

1. TÉCNICA DE IMPLEMENTAÇÃO

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

Observar:

- Não se tem apenas uma ação de controle ótima, mas toda a trajetória ótima futura;
- x_0 é a condição/medição atual dos estados x
- u_0 é a ação de controle u atualmente aplicada ao processo controlado

Assim, considerando $\omega = [x_0, x_1 \dots x_{Hp}, u_0, u_1, \dots, u_{Hp-1}, \Delta u_1, \dots, \Delta u_{Hp-1}]$ como sendo as variáveis de decisão¹, e um estado atual diretamente medido do processo (representado por $x_m = x_0$) e ação de controle (u_0) atualmente na entrada da planta, a solução do NLP com a técnica *multipleshooting* pode ser escrita na forma:

$$\min_{\omega} \varphi(\omega)$$

sujeito às restrições de desigualdade que vão restringir o espaço de busca da solução:

$$g_1(\omega) = \begin{bmatrix} g(x_0, u_0) \\ g(x_1, u_1) \\ \vdots \\ g(x_{Hp-1}, u_{Hp-1}) \\ g(x_{Hp}) \end{bmatrix} \geq 0 \quad \text{Eq. 1}$$

e outras restrições de igualdade:

- para “impor” a dinâmica futura do sistema a partir de uma condição inicial x_m , conhecida pela medição atual
- para “impor” a variável de decisão Δu em função das variáveis de decisão u

$$g_2(\omega) = \begin{bmatrix} x_m - x_0 \\ f(x_0, u_0) - x_1 \\ \vdots \\ f(x_{Hp-1}, u_{Hp-1}) - x_{Hp} \\ (u_1 - u_0) - \Delta u_1 \\ \vdots \\ (u_{Hp-1} - u_{Hp-2}) - \Delta u_{Hp-1} \end{bmatrix} = 0 \quad \text{Eq. 2}$$

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 $x_{k+1} = f(x_k, u_k)$ e as saídas controladas por

¹ Observar que u_0 é a entrada atual da planta passado para o otimizador, assim como foi o x_0 . As medições são usadas para impor a técnica *multipleshooting* e o x_0 não muda, mas vale observar que u_0 , enquanto variável de decisão, vai ser alterada pelo solver, assim como todos os Δu !!

setpoint são alguns dos estados que foram extraídos para estarem disponíveis na saída através da função $y_{k+1} = h(x_{k+1})$. A função h 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: $\hat{x}_{k+1} = f(x_k, u_k)$ e as saídas controladas por *setpoint* são obtidas pela mesma função $\hat{y}_{k+1} = h(\hat{x}_{k+1})$, mas agora representadas na forma de estimativas (acento circunflexo).

2. OBJETOS:

Objetos definidos no CaSAdi, seja para a inicialização ou para contas durante a simulação.

OBJ	DESCRIÇÃO
casadi_solver	Criação do solver Casadi
Hp	Horizonte de predição
Hc	Horizonte de controle ($H_c \leq H_p$)
nx	Número de variáveis de entrada (estados) do processo
ny	Número de variáveis (saída do processo) que são controladas por <i>setpoint</i> (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)
Qy	Matriz para ponderar as saídas controladas por <i>setpoint</i>
Qu	Matriz para ponderar as ações de controle
R	Matriz para ponderar as variações nas ações de controle
Qx	Matriz para ponderar os erros de predição dos estados
x0	Condição inicial para os estados das variáveis medidas atuais e até o horizonte H_p (dimensão $1+H_p$)
u0	Condição inicial para guardar as ações de controle em todo o horizonte futuro (dimensão H_p)
BufferDeltaU	Para permitir somatório das últimas 15 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 as variáveis de decisão do MPC
ubx	Upper Bounds para as variáveis de decisão 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

3. Formato CaSAdi

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

- As variáveis de decisão do MPC precisam ser passadas na forma de colunas, razão pela qual usamos a sintaxe (:)
- X terá a dimensão de nx linhas e $1+Hp$ colunas (estado atual x_0 + estados futuros de x_1 até x_{Hp}). Assim, cada uma das colunas correspondente ao tempo de amostragem e cada uma das linhas corresponde a respectiva variável/estado;
- U terá a dimensão nu linhas e Hp colunas (ação atual u_0 que será ajustada + ações calculadas do horizonte u_1 até u_{Hp-1}). Assim, cada uma das colunas correspondente ao tempo de amostragem e cada uma das linhas corresponde a respectiva variável de entrada;
- ΔU terá a dimensão de nu linhas e $Hp - 1$ colunas. Assim, cada uma das colunas correspondente ao tempo de amostragem e cada uma das linhas correspondente a respectiva variável de entrada;

4. Formatação do problema de otimização

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

$\hat{x}(k+j) = \hat{f}(\hat{x}(k+j-1), u(k+j-1))$	$j = 1, \dots, H_p$
$\hat{y}(k+j) = \mathbf{h}(\hat{x}(k+j))$	$j = 1, \dots, H_p$
$\hat{x}_k = x_0$; $\hat{y}_k = y_0$ $u_k = u_0$ <p>Para efeito de simulação, extrairemos condições reais do processo para serem usadas como condições iniciais</p> <p>Já foi percebido que na maioria dos casos, inicializar uma variável de decisão com os valores futuros antes preditos, usualmente reduz o esforço computacional. Assim, a cada nova chamada solve, optamos por inicializar u_k com valores antes preditos. O Δu_k, por sua vez, pensamos em inicializar, mas concluímos ser desnecessário, uma vez que serão variáveis de decisão impostas por restrições de igualdade</p>	$k = 0$

4.1. Função custo genérica

$J_k(x, u) = \left(\sum_{j=1}^{H_p} \ \hat{y}(k+j) - \mathbf{y}_{sp} + e_y\ _{Q_y}^2 \right) + \left(\sum_{j=1}^{H_c} \ u(k+j) - u_{eng}\ _{Q_u}^2 \right)$	Eq. 3
$+ \left(\sum_{j=1}^{H_c-1} \ \Delta u(k+j)\ _R^2 \right) + \ e_x\ _{Q_x}^2$	

Onde:

- $Q_x \in \mathbb{R}^{nx \times nx}$
- $Q_y \in \mathbb{R}^{ny \times ny}$
- $Q_u \in \mathbb{R}^{nu \times nu}$
- $R \in \mathbb{R}^{nu \times nu}$

$\hat{y}(k+j) - y_{sp}$ representa a diferença entre as saídas estimadas em todo H_p e o valor de *setpoint* desejado para as variáveis controladas por *setpoint*. No caso específico, as variáveis eleitas para serem controladas por *setpoint* são a PChegada e a Vazão.

$e_y = y(k) - \hat{y}(k-1)$ 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)

$u(k+j) - u_{eng}$ é a diferença entre a ação de controle aplicada na planta e os alvos definidos pela engenharia, em todo o H_c

$\Delta u(k+j) = u(k+j) - u(k+j-1)$ é a variação na ação de controle analisada em todo o futuro até H_c

$e_x = x(k) - \hat{x}(k-1)$ 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)

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

É necessário lembrar que as variáveis de decisão para o solver do MPC estão na forma: $[X(:); U(:); \Delta U(:)]$, onde:

- X terá a dimensão de $[nx \times 1+H_p]$, todos com limites em $[\text{LimitesMin}, \text{LimitesMax}]^*$;
- U terá a dimensão de $[nu \times H_p]^{**}$, todos com limites em $[Umin, Umax]^{***}$;
- ΔU terá a dimensão de $[nu \times (H_p - 1)]$, todos com limites $[-\Delta Umax, \Delta Umax]$

* LimitesMin e LimitesMax foram definidos na inicialização apenas para restringir o espaço de busca. Observar ainda que os estados X (variáveis do processo) e mesmo as saídas que nada mais são senão parte dos estados, ambas têm limites que mudam em função da frequência (atual e futura), neste caso, as restrições para estas variáveis serão tratadas em lbx/ubg e por isso foram “liberadas” em lbx/ubx.

** Importante observar que em sendo $H_p > H_c$, será necessário calcular a ação de controle ótima até o horizonte $H_p - 1$. Por outro lado, em atendimento a teoria, a ponderação das ações de controle na função custo deve considerar apenas até o horizonte H_c . O mesmo ocorrerá com o ΔU , o qual vai compor a função custo apenas até $H_c - 1$

*** Observar que a PMonAlvo, na prática, é a proposição para a PChegada. A PChegada, por sua vez, tem restrições que são função da frequência (atual e futura). Assim, é importante que as restrições da entrada U associadas a PMonAlvo (atuais e futuras), sejam também avaliadas em função das restrições da PChegada. Isso será tratado nas restrições em lbx/ubg.

OBSERVAÇÃO:

Fizemos experimentos na tentativa de tratar o ΔU por faixas, ou seja, $\Delta Umax > \Delta Umin > 0$, o que corresponderia a busca do solver as faixas $[-\Delta Umax \text{ até } -\Delta Umin]$, $[zero]$, $[\Delta Umin \text{ até } \Delta Umax]$. 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.

4.3. Restrições de igualdade em lbg/ubg

Observe que temos a função do estimador que nos oferece $\hat{x}_{k+1} = f(x_k, u_k)$ e as saídas controladas por *setpoint* são obtidas pela função $\hat{y}_{k+1} = h(\hat{x}_{k+1})$. Assim, dada uma condição atual x_0 na entrada do processo e uma ação de controle u_0 atualmente aplicada, podemos estimar $\hat{x}_1 = f(x_0, u_0)$.

O papel do otimizador começa na busca da primeira ação de controle u_1 ótima (u_1^*) que atenderá as restrições definidas. Uma vez conhecido o estado futuro \hat{x}_1 e a ação de controle ótima futura u_1^* , é possível estimar um estado futuro para o passo seguinte $\hat{x}_2 = f(\hat{x}_1, u_1^*)$. A sequência até o horizonte de predição desejado², nos leva a completar a seguinte tabela:

Estado Dimensão = $1 + Hp$	Ação Atual Dimensão = Hp	Δu Dimensão = $Hp - 1$	Instante k futuro
$x_k = x_0 = x_m$	$u_k = u_0$		k=0 (atual)
$\hat{x}_1 = f(x_0, u_0)$	Cálculo de u_1^*	$\Delta u_1 = u_1^* - u_0$	k=1
$\hat{x}_2 = f(\hat{x}_1, u_1^*)$	Cálculo de u_2^*	$\Delta u_2 = u_2^* - u_1^*$	k=2
\vdots	\vdots	\vdots	\vdots
$\hat{x}_{Hp-1} = f(\hat{x}_{Hp-2}, u_{Hp-2}^*)$	Cálculo de u_{Hp-1}^*	$\Delta u_{Hp-1} = u_{Hp-1}^* - u_{Hp-2}^*$	k=Hp-1
$\hat{x}_{Hp} = f(\hat{x}_{Hp-1}, u_{Hp-1}^*)$			k=Hp

Como comentado pela técnica do *multipleshooting*, as restrições de igualdade devem impor a dinâmica (ver Eq. 2). Da mesma forma, serão usadas restrições de igualdade para impor $\Delta u(k)$ e deixar apenas as ações de controle como reais variáveis de decisão. Assim, observada a primeira e a terceira colunas da tabela anterior, podemos escrever as restrições de igualdade na forma:

$$\begin{bmatrix} x_m - x_0 \\ \hat{x}_1 - x_1 \\ \vdots \\ \hat{x}_{Hp} - x_{Hp} \\ (u_1 - u_0) - \Delta u_1 \\ \vdots \\ (u_{Hp-1} - u_{Hp-2}) - \Delta u_{Hp-1} \end{bmatrix} = 0$$

4.4. Restrições de desigualdade em lbg/ubg

Apenas para efeito de esclarecimento, quando usamos a expressão $\forall k$, 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 (especialmente 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.

² O texto ilustra a sequência para as predições e ações de controle futuras, no entanto, é importante lembrar que pela técnica *multishooting*, não é feito um passo por vez, mas todos os passos de uma vez só !!

- Os estados preditos (atuais e futuros) precisam estar dentro dos limites operacionais calculados em função da frequência, ou seja $x_{min}(Frequencia(k)) \leq x(k) \leq x_{max}(Frequencia(k))$, $\forall k$
- As saídas controladas por *setpoint* (atuais e futuras) precisam estar dentro dos limites operacionais calculados em função da frequência, ou seja, $y_{min}(Frequencia(k)) \leq y(k) \leq y_{max}(Frequencia(k))$, $\forall k$
- As variáveis de entrada a serem aplicadas no sistema (ações de controle que são Frequência e PMovAlvo, representadas como $[u_{Freq}(k) u_{Pmon}(k)]$), precisam respeitar os limites operacionais, ou seja, $u_{min} \leq u(k) \leq u_{max}$ $\forall k$
- A ação de controle correspondente a PMonAlvo ($u_{Pmon}(k)$) precisa respeitar os limites dinâmicos correspondentes a PChegada, que mudam em função da frequência, ou seja, $u_{min}(PChegada(k)) \leq u_{Pmon}(k) \leq u_{max}(PChegada(k))$ $\forall k$
- A variação no esforço de controle a ser aplicado, precisa respeitar os limites operacionais, ou seja, $-\Delta u_{max} \leq \Delta u(k) \leq \Delta u_{max}$ $\forall k$, sendo $\Delta u(k) = u(k) - u(k-1)$
- Na ação de controle correspondente frequência, definida como $u_{Freq}(k)$, 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. Sugerimos atuar, pelo menos, com *MargemPercentual* = 1.

$$LimiteMax = LimiteMax * \left(1 - \frac{MargemPercentual}{100}\right)$$

$$LimiteMin = LimiteMin * \left(1 + \frac{MargemPercentual}{100}\right)$$

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 tempo de amostragem (T_s) = 10s, de modo que os 450s correspondem a 45 amostragens.

Observar que existem duas ações de controle. Nesta formulação representaremos a ação correspondente a frequência pela variável $u_{Freq}(k)$ e sua variação é dada por $\Delta u_{Freq}(k)$

Para implementação no instante k , teremos de ofertar ao controlador, um vetor com as últimas 44 atuações $\Delta u_{Freq}(k)$ do controlador³. Para calcular o $\Delta u_{Freq}(k+1)$, o otimizador deve então considerar as seguintes restrições:

³ Apesar do controlador só atuar de 30 em 30s, optamos por fazer a conta a cada amostragem de 10s, mesmo sendo o caso de somar $\Delta u_{Freq} = 0$. Isso, pois, nos casos em que o controlador der *unfeasible*, ele tentará uma nova ação no ciclo seguinte. Assim, não há sincronismo com os 30s, mas sim com o período de amostragem de 10s.

$$\left| \left(\sum_{j=1}^{44} \Delta u_{Freq}(k+1-j) \right) + \Delta u_{Freq}(k+1) \right| \leq 1$$

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

5. Implementação em código

5.1. Obter dados no instante atual do processo

O bloco CaSAdi deve receber:

- $X0$ = medições dos estados atuais da planta
- $U0$ = Informação atual da ação de controle que foi efetivamente aplicada na planta
- Alvos de engenharia definidos pelo usuário (Freq e PMonAlvo)

5.2. Variáveis de decisão para o solver

A inicialização das variáveis de decisão para o solver será: $[X; U; \Delta U]$, em formato de coluna única

- X terá a dimensão de $[nx \times (1 + Hp)]$
- U terá a dimensão de $[nu \times Hp]$
- ΔU terá a dimensão de $[nu \times (Hp - 1)]$

A dimensão das variáveis de decisão será então: $nx \cdot (1 + Hp) + nu \cdot Hp + nu \cdot (Hp - 1) + ny$, lembrando que pelo método *multipleshootig*, a variável X é tratada com restrições de igualdade, ou seja, não requer busca do solver e as variáveis de decisão efetivamente utilizadas são as ações de controle U em todo o horizonte Hp . Também vale registrar que, sem perda de generalidade, podemos inicializar ΔU com zeros, já que, apesar de serem variáveis de decisão, não haverá busca destas variáveis (atuais e futuras), uma vez que vão ser valores impostos por restrições de igualdade.

IMPORTANTE:

- Tentamos usar apenas U ou apenas ΔU como variável de decisão, mas não foi possível tratar de forma isolada. Isso ocorre pelo fato de que precisamos de decisões que restrinjam diretamente o ΔU em limites máximos, assim como, as restrições dinâmicas futuras precisam da variável de decisão em U , atual e futura. Neste contexto, entendemos ser necessário passar ambas como variáveis de decisão. Por outro lado, como há uma relação direta entre elas, ΔU foi tratada com restrições de igualdade
- No código implementado, para padronizar e facilitar o entendimento de todos, **AS INFORMAÇÕES NO TEMPO “k” ESTARÃO EM COLUNAS**, ou seja, as linhas representam as variáveis e cada coluna da matriz corresponderá ao respectivo estado atual/futuro destas variáveis.

5.3. 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:

$$P = [x_0 ; u_0 ; AlvoEng ; Ysp ; e_x ; e_y ; Buffer\Delta Freq ; Reservatório_ESN]$$

Onde:

- x_0 deve conter a medição atual das variáveis do processo (dimensão nx)
- u_0 deve conter a ação de controle atualmente aplicada na planta (dimensão nu)

- *AlvoEng* é o alvo correspondente ao ponto de operação definido pela engenharia (valor para a Frequência e PMonAlvo desejados) (dimensão nu)
- *Ysp* são as saídas consideradas desejadas para as variáveis controladas por *setpoint* (PChegada e Vazão)
- $e_x = x(k) - \hat{x}(k - 1)$ corresponde ao erro entre os estados medidos e a estimação dos estados feita no instante anterior (dimensão nx)
- $e_y = y(k) - \hat{y}(k - 1)$ corresponde ao erro atual entre as saídas medidas e a estimação das saídas feitas no instante anterior. (dimensão ny)
- *BufferDeltaFreq* tem dimensão 45 e traz o valor de $\Delta Freq$ atual proposto além dos 44 últimos valores de $\Delta Freq$ efetivamente aplicados ao processo
- *Reservatório_ESN* 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. (dimensão nx_ESN que corresponde ao tamanho do reservatório da ESN utilizada)

OBSERVAÇÃO:

Assumir que o valor *Ysp* desejado para a saída referente a PChegada é o próprio valor da PMonAlvo dado pela engenharia, assim como, para a saída que se refere a Vazão, vamos assumir que o set de vazão é a vazão estimada para o ponto da Frequência e PMonAlvo dados pela engenharia.