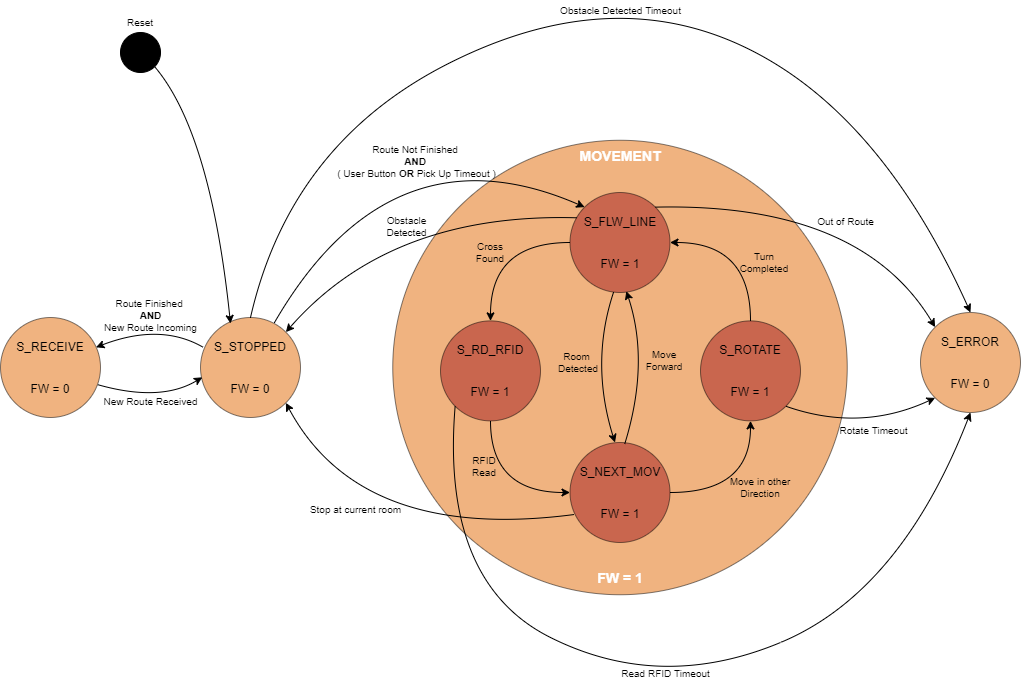
**Máquina de Estados**

O DWR tem de executar ações distintas durante toda a sua atividade. De forma a ser possível executar a ação correta a todo o momento, 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 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 o operador 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 do 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.

Inicialmente, o DWR encontra-se no estado S\_STOPPED. Apenas transita deste estado quando o operador iniciar a transmissão de uma nova rota. Para garantir que o robô executa cada rota na sua totalidade confirma-se, também, se a última rota já foi concluída. Verificadas estas duas condições, o sistema evolui para o estado S\_RECEIVE, e permanece neste estado até que a comunicação com o operador seja concluída, regressando, novamente, ao estado de S\_STOPPED.

Uma vez no estado S\_STOPPED e já com uma rota guardada na memória (por concluir ou por iniciar), o DWR espera que lhe seja dada ordem de início de marcha através do botão de pressão presente na sua lateral. Assim que este botão seja pressionado, o sistema evolui para o estado de S\_FLW\_LINE. Este estado tem três saídas possíveis. Caso o percurso esteja obstruído por obstáculo, o DWR para evitando a colisão com o mesmo, ou seja, volta ao estado de S\_STOPPED. Se for detetada uma linha horizontal por ambos os sensores das extremidades do QTR8A, significa que se está na presença de um cruzamento e é necessário efetuar a leitura de um cartão RFID, fazendo com que o sistema transite para o estado S\_RD\_RFID. Caso seja detetada uma linha horizontal apenas por um dos sensores das extremidades do QTR8A, significa que se está na presença de um quarto e tem de se verificar se este quarto é um local de paragem, ou seja, o sistema evolui para o estado de S\_NEXT\_MOV. A outra transição possível para este estado, acontece quando o sistema se encontra no estado de S\_RD\_RFID e o cartão é lido com sucesso.

Uma vez no estado de S\_NEXT\_MOV, o sistema tem de determinar qual o próximo passo a efetuar. Caso o DWR encontre num cruzamento e não seja para mudar de direção, ou o quarto que detetou não seja um local de paragem, o sistema volta ao estado de S\_FLW\_LINE, continuando o percurso atual. Caso seja necessário efetuar paragem no quarto atual, há uma transição para o estado S\_STOPPED. Se o robô se encontrar num cruzamento e for necessário mudar de direção, o sistema evolui para o estado S\_ROTATE. Uma vez neste estado, o DWR roda até efetuar a mudança de direção, voltando ao estado S\_FLW\_LINE.

O sistema pode entrar no estado S\_ERROR por diversas razões. Sempre que o DWR se encontrar parado à espera de que a via seja desobstruída, quando sai totalmente da linha durante o percurso ou quando acontece algum problema na mudança de direção ou quando ocorre um erro na leitura de um cartão RFID. Uma vez neste estado, o DWR precisa da intervenção de um responsável para que possa voltar ao seu estado de funcionamento normal.

**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 dos de rotação motores tem 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 rodar a uma velocidade superior comparativamente com a do motor do lado direito. Quando a trajetória é uma curva à direita, passa-se exatamente o oposto.

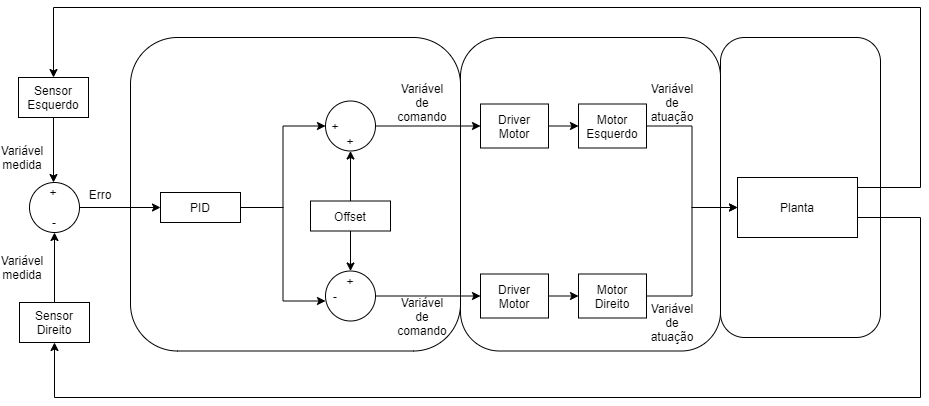
As variáveis medidas são as leituras efetuadas pelos dois sensores responsáveis pelo módulo do seguidor de linha. As variáveis a controlar são a velocidade de translação e velocidade de rotação do centro de massa do robô. As variáveis de atuação são os binários dos motores. As variáveis de comando são 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 microcomputador. 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 microcomputador 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 1.

Falar dos Sensores???

O controlador irá ser executado pelo microcomputador. 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 da 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 se pode concluir que a variável de referência é zero, ou seja, nenhuma diferença entre as leituras dos dois sensores. Assim, a entrada do controlador (variável de erro) será a diferença entre a leitura dos mesmos.



**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 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* e o tempo de estabilização e produz um *off-set* inversamente proporcional ao ganho. A segunda produz uma ação de controlo gradual proporcional à integral do erro, responde ao passado do erro enquanto este for diferente de zero, elimina o *off-set* e reduz o tempo de subida. Porém, aumenta o *overshoot*, o período de oscilação e tempo de estabilização, produz respostas lentas e oscilatórias e tende a instabilizar a malha. A terceira produz uma ação antecipatória e proporcional à derivada do erro. É usada para acelerar e estabilizar a malha. Reduz o *overshoot*, o erro máximo e o período de oscilação. No entanto, não é indicada para processos com ruído.

Na Figura, estão presentes as três ações de controlo descritas. A figura (a) mostra a ação proporcional. A saída do sistema corresponde à variável de erro multiplicada por uma dada constante. A figura (b) mostra a ação integral. A saída do sistema corresponde à integral da varável de erro. Como a integral de uma constante é uma reta, quando a entrada (variável de erro) é 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 ação derivativa. A saída do sistema corresponde à derivada da variável de erro. Como a derivada de uma reta é uma constante, quando a entrada é uma 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) |

**Controlador 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 microcomputador é necessária uma versão digital do controlador.