1. TÉCNICA DE IMPLEMENTAÇÃO

É implementada a técnica de *multishooting*, 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 e u_0 são as condições atuais dos estados x e da ação de controle u atualmente aplicada ao processo controlado.

Assim, considerando $\omega = \left[x_0, \ldots x_{Hp}, u_0, \ldots, u_{Hp-1}\right]$ como sendo as variáveis de decisão, e um estado atual diretamente medido do processo (representado por $x_m = x_0$), a solução do NLP com a técnica *multishooting* pode ser escrito na forma:

$$\underbrace{\min}_{\omega} \varphi(\omega)$$

sujeito às seguintes restrições de desigualdade:

$$g_1(\omega) = \begin{bmatrix} g_1(x_0, u_0) \\ g_1(x_1, u_1) \\ \vdots \\ g_1\left(x_{Hp-1}, u_{Hp-1}\right) \\ g_1\left(x_{Hp}\right) \end{bmatrix} \geq 0 \qquad \textit{Eq. 1}$$

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

$$g_2(\omega) = \begin{bmatrix} x_m - x_0 \\ f(x_o, u_o) - x_1 \\ \vdots \\ f(x_{Hp-1}, u_{Hp-1}) - x_{Hp} \end{bmatrix} = 0 \qquad \textit{Eq. 2}$$

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 *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. 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 (Hc≤Hp)
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*
u0	Valor da ação de controle que foi efetivamente aplicada no processo
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

3. 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 $1{+}Hp$ linhas (estado atual x_0 + estados futuros de x_1 até x_{Hp}), e nx colunas, cada uma delas correspondentes a respectiva variável.
- U terá a dimensão de Hp linhas (ação atual u_0 + ações calculadas do horizonte u_1 até u_{Hp-1}), e nu colunas, cada uma delas correspondentes a respectiva variável.
- Ysp terá a dimensão de uma linha e ny 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 y nada mais são do que parte dos estados x. 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 u, mas no caso específico da aplicação, optamos por usar o Δu (e não u) como variável de decisão. Isso será tratado na sequência deste doc e na a implementação .

4. Formatação do problema de otimização

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

$\hat{\boldsymbol{x}}(k+j) = \hat{\boldsymbol{f}}(\hat{x}(k+j-1),\ u(k+j-1))$	$j = 1, \dots H_p$
$\hat{y}(k+j) = \boldsymbol{h}(\hat{x}(k+j))$	$j = 1, \dots H_p$
$\hat{x}_k = x_0$ $\hat{y}_k = y_0$ $u_k = u_0$ Para efeito de simulação, extrairemos condições reais do processo para serem usadas como condições iniciais	$k = 0$

4.1. Função custo

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

Onde:

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

$\hat{\mathbf{y}}(k+j) - \mathbf{y}_{sp}$ representa a diferença entre as saídas estimadas em todo Hp e o valor de *setpoint* calculado pelo otimizador para as variáveis controladas por *setpoint*

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

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

$e_x = x(k) - \hat{x}(k-1)$ 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 os chamados estados do MPC estão na forma: [X(:); Δu(:); Ysp(:)], onde

- X terá a dimensão de $[1{+}Hp \times nx]$, todos com limites em [-inf, +inf]*;
- Δu terá a dimensão de $[Hc \times nu]$**, todos com limites em [-dumax, dumax]***;
- Ysp terá a dimensão $[1 \times ny]$, 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 $Hp > Hc$, será necessário calcular a ação de controle ótima até o horizonte $Hp - 1$. Por outro lado, a ponderação das ações de controle na função custo deve considerar apenas até o horizonte Hc.

  *** 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.

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 desejado, nos leva a completar a seguinte tabela:

Estado	Ação Atual	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^*	k=1
$\hat{x}_2 = f(\hat{x}_1, u_1^*)$	Cálculo de u_2^*	k=2
$\vdots$	$\vdots$	$\vdots$
$\hat{x}_{Hp-1} = f(\hat{x}_{Hp-2}, u_{Hp-2}^*)$	Cálculo de u_{Hp-1}^*	k=Hp-1

$\hat{x}_{Hp} = f(\hat{x}_{Hp-1}, u^*_{Hp-1})$		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:

$$\begin{bmatrix} x_m - x_0 \\ \hat{x}_1 - x_1 \\ \vdots \\ \hat{x}_{Hp} - x_{Hp} \end{bmatrix} = 0$$

Lembrar que o solver atua com variáveis simbólicas e variáveis de decisão. Conclua que a técnica *multishooting* propõe assumir os estados x 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 Δu (atual e futuro) e o *setpoint* para as variáveis controladas por *setpoint*.

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 (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 $x_{min}(Frequencia(k)) \leq x(k) \leq x_{max}(Frequencia(k))$, $\forall k$
- 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

<u>Observação 1:</u>

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

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

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

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 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 $u_{Freq}(k)$ 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 14 atuações $\Delta u_{Freq}(k)$ do controlador. Para calcular o $\Delta u_{Freq}(k+1)$, o otimizador deve então considerar as seguintes restrições:

$$\left| \left(\sum_{j=1}^{14} \Delta u_{Freq}(k+1-j) \right) + \Delta u_{Freq}(k+1) \right| \leq 1$$

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 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)

5.2. Variáveis de decisão para o solver

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

- X terá a dimensão de $[1{+}Hp \times nx]$
- Δu terá a dimensão de $[Hc \times nu]$
- Ysp terá a dimensão $[1 \times ny]$

A dimensão das variáveis de decisão será então: $(1 + Hp).nx + Hc.nu + ny$, 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 Δu (em todo o horizonte Hc) e o Ysp.

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:

$$P = [x_0 \ ; \quad u_0 \quad ; \quad \Delta u_0 \ ; \quad AlvoEng \quad ; \ e_x \ ; \quad e_y \ ; \ Buffer\Delta u \ ; \ Reservatório\_ESN \ ]$$

Onde:

- x_0 deve conter a medição atual das variáveis do processo, mas a variável precisará levar também os valores para todo o Hp
- u_0 deve conter o valor da ação de controle atual efetivamente aplicada no processo
- Δu_0 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 Hc
- $AlvoEng$ é o alvo correspondente ao ponto de operação definido pela engenharia (valor para a Frequência e PMonAlvo desejados)
- $e_x = x(k) - \hat{x}(k-1)$ corresponde ao erro entre os estados medidos e a estimação dos estados feita no instante anterior
- $e_y = y(k) - \hat{y}(k-1)$ corresponde ao erro atual entre as saídas medidas e a estimação das saídas feitas no instante anterior
- $Buffer\Delta u$ traz os 14 últimos valores de Δu 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.

Observar que:

- x_0 terá a dimensão de $[1{+}Hp \times nx]$
- u_0 terá a dimensão de $[1 \times nu]$
- Δu_0 terá a dimensão de $[Hc \times nu]$
- $AlvoEng$ terá a dimensão de $[1 \times nu]$
- e_x terá a dimensão de $[1 \times nx]$
- e_y terá a dimensão de $[1 \times ny]$
- $Reservatório\_ESN$ terá a dimensão do reservatório da ESN utilizada com preditor no MPC