4
Proposta deMPC com Aprendizado por Reforço

4.1.
Introdução

Conforme mencionado no Capítulo 1, este trabalho teve como objetivo o desenvolvimento e a implementação de uma estratégia de controle de produção de reservatórios petrolíferos baseada em controle preditivo e em modelo com aprendizado por reforço ou *Model preditive control – Reinforcement Learning* (MPC-RL). Este controle é efetuado em nível global, com o planejamento da operação de todos os poços produtores do reservatório e também com o planejamento da operação dos poços injetores, o que o caracteriza como um controle em *slow loop*. (Campos, Silva, *et al.*, 2006)

A partir do controle de operação dos poços, deseja-se estender o tempo de vida dos mesmos e com isto otimizar a recuperação de óleo (Saputelli, Nikolaou e Economides, 2003). Assume-se, para este propósito, que o sistema de recuperação de óleo do reservatório petrolífero utiliza poços inteligentes que podem ser controlados e monitorados. Em uma visão mais global, com o desenvolvimento de poços inteligentes, isto é, de poços com instrumentação na perfuração, surge a possibilidade de alcançar um gerenciamento da produção do campo inteiro que permite, entre outras coisas, aumentar a vida útil do campo, e também dos poços (Tupac e Talavera, 2009), (Campos, Silva, *et al.*, 2006).

4.2.
Modelo MPC-RL

O modelo MPC-RL proposto consta de três partes principais:

1. Um modelo identificado do processo. Neste caso a *Proxy*, como mencionado no Capítulo 1, é o modelo identificado da planta que fornece as previsões do modelo do processo.

2. Um modelo de controle que encontra a melhor política frente à previsão fornecida pelo modelo do processo.
3. O processo a ser controlado ou planta. Neste caso, o modelo proposto pode ser não-linear e multivariável.

A Figura 12, ilustra o modelo proposto, onde as subseções seguintes cada bloco será descrito detalhadamente com o sentido de controlar a produção de óelo de um reservatório.

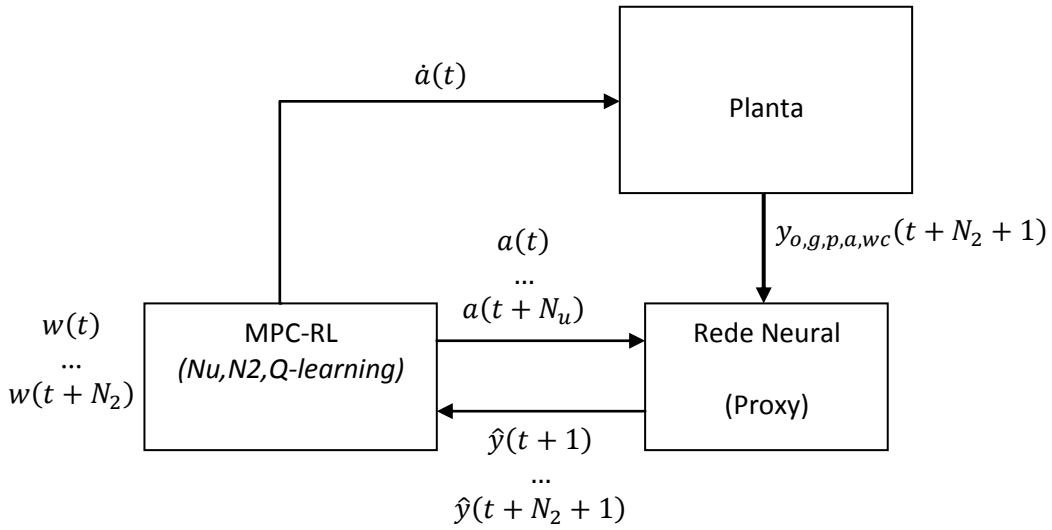

Figura 12 – Modelo MPC-RL proposto.

4.2.1. Planta Estudada

A planta estudada neste trabalho foi um reservatório petrolífero. Onde o problema consiste em um sistema de extração de óleo por recuperação secundária (injeção de água) o qual é implementado usando um modelo de reservatório sintético de três camadas e dois poços: um poço injetor com três completações independentes e um poço produtor também com três completações, uma em cada camada. A Figura 13 ilustra o reservatório e os poços.

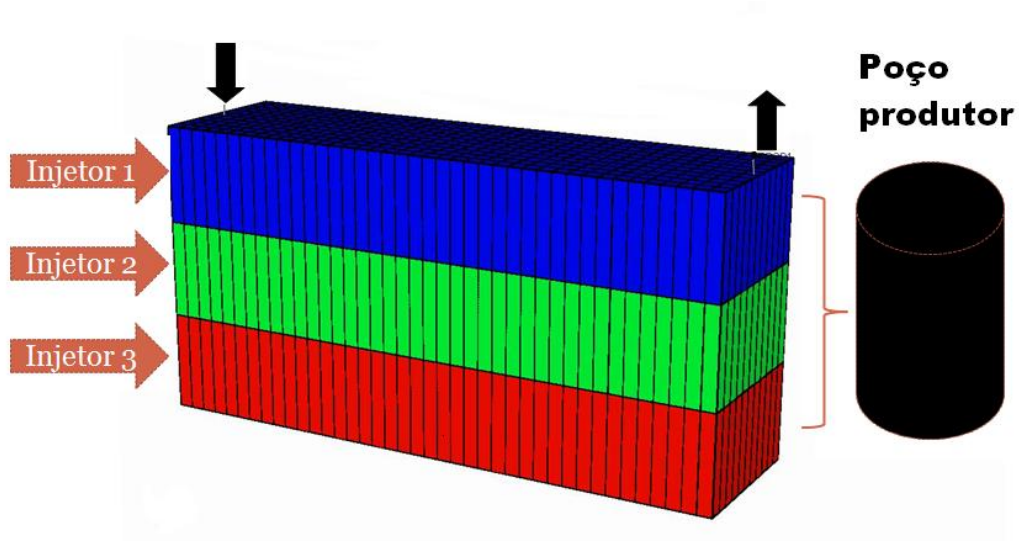

Figura 13 – Reservatório estudado.

O poço da esquerda, com seta para baixo é o injetor com três completações independentes, uma por cada camada. O poço da direita, com seta para cima é o poço produtor. As variáveis selecionadas do reservatório são: pressão média no reservatório, taxas de produção diária de óleo, gás, água e *water cut*. Que podem ser medidas tanto no poço produtor (saída), como no poço injetor (entrada).

Além disso, tem-se três variáveis de entrada (uma por cada camada), que representam as taxas de injeção de fluxo de água do reservatório, medidas em cada completação do poço injetor.

4.2.2. Proxy do Reservatório

Este bloco indica que deve existir uma modelagem matemática da dinâmica do comportamento do reservatório. O presente trabalho propõe o emprego de *proxies*, isto é, de modelos que são ajustados a partir de dados históricos da planta. A utilização da inteligência computacional em áreas de petróleo para o modelo de reservatórios, em especial redes neurais, estão sendo utilizadas com mais freqüência devido a seu processamento de informação paralela e distribuída.

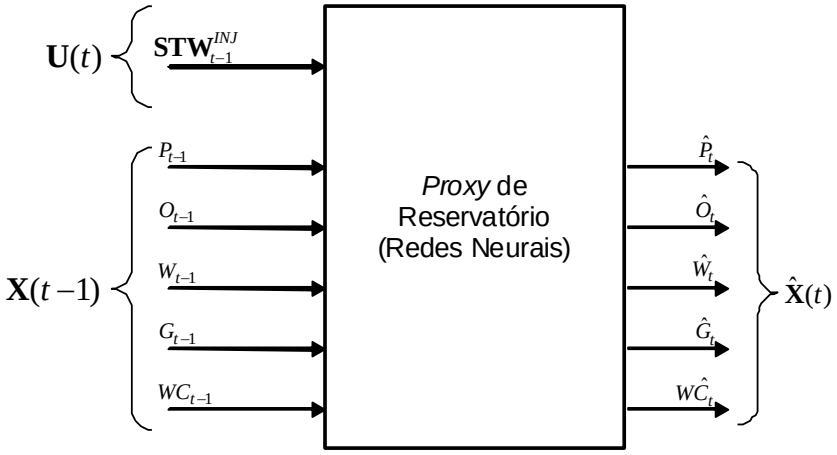

Figura 14 – Proxy do reservatório.

A rede utilizada foi uma rede MLP *Multi-Layer Perceptron, Back-propagation* com algoritmo de treinamento *Levenberg-Marquardt* (LM). Para a identificação foram empregadas as seguintes variáveis: as variáveis de controle **U**(*t*) que são as taxas de injeção **STW**=(STW_1, STW_2, STW_3), sendo três, uma para cada camada, a pressão média do reservatório *P*(*t*-1), taxas de produção diária de óleo, gás, água e *water cut*: *O*(*t*-1), *G*(*t*-1), A(*t*-1), e *WC*(*t*-1), que representa a razão entre a água produzida e o total de líquidos produzidos pelos poços. A idéia é ter um modelo identificado que permita obter previsões de *P*(*t*), *O*(*t*), *G*(*t*), A(*t*), e *WC*(*t*), dadas as injeções de água **STW**(*t*). Uma vez obtidas estas previsões, novos valores **STW** podem ser testados para os próximos passos (*t*+1, *t*+2, *t*+3...) tais que possam ser obtidas previsões *multistep*.

A Figura 14 mostra o modelo macro da *proxy* modelada. Durante os processos de treinamento, observou- se que as variáveis de *A(t)* e *Wc*(t) são as mais difíceis de prever a partir das entradas em (*t*-1) e das variáveis de controle *U(t)* (ou *STW(t)*). Para solucionar este problema foram utilizadas as dependências dos valores *P(t)*, *O(t)* e *G(t*), os quais foram incluídos como entradas nos modelos para prever *A(t)* e *WC(t)*. Assim, a configuração apresentada na Figura 15, foi a que apresentou melhores aproximações para as produções em tempo (t).

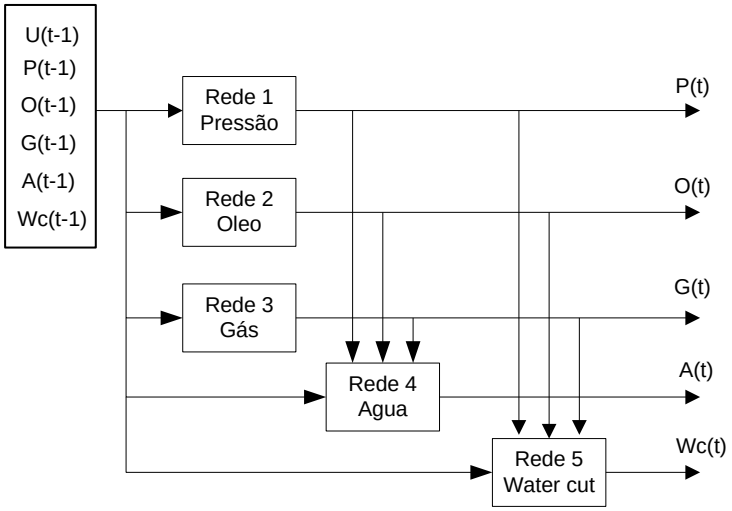

Figura 15 – Configuração da Proxy proposta

4.2.3. Configurações do RL e MPC

Nesta seção a Tabela 1 descreve as configurações do aprendizado por reforço e controle preditivo onde: os estados são os erros relativos *e*, *SP* é o *setpoint* ou referência desejada da taxa da produção de óleo e *y* é a saída da taxa da produção de óleo. Os erros considerados são [-70 -40 -20 -10 -5 -2 0 2 5 10 20 40 70] divididos entre o *setpoint* desejado, onde os erros são mais ajustados ou sintonizados perto do zero e mais espaçados nos extremos.

Tabela 1 – Configuração do agente RL.

RL	**Problema Controle**
Estados	Erro relativo: (e(t) = SP-y(t))/SP
Ações	Valores de Injeção nas completações controladas
Ambiente	Modelo de reservatório e simulador IMEX
Prêmio Imediato	Função custo
Política	ε-greedy
α	0.8
γ	0.7
Épsilon	1
Algoritmo RL	Q-Learning
Passos na frente	3

O premio imediato do agente RL é:

$$r = \frac{1}{1 + |setpoint - saida|^2} \tag{4.26}$$

A Figura 16 apresenta o algoritmo *ε-greedy*, onde: *Q* é a matriz gerada pelo aprendizado por reforço, contendo os estado e ações; *s* é o estado atual; *actions* são as ações propostas (**Tabela 2**); a função *GetBestAction* escolhe a melhor ação no estado *s* da matriz *Q;* e a função *randint* escolhe aleatoriamente o índice de alguma ação.

```
if (rand()>epsilon)
    a = GetBestAction(Q,s);
else
    a = randint(1,1,actions)+1;
end
epsilon = epsilon*0.95;
```

Figura 16 – Algoritmo do *ε-greedy*.

As ações têm valores de abertura da válvula entre 0 a 450 m^3/dia, cada vetor de ações das válvulas do poço produtor inteligente não deve passar 450 m^3/dia, por condições do reservatório. O MPC tem um horizonte de controle de 3 passos a frente, e o horizonte de previsão é de 91 passos segundo o explicado na seção (4.2.2).

4.2.3.1. Melhora de *ε*-greedy

No modelo MPC-RL, propõe-se uma nova estratégia para acelerar a política *ε-greedy*, isto se fez para diminuir o custo computacional que o MPC-RL precisa para atingir o resultado desejado.

Para isso modificou-se a estratégia da política *ε-greedy*. Primeiro são testadas todas as ações da Tabela 2 no *proxy*, escolhendo a ações que apresentem menor erro, isso se faz com alguns episódios iniciais para cada passo de simulação.

Tabela 2 - Ações do agente RL.

Ações	Válvula 1	Válvula 2	Válvula 3
a_1	0	0	0
a_2	112.5	50	112.5
a_3	50	50	50
a_4	50	112.5	112.5
a_5	50	50	175
a_6	112.5	112.5	50
$a7$	50	50	112.5
a_8	175	50	50
a_9	50	175	50
a_{10}	112.5	50	50

Em seguida, nos próximos episódios utilizam-se *ε-greedy*, como consequência disso obtém-se como resultado umas boas sementes inicias na tabela *Q-learning*. Assim sendo, o tempo de simulação é reduzido notavelmente, como se apresentara no capitulo 5.

4.3. Algoritmo MPC-RL

O modelo MPC-RL proposto é dividido em duas etapas, a etapa de aprendizado e a de aplicação do conhecimento do agente RL. Na primeira etapa, MPC e RL interagem entre si colaborativamente. O aprendizado por reforço ajuda ao MPC a calcular a função custo no senso de um modelo de decisão de Markov ao mesmo tempo o agente *aprende* o controle do processo e atualiza seu conhecimento. Nessa etapa o MPC colabora basicamente com a robustez do modelo e toma a decisão ao longo de um intervalo curto de tempo. Essa etapa é apresentada na Figura 12 (página 58, seção 4.2).

Nessa primeira etapa o agente RL começa aplicando uma seqüência de ações em cada passo de tempo t, em um horizonte de controle N_u a fim de levar de um estado inicial s_0 a um estado desejado s_t minimizando ou maximizando a função custo do MPC. O agente RL recebe alguma representação do ambiente

(modelo não-linear explícito do MPC, em nosso caso a *proxy*) $s_t \in S$ onde S é o conjunto de todos os possíveis estados. Nessa base seleciona-se a ação $a_t \in \mathcal{A}(s_t)$, onde $\mathcal{A}(s_t)$ é o conjunto de ações Tabela 2 avaliadas no estado s_t após um passo do tempo por conseqüência dessa ação o agente RL recebe um prêmio numérico $r_{t+1} \in \mathcal{R}$, encontrando assim, um novo estado s_{t+1}. Neste sentido o agente segue armazenando o conhecimento adquirido em uma tabela *Q(s,a)* armazenado o conhecimento utilizando *Q-learning* como foi apresentado na seção (2.3.2.3.2).

O episódio nesse modelo é o número de vezes que o horizonte de controle N_u foi programado, ou seja, se $N_u = 5$, e o episódio é 1000. Logo, se tem um total de 5000 simulações para achar a melhor política de controle para o primeiro passo de tempo do horizonte de predição N_2 (cada passo de tempo do processo). Assim, quanto maior a quantidade de episódios, mais o agente RL irá aprender e atualizar seu conhecimento. No entanto, aumentar o número de episódios influencia diretamente no tempo computacional.

Após cada passo de tempo do processo, o agente RL calcula o valor do somatório da função valor *Q-learning* de cada episódio e armazena a informação conforme especificado na Tabela 3. Assim, quando o primeiro passo de tempo do processo for finalizado, será escolhido o valor ótimo, ou seja, o maior valor acumulado pela função *Q-learning* (Max ∑Qn). Somente é aplicada a primeira ação da seqüência ótima, no primeiro estado, e o ciclo começa de novo para o seqüente passo do processo.

Tabela 3 – Conhecimento do agente RL.

	at	at+1	at+2	...	at+9
St	Q1(st,at)	Q3(st,at)	Q2(st,at)		Q4(st,at)
St+1	Q4(st+1,at+1)	Q1(st+1,at+1)	Q3(st+1,at+1)		Q2(st+1,at+1)
St+2	Q4(st+2,at+2)	Q1(st+2,at+2)	Q2(st+2,at+2)		Q3(st+2,at+2)
...					
St+12	Q1(st+n,at+n)	Q4(st+n,at+n)	Q3(st+n,at+n)		Q2(st+n,at+n)
	∑Q1	∑Q2	∑Q3		∑Q4

O algoritmo do modelo proposto na etapa de aprendizado fica como apresentado na Figura 17.

Algoritmo 1: Algoritmo proposto.

```
1  Inicializar Q(s,a);
2  Inicializar N2;
3  Observar s;
4  inicializarHorizonteControle Nc;
5  Fazer:
6  * Escolher a e executar no modelo;
7  * Receber r;
8  * Observar s';
9  * Atualizar Q(s,a);
10 s' = s;
11 Finalizar Nc;
12 Escolher a1 pertencente ao SUM(Q(s,a)) otimos;
13 Executar a1 ao processo;
14 Ir ao seguinte passo N2;
15 Fim N2;
```

Figura 17 – Algoritmo proposto.

Na segunda etapa, á aplicação do conhecimento do agente RL, é ilustrado pela Figura 18, onde o conhecimento adquirido na tabela $Q(s,a)$ é utilizado para controlar o processo sem a necessidade do MPC e do modelo explícito da planta (*Proxy*). Para cada ação a_t que o agente RL execute em um estado inicial s_0 o processo (IMEX, simulador do reservatório petrolífero) atribuirá o novo estado ao agente, e este consultará na tabela *Q(s,a)*, que apontará qual será a melhor ação que leve o estado atual ao estado desejado.

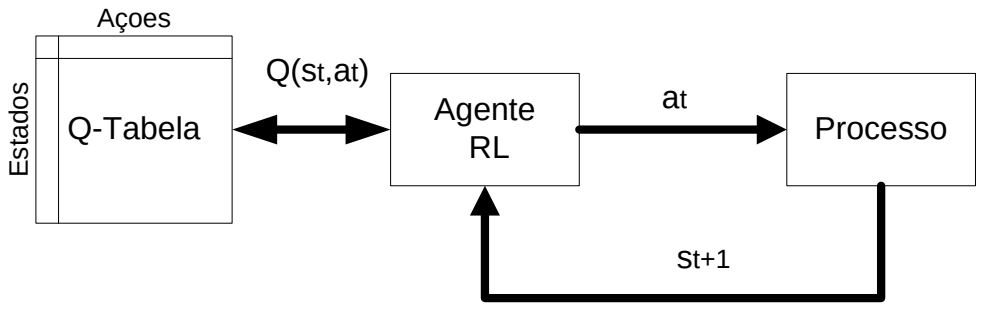

Figura 18 – Etapa de aplicação do conhecimento do agente RL.