

Modelagem e otimização de problemas utilizando ferramentas gráficas

Modelagem e simulação baseada em blocos

Em muitas aplicações, a modelagem e a simulação são ferramentas utilizadas para encontrar valores ótimos para os parâmetros de um sistema ou para obter a estratégia de controle mais eficiente para uma determinada planta. Este tipo de problema pode ser resolvido através da definição de uma função custo associada a um modelo e a um script de otimização do **solidThinking Compose**, por exemplo. No entanto, para quem prefere uma solução mais visual e integrada aos diagramas de bloco de um modelo existente, o **solidThinking Activate** possui ferramentas de otimização nativas, que contam com toda a simplicidade de *clique, arraste e solte* do programa, sem a necessidade de nenhuma *toolbox* adicional.

Objetivos

O objetivo principal deste problema é introduzir as ferramentas e as diferentes formas de otimizar problemas utilizando os blocos de otimização do Activate. O objetivo secundário é demonstrar uma aplicação prática destas ferramentas, neste caso, a melhor performance do câmbio de um veículo para maximizar distância percorrida e diminuir o tempo gasto.

Entendendo o problema

Contando com um veículo com câmbio de 5 marchas, o projeto busca alcançar a maior distância possível dentro de um intervalo de 30 segundos, partindo de uma velocidade inicial pré-estabelecida. Para alcançar o melhor resultado, busca-se encontrar a relação de transmissão ótima para cada uma das marchas, tendo em vista que a curva de torque do veículo é função da rotação do motor.



Figura 1: Problema de otimização automotiva

Modelando o problema

O problema pode ser modelado usando as equações dinâmicas de um veículo, como seguem abaixo:

$$\dot{\omega} = g_i^2 T(\omega) \frac{q}{c} - a\omega - bv^2$$

$$v = \frac{c}{g_i} \omega$$

$$\dot{x} = v$$

sendo:

ω : velocidade angular do motor

g_i : relação de transmissão para a engrenagem i

v : velocidade do veículo

x : distância percorrida

$T(\omega)$: torque como função da velocidade angular

a e b : constantes do coeficiente de atrito

q e c : constantes que dependem da característica do veículo

Quando ocorre uma mudança de marcha, ω dá um salto para que v seja mantida constante.

Modelando o problema no Activate

Uma vez que esteja definido o problema e as equações sejam conhecidas, é possível montar o problema no Activate, como mostrado na Figura 2. A partir da saída da caixa do seletor (g_i), é possível chegar às equações definidas para o problema.

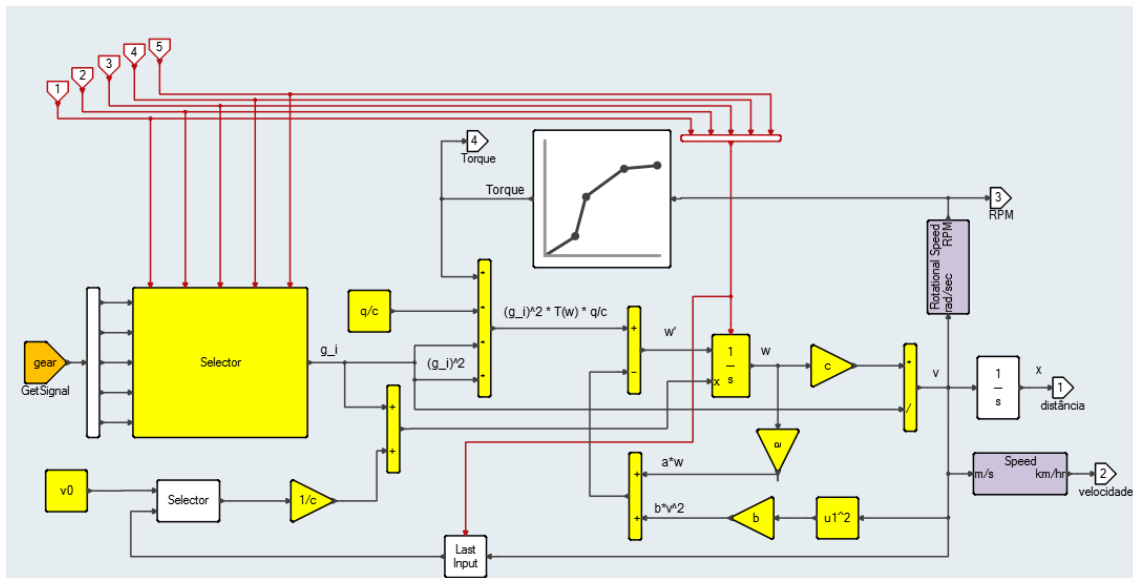


Figura 2: modelo do veículo no Activate

Nesta representação é importante notar que os componentes em vermelho representam elementos externos, assim os 5 links de ativação externa que entram no seletor modelam as trocas de marcha que podem ocorrer durante a simulação. De acordo com cada valor de entrada, o seletor tem como saída a relação de transmissão correspondente. A curva de torque é modelada através do bloco *Lookup table*, configurada conforme mostrado na *Figura 3* e com os parâmetros definidos na *Figura 4* (com os demais valores do problema).

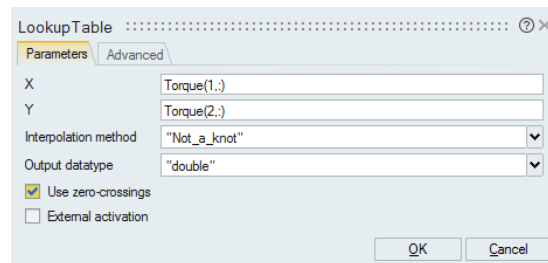


Figura 3: configuração da Lookup table do Torque

Uma das vantagens do uso do Activate é poder utilizar variáveis no diagrama de blocos e defini-las separadamente. Assim, qualquer alteração pode ser facilmente implementada modificando o valor em apenas um lugar do programa, sem ter que modificar cada bloco onde a variável se encontre.

Outro aspecto interessante deste primeiro modelo é a possibilidade de usar os blocos de conversão de unidades. Para a problema em questão, foram utilizadas as conversões de m/s para km/h e rad/s para rpm, mas o bloco permite uma série de conversões das mais diversas unidades, como pressão, massa, temperatura etc. Deste modo, fica facilmente visível para o usuário a mudança de unidade e não é preciso preocupar-se em inserir as constantes ou fórmulas de mudança no problema.

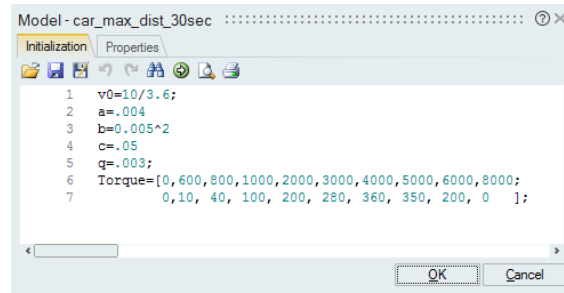


Figura 4: constantes definidas para a simulação

Depois de modelar o carro, pode-se utilizar o conceito de **superbloco** para transformar todo este diagrama em um único bloco que tem determinadas entradas e determinadas saídas especificadas pelo usuário. O superbloco traz a facilidade de deixar o problema segmentado, uma tela mais limpa e permite observar o sistema separadamente. Após aplicar esta mudança, todo o sistema da Figura 2 fica com o aspecto da Figura 5, mantendo as mesmas funcionalidades e gerando as mesmas saídas.

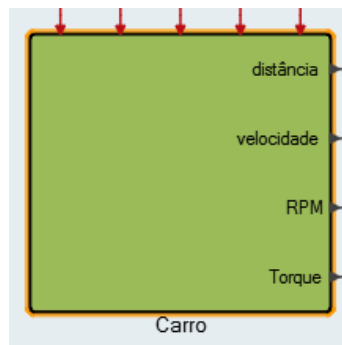


Figura 5: Superbloco representando o veículo

Otimizando o problema

Para o problema em questão, são consideradas variáveis de otimização as relações de transmissão de cada engrenagem (5 variáveis) e, por simplificação, o tempo de troca entre as marchas (4 variáveis). A função custo, que se deseja otimizar, é a distância percorrida, dado proveniente do superbloco Carro. Para fazer todo o trabalho de otimização, deverá ser usado o bloco BOBYQA, nativamente incorporado ao Activate.

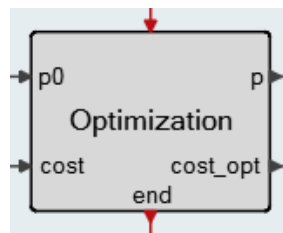


Figura 6: Bloco BOBYQA

O otimizador precisa ser configurado com os limites mínimos e máximos dos parâmetros (notando que as relações das engrenagens não podem ser nulas). Além disso, é possível definir se o desejável é maximizar ou minimizar a função custo.

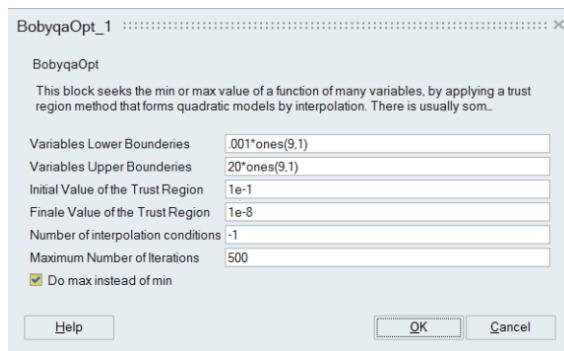


Figura 7: parâmetros do otimizador

O funcionamento do otimizador se dá da seguinte forma: Quando o tempo de 30s de simulação é atingido, o bloco otimizador (Figura 8), recebe os parâmetros de otimização *gear* (5 variáveis) e *T* (4 variáveis) e a função custo *distância* e retorna os novos valores. Isso é feito até atingir uma margem de erro aceitável e definida, quando a saída *end* chama o fim da simulação.

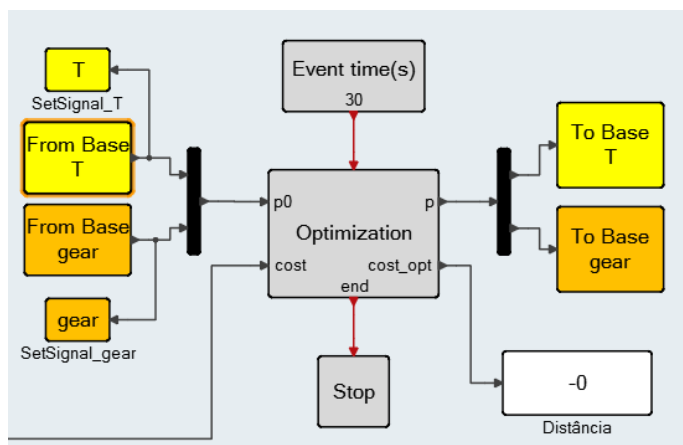


Figura 8: Otimizador, variáveis e saídas

É interessante notar neste caso, o uso dos blocos *From Base* e *To Base*. Estes dois blocos são utilizados para atualizar os valores das variáveis e mantê-las no sistema. A função deles é exportar e importar os valores para o *workspace* de modo que não são reiniciados a cada simulação. Neste exemplo, portanto, cada vez que o otimizador é executado, ele grava novo dados de *T* e *gear*, e depois estes dados são utilizados quando a simulação reinicia a cada 30 segundos. Esta é apenas uma das muitas formas de fazer importação/exportação no Activate.

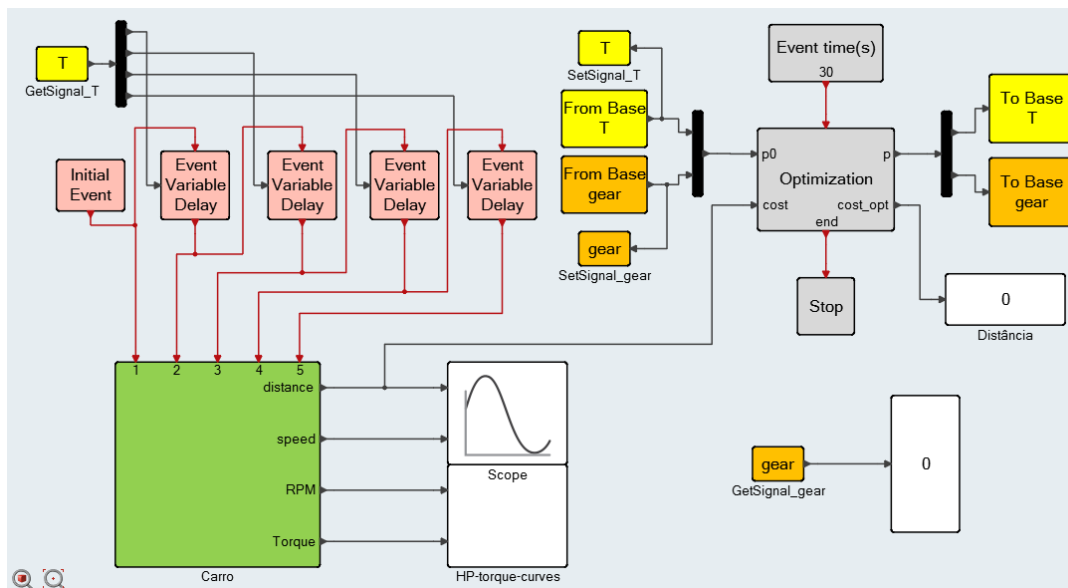


Figura 9: simulador completo

Por fim, na Figura 9, pode-se ver o simulador completo. Os blocos *setSignal* e *getSignal* são utilizados, entre outras possibilidades, para melhorar a visualização do diagrama de bloco, diminuindo a quantidade de linhas cruzando o diagrama. Os blocos de *Event Variable Delay* são utilizados para coordenar o tempo de troca dos eventos. Nesta configuração, só podem ser ativados após o bloco predecessor ter sido ativado e recebem os valores de *T* que estão gravados no *workspace* (e atualizados a cada iteração). Foram inseridos também um *scope* e mais um superbloco (Figura 10) contendo *scopes* para acompanhar visualmente as mudanças a cada nova simulação.

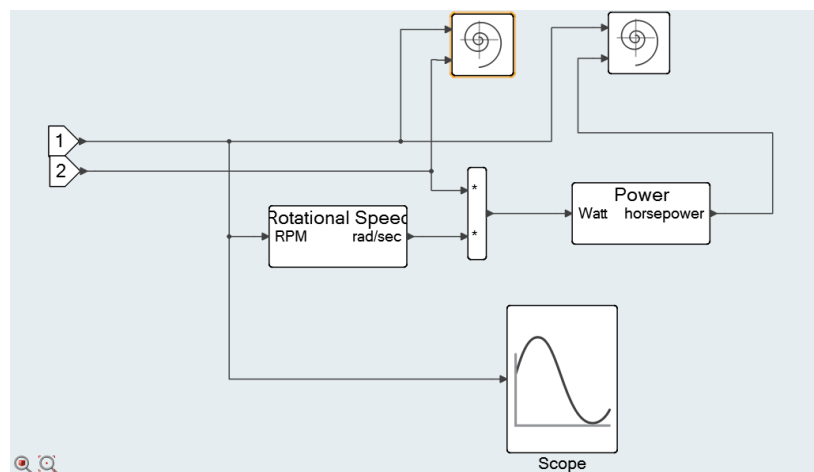


Figura 10: superbloco curvas de torque

Resultados

Em alguns segundos, o Activate é capaz de mostrar o melhor resultado para este caso. As informações vão sendo atualizadas e podem ser acompanhadas em tempo real pelo resultado através dos displays (Figura 11) e scopes (Figura 13 e Figura 14). Ao fim do processo, pode-se verificar o melhor resultado desta mesma forma. O workspace mostra o tempo e o status ao final da simulação (Figura 12) e neste espaço pode ser utilizado o comando *print* (mesmo pausando a simulação) para exibir quaisquer das variáveis que tenham sido exportadas para com o bloco *To Base*.

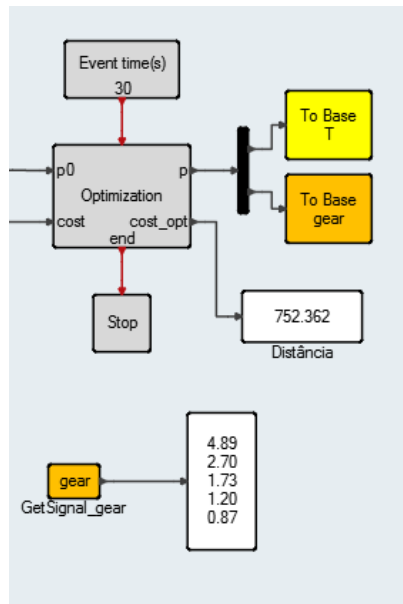


Figura 11: displays exibindo resultados

Pode-se verificar a maior distância que o veículo atinge em 30s para as configurações dadas. A simulação permite que se teste novos parâmetros com resultado da simulação em poucos segundos, o que permite ao projetista explorar novas possibilidades em curto espaço de tempo.

Nota-se, com este projeto, como é fácil fazer a integração da otimização do Activate com projetos já modelados. Ainda que os projetos tenham sido modelados em programas de terceiros, podem ser importados via Functional Mock-up Interface, desde que aceitem este padrão Modelica.

Por fim, o Activate oferece uma poderosa ferramenta, com a qual a simples inserção de alguns blocos, com comandos de arrastar e soltar, possibilita usufruir desta poderosa ferramenta de otimização e verificar o desempenho do sistema em tempo real.

```

Saving model [ car_max_dist_30sec ]
Starting Simulation
Info: Request a stop.
Block(s): End_1
Model: car_max_dist_30sec

Simulation [ car_max_dist_30sec ] information:
Result: Success
State: Stopped
Elapsed time: 39 seconds
Simulation time: 30

```

Figura 12: informações do workspace

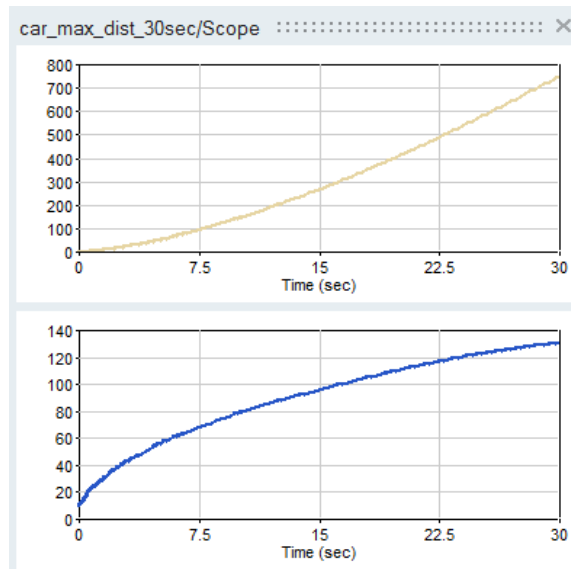


Figura 13: curvas de distância x tempo e velocidade x tempo

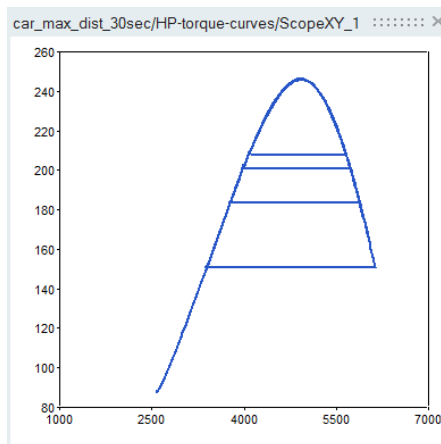


Figura 14: curva torque x rpm