# TÉCNICA DE IMPLEMENTAÇÃO

É implementada a técnica de *multishooting*, ou seja, as variáveis de decisão consideram não apenas as ações de controle , mas também os estados atuais e futuros. Assim, para o processo de otimização, ambas são vistas em todo o horizonte de predição () desejado.

Observar:

* Não se tem apenas uma ação de controle ótima, mas toda a trajetória ótima futura;
* e são as condições atuais dos estados e da ação de controle atualmente aplicada ao processo controlado.

Assim, considerando como sendo as variáveis de decisão, e um estado atual diretamente medido do processo (representado por ), a solução do NLP com a técnica *multishooting* pode ser escrito na forma:

sujeito às seguintes restrições de desigualdade:

|  |  |
| --- | --- |
|  | Eq. |

e outras restrições de igualdade que vão “impor” a dinâmica futura do sistema a partir de uma condição inicial , conhecida pela medição atual:

|  |  |
| --- | --- |
|  | Eq. |

Importante notar que restrições de igualdade são “imposições”, ou seja, tem de respeitar o valor definido. As restrições de desigualdade associam faixas, ou seja, graus de liberdade que refletem o espaço de busca da solução.

Observar ainda que, para o caso em que se tem o modelo de equações que representam o sistema dinâmico, o estado futuro pode ser obtido na forma e as saídas controladas por *setpoint* são alguns dos estados que foram extraídos para estarem disponíveis na saída através da função . A função nada mais é senão um vetor/ matriz para seleção dos estados que vão ser apresentados na saída.

Para o caso em que usamos uma técnica de I.A. qualquer para representar o modelo dinâmico, os estados futuros serão preditos como uma função das ações de controle atuais aplicadas nos estados atuais do sistema e as saídas controladas por *setpoint* são obtidas pela mesma função , mas agora representadas na forma de estimativas (acento circunflexo).

# OBJETOS:

Objetos definidos no CaSAdi, seja para a inicialização ou para contas durante a simulação. ISSO TEM A VER COM A CODIFICAÇÃO E POSSIVELMENTE SERÁ MELHORADO EM BREVE

|  |  |
| --- | --- |
| OBJ | DESCRIÇÃO |
| casadi\_solver | Criação do solver Casadi |
| Hp | Horizonte de predição |
| Hc | Horizonte de controle (HcHp) |
| nx | Número de variáveis de entrada (estados) do processo |
| ny | Número de variáveis (saída do processo) que são controladas por *setpoint* (no caso PChegada e Vazão) |
| nu | Número de variáveis manipuladas (ações de controle possíveis) - no caso, Freq e PMonAlvo |
| PassoMPC | Número de amostragens até a atuação do MPC (no caso = 3) |
| x0 | Condição inicial para os estados das variáveis (até 1+Hp) em cada operação do Solver |
| du0 | Condição inicial para guardar as ações de controle (DeltaU) em todo o horizonte (Hc) |
| ysp0 | Condição inicial para guardar os *setpoints* ótimos calculados para as variáveis controladas por *setpoint* |
| BufferDeltaU | Para guardar as últimas 14 variações na ação de controle |
| Predicao | Para guardar a predição no instante anterior |
| ModeloPreditor | Criação da variável para guardar modelo de preditor do processo e que será utilizada pelo solver para a predição |
| EstimaVazao | Função para carregar uma única vez a 'f\_Interpola\_casadi\_vazao\_sym' e para, com base na frequência e na PChegada (atual ou futura) poder proceder a estimativa da vazão (atual ou futura) |
| Funcao\_h | Para proceder a conta y=h(x) e obter as saídas em função da matriz h definida |
| lbx | Lower Bounds para os Estados do MPC |
| ubx | Upper Bounds para os Estados do MPC |
| lbg | Lower Bounds para as restrições [g] que forem criadas |
| ubg | Upper Bounds para as restrições [g] que forem criadas |
| contador | Variável para guardar o contador de passos de amostragem - usado para definir momentos de atuação do MPC |

# Formato CaSAdi

Para a implementação no modelo CaSAdi, os estados MPC devem ser colocados na forma [X(:); U(:); Ysp(:)], onde:

* Os estados do MPC precisam ser passados na forma de colunas, razão pela qual usamos a sintaxe (:).
* X terá a dimensão de *1+* linhas (estado atual + estados futuros de até ), e colunas, cada uma delas correspondentes a respectiva variável.
* U terá a dimensão de linhas (ação atual + ações calculadas do horizonte até ), e colunas, cada uma delas correspondentes a respectiva variável.
* Ysp terá a dimensão de uma linha e colunas. Neste caso, considerando que o *setpoint* para as variáveis controladas por *setpoint* é também uma variável de decisão, poderíamos incluir estas decisões para o horizonte futuro. Não consideramos isso na nossa função custo, pois:
  + As saídas nada mais são do que parte dos estados . Assim, como os estados futuros já estão sendo vistos e suas restrições (atuais e futuras) sendo tratadas, de uma forma indireta, as saídas, quaisquer que sejam as variáveis selecionadas pela matriz *h*, já estão sendo tratadas;
  + A formulação pela saída controlada por *setpoint* atende a teoria. No caso, porém, o foco não está no uso dos *setpoints* para estas variáveis, mas sim em seguir o alvo desejado pela engenharia, o que altera o foco do controlador e não parece haver motivação para tratar todo o Ysp futuro. Neste cenário, a variável Ysp que contém os *setpoint*s ótimos estarão disponíveis para visualização, mas observe que não fazem parte do que será oferecido para o controle da planta;
  + Inserir o horizonte futuro de Ysp nas variáveis de decisão certamente fará aumentar o esforço computacional do otimizador.

OBS: Neste início do documento mantivemos a ação de controle , mas no caso específico da aplicação, optamos por usar o (e não ) como variável de decisão. Isso será tratado na sequência deste doc e na a implementação .

# Formatação do problema de otimização

O problema de otimização do MPC deve atender as seguintes condições:

|  |  |
| --- | --- |
|  |  |
|  |  |
| Para efeito de simulação, extrairemos condições reais do processo para serem usadas como condições iniciais |  |

## Função custo

|  |  |
| --- | --- |
|  | Eq. |
|  |  |

Onde:

representa a diferença entre as saídas estimadas em todo e o valor de *setpoint* calculado pelo otimizador para as variáveis controladas por *setpoint*

corresponde ao erro atual entre as saídas medidas e a estimação das saídas feitas no instante anterior. Avaliado apenas no primeiro instante (não em todo horizonte futuro)

é a diferença entre a ação de controle aplicada na planta e os alvos definidos pela engenharia, em todo o

é a variação na ação de controle analisada em todo o futuro até

corresponde ao erro entre os estados medidos e a estimação dos estados feita no instante anterior. Avaliado apenas no primeiro instante (não em todo horizonte futuro)

## Variáveis de decisão do MPC e suas restrições em lbx/ubx:

É necessário lembrar que os chamados estados do MPC estão na forma: [ X(:); (:); Ysp(:) ], onde

* X terá a dimensão de [*1+*, todos com limites em [-inf, +inf]\*;
* terá a dimensão de  \*\*, todos com limites em [-dumax, dumax]\*\*\*;
* Ysp terá a dimensão , todos com limites em [-inf, +inf]

\* Observar que os estados X (variáveis do processo e mesmo as saídas que nada mais são senão parte dos estados) têm limites que mudam em função da frequência, neste caso, as restrições para estas variáveis serão tratadas em lbg/ubg e por isso foram “liberadas” em lbx/ubx.

\*\* Importante observar que em sendo , será necessário calcular a ação de controle ótima até o horizonte . Por outro lado, a ponderação das ações de controle na função custo deve considerar apenas até o horizonte .

\*\*\* Fizemos experimentos na tentativa de tratar por faixas, ou seja, dumax>dumin>0, o que corresponderia a busca do solver as faixas [-dumax até - dumin], [zero], [dumin até dumax]. Não vale a pena pois esta descontinuidade viola condições do solver, o qual assume a premissa de que as restrições são diferenciáveis, portanto, não devem existir descontinuidades no espaço de busca da solução.

## Restrições de igualdade em lbg/ubg

Observe que temos a função do estimador que nos oferece e as saídas controladas por *setpoint* são obtidas pela função . Assim, dada uma condição atual na entrada do processo e uma ação de controle atualmente aplicada, podemos estimar .

O papel do otimizador começa na busca da primeira ação de controle ótima () que atenderá as restrições definidas. Uma vez conhecido o estado futuro e a ação de controle ótima futura, é possível estimar um estado futuro para o passo seguinte . A sequência até o horizonte de predição desejado, nos leva a completar a seguinte tabela:

|  |  |  |
| --- | --- | --- |
| Estado | Ação Atual | Instante k futuro |
|  |  | k=0 (atual) |
|  | Cálculo de | k=1 |
|  | Cálculo de | k=2 |
|  |  |  |
|  | Cálculo de | k=Hp-1 |
|  |  | k=Hp |

Como comentado pela técnica do *multishooting*, as restrições de igualdade devem impor a dinâmica (ver Eq. 2). Assim, observada a primeira coluna da tabela anterior, podemos escrever as restrições de igualdade na forma:

Lembrar que o solver atua com variáveis simbólicas e variáveis de decisão. Conclua que a técnica *multishooting* propõe assumir os estados como variáveis de decisão, no entanto, não dá grau de liberdade para isso, fazendo com que os estados sejam exatamente iguais aos estados (atuais e preditos). Assim, a busca do solver focará nas demais variáveis de decisão, que no nosso caso são o (atual e futuro) e o *setpoint* para as variáveis controladas por *setpoint*.

## Restrições de desigualdade em lbg/ubg

Apenas para efeito de esclarecimento, quando usamos a expressão , é importante entender que estamos nos referindo ao instante atual e a qualquer outro instante futuro predito. Isso é importante, por exemplo, para entender que as ações de controle futuras também precisam considerar as restrições dinâmicas futuras, que dependem da própria ação de controle (frequência) futura. Da mesma forma, a estimação da vazão é feita com base na Frequência e na PChegada, portanto, estados futuros da vazão dependem de valores futuros da Frequência e da PChegada e as restrições futuras da futura vazão também precisam estar contempladas.

* + Os estados preditos (atuais e futuros) precisam estar dentro dos limites operacionais calculados em função da frequência, ou seja ,
  + As saídas controladas por *setpoint* (atuais e futuras) precisam estar dentro dos limites operacionais calculados em função da frequência, ou seja, ,
  + As variáveis de entrada a serem aplicadas no sistema (ações de controle que são Frequência e PMovAlvo, representadas como []), precisam respeitar os limites operacionais, ou seja,
  + A ação de controle correspondente a PMonAlvo () precisa respeitar os limites dinâmicos correspondentes a PChegada, que mudam em função da frequência, ou seja,
  + A variação no esforço de controle a ser aplicado, precisa respeitar os limites operacionais, ou seja, , sendo
  + Na ação de controle correspondente frequência, definida como , a variação máxima deve respeitar limites definidos pelo fabricante. No caso do JUB27, no máximo 1Hz a cada 7,5min

**Observação 1:**

Os limites mínimos e máximos das variáveis do processo são calculados de forma dinâmica, em função da frequência. Estes limites correspondem a valores de alarmes L e H definidos pela empresa. Considerando que há alarmes que podem causar trip da planta, a implementação do código deve considerar a possibilidade de uma margem de tolerância percentual definida pelo usuário, tal qual indica a formulação seguinte. A definição de uma *MargemPercentual* = ZERO faz com que os limites originalmente calculados sejam mantidos

**Observação 2:**

Para implementar a última restrição, sabemos que a variação máxima permitida para a frequência é de 1Hz (MaxDeltaHz=1) a cada 7,5min (TempoLimite=450s).

Das configurações originais do sistema, temos um PassoMPC = 3 e um tempo de amostragem (Ts) = 10s. Assim, o tempo de atuação do MPC é dado por TempoMPC = PassoMPC\*Ts = 3 \* 10 = 30s.

Em uma janela de 7,5min (TempoLimite=450s) a quantidade máxima de atuações do controlador sobre a frequência é dada por *QteMaxDeltaFreq* = TempoLimite/TempoMPC = 450/30 = 15.

Observar que existem duas ações de controle. Nesta formulação representaremos a ação correspondente a frequência pela variável e sua variação é dada por

Para implementação no instante *k*, teremos de ofertar ao controlador, um vetor com as últimas 14 atuações do controlador. Para calcular o , o otimizador deve então considerar as seguintes restrições:

Importante lembrar que esta restrição vale , ou seja, precisará ser vista para todo o horizonte futuro das ações de controle preditas, independentemente do tamanho do horizonte futuro.

# Implementação em código

## Obter dados no instante atual do processo

O bloco deve receber:

* Medições dos estados atuais da planta
* Informação atual da ação de controle que foi efetivamente aplicada na planta
* Alvos de engenharia definidos pelo usuário (Freq e PMonAlvo)

## Variáveis de decisão para o solver

A definição das variáveis de decisão para o solver será: [X; ; Ysp], em formato de coluna única

* X terá a dimensão de [*1+*
* terá a dimensão de
* Ysp terá a dimensão

A dimensão das variáveis de decisão será então: , lembrando que pelo método *multishooting*, a variável X é tratada com restrições de igualdade, ou seja, não requer busca do solver e as variáveis de decisão efetivamente utilizadas são o (em todo o horizonte ) e o Ysp.

## Parâmetros para serem enviados para o solver

Para cada chamado do solver, teremos de atualizar os parâmetros que lhe serão enviados, e que deverão compor um vetor coluna na forma:

Onde:

* deve conter a medição atual das variáveis do processo
* deve conter o valor da ação de controle atual efetivamente aplicada no processo
* deve conter o valor da variação na ação de controle que foi efetivamente aplicado no processo, mas a variável precisará levar também os valores para todo o
* é o alvo correspondente ao ponto de operação definido pela engenharia (valor para a Frequência e PMonAlvo desejados)
* corresponde ao erro entre os estados medidos e a estimação dos estados feita no instante anterior
* corresponde ao erro atual entre as saídas medidas e a estimação das saídas feitas no instante anterior
* traz os 14 últimos valores de efetivamente aplicados ao processo
* são os estados do reservatório da ESN que precisam ser atualizados a cada passo de amostragem para que a ESN não se perca ao longo do tempo.

Observar que:

* terá a dimensão de [*1+*
* terá a dimensão de
* terá a dimensão de
* terá a dimensão de
* terá a dimensão de [*1*
* terá a dimensão de [*1*
* terá a dimensão do reservatório da ESN utilizada com preditor no MPC