

# MEMOS

Grupo 1 : Pedro Sousa (up201704307) e Leonardo Hügens (up201705764)

Projeto 6 : Controlador de Semáforos

## MEMO 1 (14/04)

- Objetivo geral do projeto: implementação em LabView de vários semáforos numa estrutura encontrada em casos reais, inicialmente pensamos na interseção entre a Rua Júlio Dinis e a Praça de Mouzinho de Albuquerque.
- As variáveis que estamos planeamos considerar são: número de carros numa determinada área, a sua velocidade média, input de peões através dos botões para oportunidade de atravessar, altura do dia (fluxo de carros e peões está altamente correlacionado com a altura do dia).
- A estruturas lógicas que já prevemos implementar de raiz são as relações de oposição entre semáforos de veículos e semáforos de peões e as que existem entre todos os semáforo de veículos, com vista na otimização de facilidade de trânsito na estrutura.
- Realizamos uma simulação de um semáforo funcional, interligado com um semáforo para peões.
- O semáforo para peões apresenta um "timer" que apresenta o tempo restante em que o semáforo fica verde.

## MEMO 2 (21/04)

- Discutir a maneira de aumentar a eficiência dos semáforos, isto é, dar a informação ao semáforo do transito de forma a ele ficar mais ou menos tempo verde/vermelho de forma a otimizarmos o fluxo de trânsito.
- Organizarmos o nosso programa original de forma a conseguirmos associar um semáforo isolado a um subVi. Assim, quando estivermos a estudar um sistema mais complexo, vamos ter uma maneira mais organizada de trabalhar o nosso diagrama de blocos o que irá facilitar no debugging e na leitura do mesmo.
- Estudar a aplicação do acionamento antecipado por controlo de veículos e adicionar ao nosso sistema.

## MEMO 3 (28/04)

- Numa tentativa de generalizar o semáforo para o poder clonar, implementámos um Event Structure (uma estrutura do LabView apenas acionada quando alguma variável muda de valor). Depois de implementada (alfa\_0.2), revelou-se muito difícil fazer uma subVI com essa estrutura, pelo que desistimos dela.
- Durante esta semana, estudamos um cruzamento simples, implementado-o em estruturas mais fáceis de conter dentro de uma subVI, como Case Structure e Flat Sequence.
- Temos então dois semáforos no cruzamento, e representamos a quantidade de carros imediatamente atrás de cada semáforo com Vertical Pointer Sliders, e também temos indicadores

de excesso de velocidade e de congestionamento em cada via. Já conseguimos implementar uma medida automática de gestão de congestionamento nas duas vias.

- Para a próxima semana, para além de acabar de configurar um dos Sliders de congestionamento, temos o nosso objetivo é aumentar a complexidade da estrutura, adicionando mais vias, cuja configuração ainda está por decidir.

#### MEMO 4 (05/05)

- A estrutura que decidimos implementar por agora é o cruzamento das Ruas Latino Coelho e Alegria, no Porto. É um cruzamento de duas vias, sendo que uma delas tem duas vias e outra uma.
- Para uma melhor perceção do funcionamento dos semáforos, implementamos uma representação visual do cruzamento em si.
- Fizemos uma atualização ao funcionamento dos semáforos, em que implementamos um rácio de entrada e um controlador de velocidade para cada via, e todos independentes entre si (a priori, claro que o funcionamento dos semáforos em si vai implicar uma dependência da quantidade de carros em cada via).

#### MEMO 5 (12/05)

- Com o objetivo de tornar a simulação do nossos semáforo independente de input de utilizador em tempo real, implementámos uma função que gera as velocidades dos carros da via com uma determinada distribuição de probabilidade. Esta função foi escrita num script de python, e fizemos a ligação entre este e o nosso .vi principal através do Python Node. A distribuição de probabilidade que estamos a usar por agora é a distribuição de cauchy, que visualmente corresponde a uma versão mais picada de uma distribuição normal, sendo que o valor médio desta distribuição varia inversamente com o número de carros nessa via, através de uma função sigmoide. A ideia da utilização desta função veio do seu uso frequente em redes neuronais como função de ativação de neurónios.
- Também escrevemos nesse script de python uma função que descreve a evolução ao longo do dia dos rácios de entrada. Para essa evolução escolhemos uma combinação linear de duas distribuições de cauchy, centradas uma às oito da manhã e outra às cinco da tarde, e bastante alargadas.
- Inicialmente estávamos a criar uma sessão para cada valor que precisávamos, mas depressa percebemos que não é uma boa prática em termos de gestão de recursos computacionais, pelo que alteramos o script, escrevendo uma função 'get' que recebe um array de todos os inputs e dá output de um array com todos os valores que precisamos, e assim para cada ciclo do código só é necessário criar duas vezes uma sessão de python.

#### MEMO 6 (19/05)

- Como já consideramos o nosso projeto quase na sua fase final decidimos apenas implementar um método fácil de saber a "hora do dia" a que se encontra o programa de forma a termos uma leitura mais direta dos valores para o congestionamento e para a velocidade média.
- Planeamos para o resto da semana começar a escrever o relatório e fazer um ou outro ajuste ao programa que achemos necessário.

## MEMO 7 (26/05)

- Iniciamos a escrever o relatório, usando um esquema semelhante ao divulgado pelos professores no moodle.
- As secções abordadas esta semana foram: introdução, planeamento (onde apresentamos um breve resumo das várias versões a apresentar) e desenvolvimento (onde fazemos um estudo pormenorizado das várias versões do programa).