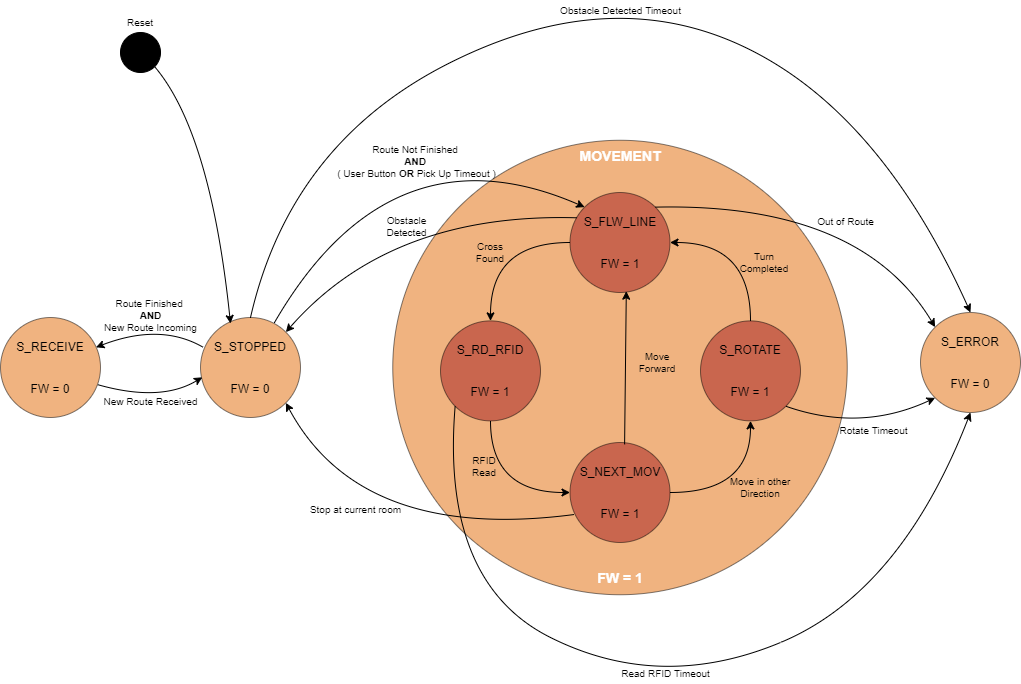
**Máquina de Estados**

O DWR tem de executar ações distintas durante toda a sua atividade. De forma a ser possível executar a ação correta em todos os momentos, é necessário um sistema que faça a gestão do estado atual do robô e das entradas que possam despoletar a alteração deste mesmo estado. Na fig x, está representado o diagrama que representa a máquina de estados implementada para este robô. Esta é composta por quatro estados principais: S\_STOPPED, S\_RECEIVE, MOVEMENT e S\_ERROR. O estado de MOVEMENT pode ser subdividido em quatro estados secundários: S\_FLW\_LINE, S\_RD\_RFID, S\_NEXT\_MOV e S\_ROTATE.



Legenda: Diagrama da máquina de estados.

Inicialmente, o DWR encontra-se no estado S\_STOPPED e transita para o estado S\_RECEIVE quando a rota atual estiver terminada e o operador iniciar a transmissão de uma nova rota. Após receber a rota selecionada pelo operador na aplicação, o robô transita do estado S\_RECEIVE para o estado S\_STOPPED, onde espera que lhe seja dada a permissão para iniciar o seu movimento, através da pressão do botão presente na sua lateral. Assim que este botão seja pressionado, o sistema evolui para o estado de S\_FLW\_LINE. Neste estado o robô segue a linha até que aconteça algum dos seguintes imprevistos: presença de um obstáculo no percurso do robô, deteção de uma cruz de paragem (quarto ou cruzamento) ou ocorrência de falha no controlador do seguidor de linha. Caso o percurso esteja obstruído por um obstáculo, o DWR para, evitando a colisão com o mesmo, voltando para estado S\_STOPPED. Se for detetada uma linha horizontal por ambos os sensores das extremidades do *array* de sensores (cruz de paragem), significa que o robô está na presença de um quarto ou cruzamento, sendo necessário efetuar a leitura do cartão RFID a este associado, no estado S\_RD\_RFID. Caso o cartão RFID seja lido com sucesso, o sistema evolui para o estado S\_NEXT\_MOV, onde determina qual o próximo estado do DWR, de acordo com a rota selecionada. No estado S\_NEXT\_MOV, o robô transita para o estado S\_FLW\_LINE se o RFID detetado for relativo a um quarto que não esteja marcado como local de paragem na rota ou a um cruzamento em que não seja necessário efetuar a mudança de direção. Pelo contrário, caso seja necessário efetuar paragem no quarto atual, dá-se uma transição para o estado S\_STOPPED. No caso do RFID detetado ser relativo a um cruzamento e for necessário mudar de direção, o sistema transita para o estado S\_ROTATE. Uma vez neste estado, o DWR roda na direção indicada pela rota, até estar orientado na direção pretendida, voltando ao estado S\_FLW\_LINE.

Todas estas ações têm *timeouts* associados, que são ativos quando a ação demora mais tempo a ser realizada do que o esperado, permitindo ter controlo sobre o robô em casos imprevisíveis, transitando para o estado S\_ERROR, por exemplo, quando o DWR se encontrar parado à espera de que o percurso seja desobstruído por um longo período de tempo, quando ocorre um erro (cartão errado ou demasiado tempo) na leitura de um cartão RFID ou quando ocorre um erro na mudança de direção do robô. Além disso, o robô transita para o estado S\_ERROR quando o robô sai totalmente da linha durante o seu percurso. Uma vez neste estado, o DWR precisa da intervenção de um responsável para que possa voltar ao seu estado de funcionamento normal.

**O que é um Controlador**

Um controlador é responsável pelo controlo de processos através de algoritmos específicos. O seu principal objetivo consiste na monitorização, identificação e interpretação de processos, via modelos matemáticos, de forma a produzir uma ação de controlo conveniente. Existem três ações de controlo distintas: proporcional, integral e derivativa, que podem ser conjugadas entre si. A primeira tem uma ação imediata, proporcional ao valor atual do erro, acelera a resposta de um processo controlado, reduz o tempo de subida e o erro máximo. No entanto, aumenta o *overshoot*, o tempo de estabilização e produz um *offset* inversamente proporcional ao ganho. A ação integral produz uma ação de controlo gradual proporcional à integral do erro, respondendo, assim, ao passado do erro enquanto este for diferente de zero, elimina o *offset* e reduz o tempo de subida. Porém, aumenta o *overshoot*, o período de oscilação e tempo de estabilização, produzindo respostas lentas e oscilatórias. A ação derivativa produz uma ação antecipatória e proporcional à derivada do erro. É usada para acelerar e estabilizar a malha, reduzir o *overshoot*, o erro máximo e o período de oscilação. Contudo, não é indicada para processos com ruído.

Na Figura, estão presentes as três ações de controlo descritas. A figura (a) mostra a resposta de um sistema a uma ação proporcional. A saída deste corresponde à variável de erro multiplicada por uma dada constante. A figura (b) mostra a resposta de um sistema a uma ação integral, onde a saída deste corresponde à integral da variável de erro. Como a integral de uma constante é uma reta, quando a entrada (variável de erro) é do tipo degrau, a resposta do sistema vai corresponder a uma rampa de declive igual à amplitude da variável de entrada multiplicada por uma constante. A figura (c) mostra a resposta de um sistema a uma ação derivativa, em que a saída deste corresponde à derivada da variável de erro. Como a derivada de uma reta é uma constante, quando a entrada é do tipo rampa, a saída do sistema corresponde a um degrau de amplitude igual à amplitude da rampa multiplicada por uma constante. Um controlador que conjugue as três ações é denominado por controlador proporcional integral derivativo (PID).

|  |  |  |
| --- | --- | --- |
|  |  |  |
| (a) | (b) | (c) |

**Análise do sistema do controlo**

O primeiro passo num projeto de controlo está relacionado com a compreensão qualitativa do sistema. Este passo é absolutamente fundamental. Se esta análise falhar, quaisquer que sejam as ferramentas matemáticas e o tempo posteriormente despendido, o projeto dificilmente irá funcionar corretamente. É necessário compreender como o sistema a controlar funciona fisicamente, quais as variáveis medidas, a controlar, de atuação, distúrbio e comando, bem como o funcionamento dos atuadores, sensores e controlador.

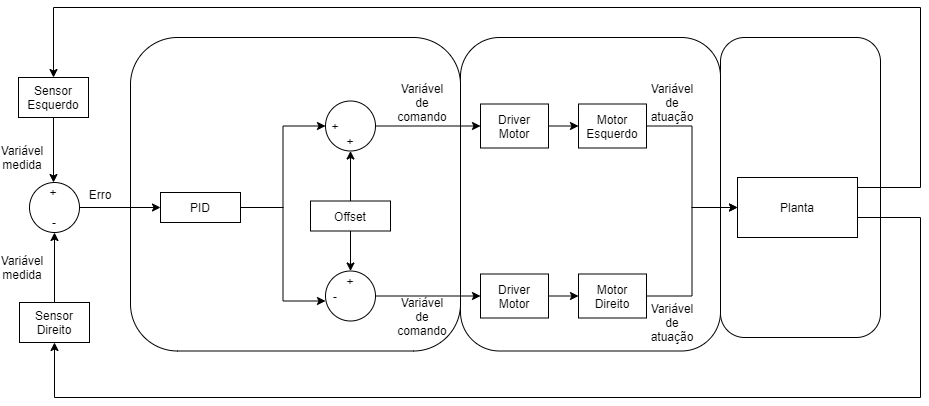
O sistema a ser controlado assemelha-se a um paralelepípedo com uma roda posicionada na parte central de cada lateral do robô. Sendo o objetivo do sistema seguir uma linha, que pode conter trajetórias retas e curvilíneas, a velocidade de rotação dos motores terá de variar de modo a ser possível ajustar as velocidades de translação e de rotação do centro de massa do robô. Se se pretender que o robô siga uma trajetória retilínea, os motores terão de rodar à mesma velocidade. Se se pretender que o robô efetue uma trajetória curvilínea, o motor do lado oposto ao que se pretende efetuar a trajetória terá de ter uma velocidade de rotação superior. Ou seja, quando a trajetória é uma curva à esquerda, o motor do lado direito terá de ter uma velocidade de rotação superior comparativamente com a do motor do lado direito. Quando a trajetória é uma curva à direita, passa-se exatamente o oposto.

As variáveis medidas são as leituras efetuadas pelos dois sensores responsáveis pelo módulo do seguidor de linha, o sensor 3 e o sensor 6. As variáveis a controlar são a velocidade de translação e velocidade de rotação do centro de massa do robô. As variáveis de atuação são os binários dos motores. As variáveis de comando são a fração de modulação do amplificador PWM de cada motor. A variável de perturbação será o atrito provocado pela superfície (binário de perturbação).

O atuador é composto pelo motor DC, a ponte H e pelo microcontrolador. O binário produzido pelo motor DC pode ser alterado através da variação da tensão de alimentação do motor (produzida por um amplificador PWM). O sistema de controlo é responsável pela variação deste parâmetro, de maneira a produzir o binário adequado. Em conjunto, o microcontrolador e a ponte H implementam o amplificador PWM. O algoritmo de controlo implementado no microcontrolador produzirá na saída a variável de comando, a fração de modulação do amplificador de PWM, sendo o ganho deste amplificador correspondente a tensão aplicada ao motor DC quando a fração de modulação é um.

O controlador irá ser executado pelo microcontrolador. A regra de controlo executada por este controlador está implementada numa rotina de serviço à interrupção (ISR), que será despoletada por um *timer* com período igual ao período de amostragem escolhido pelo sistema de controlo. Esta ISR terá como parâmetros de entrada os valores das leituras dos dois sensores a utilizar e como saída o cálculo da fração da fração de PWM, que servirá de entrada aos amplificadores PWM.

Na Figura, está ilustrado o sistema de controlo implementado. Sendo o objetivo principal do DWR o seguimento de uma linha, é necessário manter os sensores na parte exterior da mesma. Quando um dos sensores se aproximar da linha, o motor do lado oposto terá de compensar o desvio da trajetória. Daqui pode-se concluir que a variável de referência é zero, ou seja, pretende‑se que diferença entre as leituras dos dois sensores seja nula. Assim, a entrada do controlador (variável de erro) será a diferença entre a leitura dos mesmos. O controlador colocará saída uma variável que terá de ser manipulada de modo a efetuar o pretendido. Se o valor de erro for positivo, significa que o sensor esquerdo se encontra mais próximo da linha do que o sensor direito, implicando que o motor direito terá de ter uma velocidade de rotação superior. À variável de saída do bloco PID soma-se um valor de offset que servirá de variável de comando para o atuador direito (*driver* motor e motor). Ao simétrico da variável de saída do bloco PID soma‑se o mesmo valor de offset, que servirá de variável de comando para o atuador esquerdo (*driver* motor e motor). Deste modo, a variável de comando do atuador direito é superior à variável comando do atuador esquerdo, garantindo-se, assim, o pretendido. Se o valor do erro for negativo, passa-se exatamente o oposto. Se a variável de saída do controlador PID for nula não é necessário fazer ajustes de direção e o DWR seguirá o percurso com velocidade de rotação constante em ambos os motores com valor igual ao valor de *offset.*



Adicionar cores aos blocos deste diagrama, como já se fez para o AWR. Escrever “variável de comando para o motor direito” e “variável de comando para o motor esquerdo”, em vez de “Variável de comando” para os dois? O mesmo para a variável de atuação? Erro -> Variável de erro: Panta -> DWR???? Trocar mais com menos a saída do PID

**Controlador Implementado**

**Bloco PID**

Um controlador PID contínuo é definido pela Equação, em que , e são referidos, respetivamente, como os ganhos proporcional, integral e derivativo, e como a variável de erro ao longo do tempo. A variável de comando, , corresponde ao somatório das ações proporcional, integral e derivativa.

|  |  |  |
| --- | --- | --- |
|  |  | (2.4) |

Para uma implementação em microcontrolador é necessária uma versão digital do controlador. De entre as diversas famílias das regras PID implementou-se o algoritmo de posição. Esta versão consiste em substituir os termos integral e derivativo pelos seus equivalentes discretos. Assim, aproximando a integral por uma soma e a derivada pela diferença divida de 1ª ordem obtém-se a Equação, em que e são, respetivamente, a ação de controlo e o erro no instante n. Em cada instante é calculado o valor real (posição) do sinal de saída do controlador. No contexto do problema, a variável de erro corresponde à diferença entre as leituras dos dois sensores, aqui representados, simbolicamente, por e (Equação).

|  |  |  |
| --- | --- | --- |
|  |  | (2.4) |
|  |  | (2.4) |

Para efeitos de cálculo, o valor de pode ser escrito de forma mais adequada, calculando o erro apenas uma vez e colocando em evidência as constantes na aproximação à integral e à derivada (Equação). Pode-se agora redefinir os ganhos proporcional, integral e derivativo (Equação) obtendo-se, por substituição, a Equação.

|  |  |  |
| --- | --- | --- |
|  |  | (2.4) |
|  |  | (2.4) |
|  |  | (2.4) |
|  |  | (2.4) |
|  |  | (2.4) |

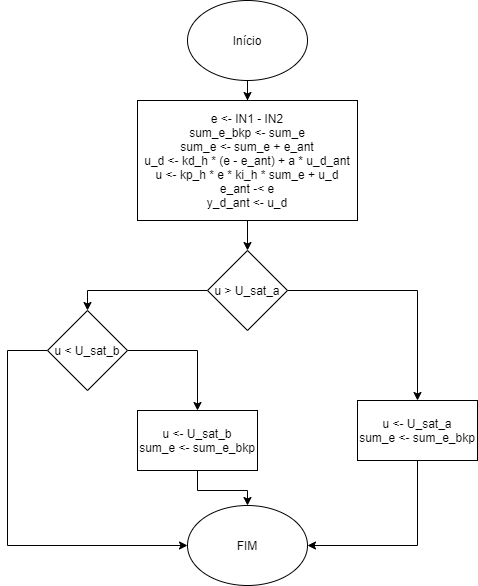
Para calcular o somatório dos erros presente na Equação, define-se a variável (Equação) que pode ser escrita de forma recursiva (Equação). Juntando todas as transformações efetuadas, obtém-se na Equação uma nova forma de calcular a expressão inicial (Equação).

|  |  |  |
| --- | --- | --- |
|  |  | (2.4) |
|  |  | (2.4) |
|  |  | (2.4) |

Uma forma de reduzir os problemas com erros de medida prende-se com a utilização de um filtro passa-baixo na ação derivativa. Na Equação apresenta-se a expressão desta ação. Calculando a sua transformada em z obtém-se a Equação. A aproximação à derivada introduz um zero em um e um polo em zero no plano z. Pode deslocar-se o polo para a direita com um filtro passa‑baixo resultando as Equações. A função de transferência tem um zero em um, como no caso anterior, mas o polo acentua-se em *a* e não em 0. A implementação desta função de transferência está representada na Equação. Pode-se então redefinir como apresentado na Equação. Adicionando a componente do filtro passa-baixo Equação chega-se à Equação. A segunda reduz-se à primeira se *a* for zero, ou seja, a ação derivativa sem filtro de passa-baixo é obtida com *a* igual a zero, o que implica que .

|  |  |  |
| --- | --- | --- |
|  |  | (2.4) |
|  |  | (2.4) |
|  |  | (2.4) |
|  |  | (2.4) |
|  |  | (2.4) |
|  |  | (2.4) |
|  |  | (2.4) |
|  |  | (2.4) |

Um dos problemas de algoritmo de posição está relacionado com a possível saturação do valor de *u*. Definem-se, então, limites de saturação superior () e inferior (). Se o resultado do cálculo do valor de *u* for superior ao valor do limite de saturação superior ou inferior ao valor do limite de comparação inferior é necessário limitar o mesmo aos valores de saturação impostos. Caso contrário, o valor passado para a rotina de saída será truncado e apenas a parte não truncada será entendida como valor a estabelecer. Isto leva a saltos indesejáveis e perigosos da variável *u* que, em absoluto, não deveriam acontecer. Modifica-se a interrupção para detetar a ultrapassagem dos valores de saturação e, caso tal aconteça, fixa-se o valor o valor de *u* nos valores de saturação. Além disso, a saturação do atuador pode dar origem a um aumento brusco dos valores dos somatórios dos erros levando a oscilações prejudiciais no valor da variável controlada. Tende em conta estes aspetos e fazendo uso da Equação é possível desenhar o fluxograma que permite a implementar o algoritmo de controlo no microcontrolador, Figura. Como se pode verificar, a atualização do somatório dos erros só é efetuada quando a variável *u* se encontra dentro dos limites de saturação. Caso contrário, a atualização é anulada, o que equivale, em tempo contínuo, a parar a integração do erro e o valor de *u* é fixado no valor de saturação respetivo.



**Bloco *Offset***

A variável de saída do bloco PID tem de ser manipulada de modo que o DWR seja capaz de fazer ajustes de trajetórias, garantindo sempre a não saturação da variável de comando. Sabe‑se que a variável de saída do bloco PID se encontra sempre dentro dos limites de saturação. Assim sendo, um modo de assegurar estes dois objetivos, é fazer uma média ponderada com os valores de saída do bloco PID e *offset.* A Equação permite calcular a variável de comando do atuador direito e a Equação permite calcular a variável de comando do atuador esquerdo.

|  |  |  |
| --- | --- | --- |
|  |  | (2.4) |
|  |  | (2.4) |

~~Na Figura, mostra-se a máquina de estados desenvolvida, sendo composta por quatro estados principais: S\_STOPPED, S\_RECEIVE, MOVEMENT e S\_ERROR. No primeiro, o DWR está parado à espera de algum estímulo. O segundo dedica-se à comunicação de novas rotas entre a unidade de controlo e o robô. O terceiro está relacionado com todo o tipo de operações que o DWR terá de efetuar durante o tempo em que se encontra em movimento. O quarto é um estado crítico e o DWR apenas deve transitar para este caso ocorra algum erro que comprometa o normal funcionamento do sistema.~~

~~O estado de MOVEMENT pode ser subdividido em quatro estados secundários: S\_FLW\_LINE, S\_RD\_RFID, S\_NEXT\_MOV e S\_ROTATE. O primeiro implementa um algoritmo responsável pelo seguimento de linha. O segundo é responsável pela leitura de um cartão RFID de identificação unívoca para cada cruzamento. O terceiro é um estado de decisão responsável por encaminhar o sistema para um estado que esteja de acordo com as entradas dos sensores e com o percurso a realizar. O quarto executa o controlo da mudança de direção num cruzamento.~~