

PROJETO FINAL

Introdução

Abaixo seguem três sugestões de projetos para o projeto final. Escolha um deles para implementar. Os projetos são:

- Simulador de tráfego usando MQTT.
- Controle de temperatura usando MQTT.
- Medidor de capacitância e indutância usando um osciloscópio.

Os dois primeiros podem ser feitos em dupla. O terceiro deve ser feito de forma individual. A entrega consistirá do código submetido ao Classroom e uma apresentação presencial ao professor.

Descrição dos projetos

Os três projetos serão detalhados abaixo. Para os dois primeiros, o simulador de tráfego e o controlador de temperatura, será realizada uma simulação de tempo discreto, ou seja, as entidades envolvidas avançam no tempo de forma sincronizada em certos instantes de tempo separados uniformemente. Isso significa que o cenário fica congelado entre estes instantes de tempo. Por exemplo, para o primeiro projeto, um carro só terá a sua posição atualizada baseada na sua velocidade nestes instantes de tempo. No intervalo entre os instantes o carro está parado em sua última posição.

Simulador de tráfego

Para este projeto, considere uma cidade cujo formato é retangular, com ruas retas que vão de um lado a outro. Ruas horizontais unem os lados esquerdo e direito da cidade, e as ruas verticais unem os lados superior e inferior. Observe que quaisquer duas ruas são ou paralelas ou perpendiculares. As ruas tem a mesma largura e possuem duas mãos de tráfego, onde cada mão tem apenas uma faixa. Isto significa que não há ultrapassagens.

Nesta cidade circulam apenas carros autônomos sem motoristas. Os carros são controlados por uma central de controle, e são usados para transportar pessoas pela cidade. As pessoas informam à central de controle que precisam de transporte, indicando sua posição. A central então verifica se há um carro livre e o direciona para a localização da pessoa. Quando o carro chega a esta localização, a pessoa informa o destino, que é comunicado à central de controle para direcionamento do carro.

Não há semáforos na cidade. Colisões são evitadas pela central de controle através do aumento ou redução das velocidades dos carros.

Este projeto consiste da simulação do cenário acima com um certo número de carros e de pessoas. Como dito acima, a simulação será em tempo discreto. Tome cuidado com a interação simultânea entre a atualização dos carros e da central de controle pelo simulador e a comunicação entre os carros

e a central de controle. Use um objeto do tipo Lock (veja as aulas de programação paralela) em cada carro e na central de controle para proteger o acesso dos dados internos de cada objeto.

Abaixo seguem os requisitos para este sistema:

- As pessoas informam da necessidade de transporte enviando uma mensagem para a central de controle com sua posição.
- A central de controle, ao receber um pedido de uma pessoa, escolhe um carro vazio e o direciona até a posição da pessoa.
- Quando o carro chegar onde a pessoa se encontra, ela deve informar a localização de seu destino à central de controle, que então irá calcular a rota que o carro deve seguir. Esta rota é transmitida pela central de controle ao carro.
- Ao chegar ao destino, o carro deve informar à central de controle. Enquanto a central não enviar um novo endereço para o carro, ele deve ser considerado como estacionado e não obstruirá a faixa de tráfego, nem se moverá.
- Deverá existir uma velocidade máxima que um carro pode se deslocar nas ruas.
- Os carros trafegam a uma velocidade constante até a central de controle enviar um comando de mudança de velocidade, quando então a velocidade é mudada instantaneamente para o novo valor.
- A central de controle poderá mudar a velocidade do carro para evitar colisões.
- Os carros possuem um sensor que os permitem saber sua posição a qualquer momento.
- Os carros devem enviar periodicamente sua velocidade e posição para a central de controle. Em particular, quando o carro chegar ao destino ou entrar em uma rua, ele deve informar à central de controle sua velocidade e posição.
- Todas as faixas das ruas tem a mesma largura, e portanto, como as ruas tem sempre duas faixas, todas as ruas tem a mesma largura.
- As ruas horizontais têm o mesmo comprimento, que é igual à largura da cidade. De forma similar, as ruas verticais têm o mesmo comprimento, que é igual ao comprimento da cidade.
- Os carros devem ficar a uma distância segura um do outro equivalente a uma largura de faixa.
- Toda comunicação dos carros e pessoas com a central de controle deve ser via o protocolo MQTT.
- A comunicação entre o simulador e os carros e a central de controle pode ser via chamada de função.
- A central de controle deve armazenar o histórico de velocidade e posição de todos os veículos durante toda a simulação.
- O sistema deve prover um servidor web para acesso a gráficos construídos a partir dos dados históricos.
- Cada rua horizontal é definida pelas coordenadas de seus cantos inferiores esquerdo e direito. As coordenadas dos cantos superiores são calculadas a partir das inferiores e da largura da rua.

- Cada rua vertical é definida pelas coordenadas de seus cantos esquerdo superior e inferior. As coordenadas dos cantos direitos são calculadas a partir das dos cantos esquerdo e da largura da rua.
- O sistema deve ser configurável no número de pessoas e carros e em suas posições iniciais.
- O sistema deve ser configurável no número de ruas, e em suas posições e largura de faixa.

Controlador de temperatura em uma edificação

Para este projeto, considere uma edificação com vários ambientes climatizados interconectados. Cada ambiente possui um sensor de temperatura e um ar condicionado controlados por uma central de controle.

A temperatura de um ambiente só influencia a temperatura dos ambientes vizinhos (embora os vizinhos, ao serem influenciados pelo 1º ambiente, influenciarão os seus outros vizinhos, e assim por diante). Como falado em sala de aula, o conjunto de temperaturas dos ambientes forma um sistema linear N -dimensional (onde N é o número de ambientes). Assuma que as equações que regem esse sistema são

$$\frac{dT_i}{dt} = \sum_{j=1}^N \alpha_{ij} (T_j - T_i) + b_i U_i = \sum_{\substack{j=1 \\ j \neq i}}^N \alpha_{ij} T_j - T_i \sum_{j=1}^N \alpha_{ij} + b_i U_i,$$

onde as constantes α_{ji} determinam a que ambientes j o ambiente i está conectado, e as constantes b_i determinam a influência do ar condicionado U_i na temperatura. Em um sistema real, os valores de α_{ij} e b_i dependem das geometrias dos ambientes, de suas interconexões e das características dos ar-condicionados. Para esta disciplina, escolheremos valores numéricos apropriados.

A partir destas equações, é possível montar o sistema linear definido por

$$\dot{\mathbf{T}} = \mathbf{A}\mathbf{T} + \mathbf{B}\mathbf{U},$$

onde $\mathbf{T} = (T_1 \cdots T_N)^T$ é o vetor-coluna das temperaturas, $\mathbf{U} = (U_1 \cdots U_N)^T$ é o vetor-coluna dos ar-condicionados, \mathbf{A} é a matriz $N \times N$ em função dos coeficientes α_{ij} , e \mathbf{B} é a matriz-diagonal $N \times N$ com os coeficientes b_i em sua diagonal. Observe que a matriz \mathbf{B} é diagonal porque assume-se que o ar-condicionado U_i influencia diretamente apenas o ambiente i .

O objetivo da Central de Controle é acionar os ar-condicionados de forma apropriada para fazer as temperaturas dos ambientes convergir para um valor $\bar{\mathbf{T}}$ configurado. A central deve aplicar um simples algoritmo de controle proporcional ao erro $\bar{\mathbf{T}} - \mathbf{T}$, i.e,

$$\mathbf{U} = P (\bar{\mathbf{T}} - \mathbf{T}),$$

onde P é uma matriz $N \times N$.

As matrizes \mathbf{A} , \mathbf{B} e P serão dadas posteriormente. Você deve escrever o seu código de tal forma que essas matrizes possam ser modificadas facilmente.

Este projeto consiste da simulação do cenário acima com um certo número de ambientes a ser informado depois. Como dito anteriormente, a simulação será em tempo discreto. Tome cuidado com a interação simultânea entre a atualização dos ambientes e da central de controle pelo simulador e a comunicação entre os ambientes e a central de controle. Use um objeto do tipo Lock (veja as aulas de programação paralela) em cada ambiente e na central de controle para proteger o acesso dos dados internos de cada objeto.

Abaixo seguem os requisitos para este sistema:

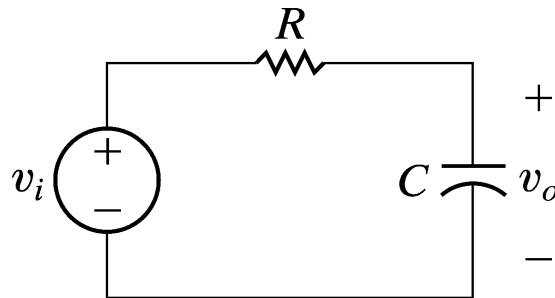
- Cada ambiente informa periodicamente à central de controle a sua temperatura T_i e o nível de potência U_i de seu ar condicionado. O período dessa comunicação será informada depois.

- Baseada nas temperaturas informadas, a central de controle calcula os novos valores \mathbf{U} da potência dos ar-condicionado de acordo com o algoritmo acima usando a matriz P .
- Toda comunicação entre os ambientes e a central de controle deve ser via o protocolo MQTT.
- A comunicação entre o simulador e os ambientes e a central de controle pode ser via chamada de função.
- A central de controle deve armazenar o histórico de temperatura \mathbf{T} e potência \mathbf{U} dos ar-condicionados de todos os veículos durante toda a simulação.
- O sistema deve prover um servidor web para acesso a gráficos construídos a partir dos dados históricos.
- O sistema deve ser configurável no número de ambientes N e nas matrizes A , B e P .

Medidor de capacitância e indutância

Este projeto consiste de um circuito para medir o valor da capacitância ou indutância de um capacitor ou indutor. Para a geração do sinal de entrada e as medições da resposta, deve-se usar um dos osciloscópios disponível no LDE.

O capacitor ou indutor a ser medido deve ser colocado como parte de um circuito elétrico linear de primeira ordem, como mostrado abaixo.



Uma onda quadrada com ciclo de trabalho de 50% deve ser colocada na entrada v_i do circuito e a tensão v_o em cima do capacitor ou do indutor é medida. O período da onda quadrada deve ser escolhido para que seja em torno de dez constantes de tempo τ do circuito de primeira ordem. Da Teoria de Circuitos Lineares (vista nas disciplinas de Circuitos Elétricos 1 e 2), sabe-se que, para o circuito acima, esta constante de tempo é $\tau = RC$. Relação semelhante pode ser obtida para um circuito com indutor. Pergunte ao professor em caso de dúvida.

Observe que, embora o valor da resistência R seja conhecido, o da capacitância não o é, uma vez que este é o valor a ser medido, e portanto o valor da constante de tempo também não o será. Parte do trabalho deste projeto, portanto, é estimar o melhor período da onda quadrada.

A escolha acima do período da onda quadrada (e, por conseguinte, da sua frequência) permite que o capacitor ou indutor se carregue quase inteiramente quando a onda quadrada tem o valor não-nulo (ou seja, na parte ativa da onda), e que se descarregue quase completamente quando a onda quadrada tem o valor 0. Da Teoria de Circuitos, sabe-se que a expressão para $v_o(t)$ quando o capacitor está se carregando é

$$v_o(t) = V_1 \left(1 - e^{-t/\tau} \right),$$

onde V_1 é a tensão da onda quadrada na parte ativa e τ é a constante de tempo do circuito. Expressão semelhante se aplica nos demais casos.

Observe que, como

$$\lim_{t \rightarrow \infty} v_o(t) = V_1,$$

podemos estimar V_1 sem precisar medir a entrada $v_i(t)$.

A partir da resposta v_o medida pelo osciloscópio, aplica-se um procedimento de ajuste de curvas por minimização de erro baseado na expressão teórica para $v_o(t)$ dada acima.

Você pode usar os módulos de Python `pyusb` e `python-usbtmc` para se comunicar com o osciloscópio.