pode-se utilizar o controle proporcional, que já traz uma melhoria na resposta da saída controlada, que trata de usar o erro do sistema  $e(t)$  e aplicar um ganho fixo (constante de proporcionalidade  $K_p$ ) antes da realimentação, representado pelo diagrama de blocos da figura 8.

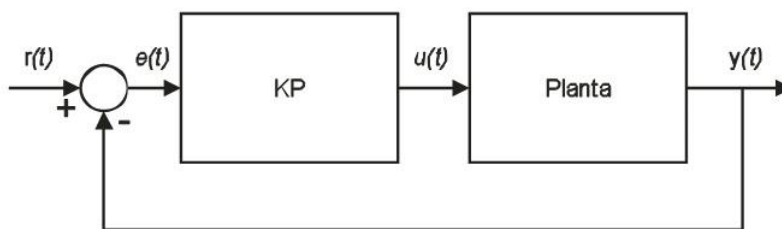


Figura 8 – Sistema de controle Proporcional.

Assim, o sinal aplicado à planta será sempre inversamente proporcional ao módulo do erro  $e(t)$ . Quanto maior o ganho  $K_p$ , menor será o erro em regime permanente, ou seja, melhor a precisão do sistema em malha fechada. O erro  $e(t)$  será diminuído com o aumento de  $K_p$ , porém nunca poderá ser anulado, pois se o erro for zero, não haverá sinal  $u(t)$  para excitar a planta. Em contrapartida, quanto maior o ganho  $K_p$ , mais oscilatório o sistema tende a ficar, podendo chegar à instabilidade.

### Controle Integral (I)

O Controle integral não é isoladamente uma técnica de controle. Deve ser associada a outras, como a proporcional, por exemplo. O controle integral consiste, assim como seu nome sugere, em fazer o sinal de controle  $u(t)$  proporcional a um ganho  $K_i$  multiplicado pela integral do sinal de erro  $e(t)$ , onde  $K_i$  é uma constante ajustável, como mostrado na figura 9. Para erro nulo, o valor de  $u(t)$  permanece estacionário.

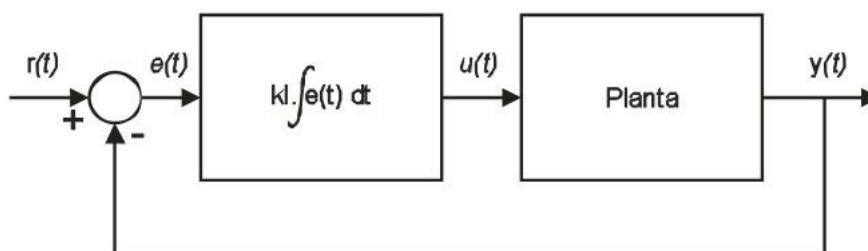


Figura 9 – Sistema de controle Integral.

Se o valor de  $e(t)$  for dobrado, por exemplo, então o valor de  $u(t)$  varia duas vezes mais rápido.

### Controle Derivativo (D)

Assim como o controle integral, o controle derivativo não é uma técnica de controle utilizada isoladamente. A ação de controle derivativa, quando adicionada a um controlador proporcional, propicia um meio de obter um controlador com alta sensibilidade. Uma vantagem em se usar ação de controle derivativa é que ela responde à taxa de variação do erro atuante e pode produzir uma correção significativa antes de o valor do erro atuante tornar-se demasiadamente grande. O controle derivativo, portanto, antecipa o erro atuante e inicia uma ação corretiva mais cedo, tendendo a aumentar a estabilidade do sistema.

O controle derivativo consiste, assim como seu nome sugere, em se aplicar ao sinal de controle  $u(t)$ , um ganho  $K_d$  proporcional à derivada do sinal de erro  $e(t)$ , onde  $K_d$  é uma constante ajustável, como mostrado na figura 10.

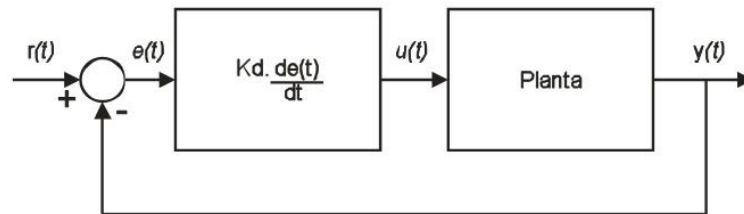


Figura 10 – Sistema de controle Derivativo.

### Controle Proporcional – Integral – Derivativo (PID)

A combinação das técnicas de controle proporcional, integral e derivativo é conhecida como controle PID, que soma a ação proporcional às ações integral e derivativa. A ação integral elimina o erro em um regime estacionário, pois apresenta um resultado de saída não nulo quando o erro do sistema for nulo. Por outro lado, a ação integral pode introduzir oscilação na resposta. A ação derivativa antecipa o erro e produz uma ação corretiva mais cedo e proporcional à taxa de variação do erro atual. A figura 11 mostra o diagrama de blocos da implementação de um controlador PID.

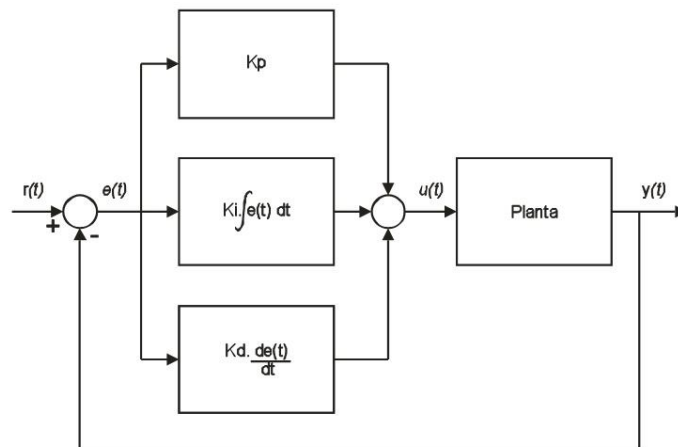


Figura 11 – Sistema de controle PID.

A equação que expressa a saída  $u(t)$  em função do erro para o controlador PID é dada por:

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \quad (2)$$

### Implementação digital do controlador PID

A equação do PID é bastante complexa e quando executada em tempo real durante o processo de controle exige muito processamento. Os microcontroladores comerciais não destinados ao processamento digital de sinais, geralmente não possuem capacidade para processar todos os cálculos desta equação em tempo real. Desta forma, para aplicarmos o controle PID em um microcontrolador da Família Arduino, por exemplo, temos que deduzir a equação para um formato digital e adotar algumas aproximações nos termos integral e derivativo.



Existem algumas técnicas matemáticas para realizar a discretização, ou aproximação digital, de uma equação no tempo, objetivando a equação das diferenças a ser implementada no microcontrolador. Assim, com as equações discretizadas, as seguintes operações devem ser realizadas para implementação do algoritmo:

1. Inicialmente, deve-se fazer a leitura atual do erro entre o setpoint e a variável que se deseja controlar (saída):

```
erro = setpoint-saida;
```

2. Calcula-se o termo proporcional da expressão:

```
proporcional = Kp*erro;
```

3. Calcula-se o termo integral da expressão. Isso é feito em dois passos: no primeiro, usa-se uma variável que serve como acumulador de erro (erro\_acc), fazendo a operação equivalente à integração, e, em seguida, multiplica-se o resultado pela constante  $k_i$ :

```
erro_acc = erro_acc+erro;  
integral = Ki*erro_acc;
```

4. Calcula-se o termo derivativo da expressão. Isso é feito em dois passos: no primeiro, usa-se uma variável que calcula a diferença entre o erro atual e o erro anteriormente medido (erro\_dif), fazendo a operação equivalente à derivação, e, em seguida multiplica-se o resultado pela constante  $k_d$ :

```
erro_dif = erro - erro_anterior;  
derivativo = Kd*erro_dif;
```

4. Calcula-se o resultado total da expressão PID, somando cada um dos termos:

```
PID_out = proporcional + integral + derivativo;
```

O valor PID\_out deve então ser apropriadamente convertido para representar a faixa de variação máxima permitida do sinal de saída que alimenta o atuador do processo, para que a variável de interesse possa ser devidamente controlada.

Na prática, os valores calculados do termo integral são testados contra valores limites, a fim de evitar um efeito chamado de *wind-up*. Esse efeito é caracterizado pelo aumento significativo no termo integral devido a um tempo de resposta da planta significativamente longo, fazendo com que o erro demore muito para diminuir. Dessa forma, o termo integral é normalmente limitado a esses valores máximos.

O mesmo ocorre com a saída do controlador PID. Normalmente, a variável PID\_out é limitada, pois representa o sinal de excitação de um atuador, uma fonte de tensão por exemplo, ou um sinal PWM, que tem um limite máximo que deve ser naturalmente respeitado.

Os valores das constantes  $K_p$ ,  $K_i$  e  $K_d$  devem ser ajustados para cada tipo de processo a ser controlado. Esses valores estão relacionados ao tempo de resposta da planta. O processo de determinação dos valores das constantes é chamado de sintonia do controlador PID (*Tuning*) e existem algumas formas de fazê-lo, sendo o método de Ziegler-Nichols a maneira mais difundida.

Os valores máximos obtidos em cada etapa de processamento podem variar significativamente de um problema para outro. Dessa forma, os ajustes necessários devem ser feitos para que se atenda às necessidades de um problema específico. ■