

SPE-206102-MS

Real-Time Optimal Resource Allocation and Constraint Negotiation Applied to A Subsea Oil Production Network

Risvan Dirza, Sigurd Skogestad, and Dinesh Krishnamoorthy, Norwegian University of Science and Technology

Copyright 2021, Society of Petroleum Engineers

This paper was prepared for presentation at the 2021 SPE Annual Technical Conference and Exhibition held in Dubai, UAE, 21 - 23 September 2021.

This paper was selected for presentation by an SPE program committee following review of information contained in an abstract submitted by the author(s). Contents of the paper have not been reviewed by the Society of Petroleum Engineers and are subject to correction by the author(s). The material does not necessarily reflect any position of the Society of Petroleum Engineers, its officers, or members. Electronic reproduction, distribution, or storage of any part of this paper without the written consent of the Society of Petroleum Engineers is prohibited. Permission to reproduce in print is restricted to an abstract of not more than 300 words; illustrations may not be copied. The abstract must contain conspicuous acknowledgment of SPE copyright.

Abstract

This paper considers the problem of steady-state optimal resource allocation in an industrial symbiotic oil production network, or in general, a large-scale oil production system network, where different organizations share common resources.

These allocation problems are typically solved in a distributed optimization framework, where the optimization problem is decomposed into smaller subproblems, a central coordinator is used to coordinate the different subproblems. However, the use of a central coordinator may introduce additional practical challenges, such as impartiality issues, or additional operating costs, which is undesirable even in the technological selection phase.

To eliminate the need for a central coordinator, this paper proposes a consensus-based optimal resource allocation, where each subproblem or organization is locally optimized, and the coupling constraints are negotiated among the different organizations over a fixed communication network with limited information exchange.

The proposed approach is applied to a large-scale subsea oil production system, where the different wells are operated by different organizations. The simulation results of the application show that the proposed approach can optimally allocate the shared resources.

Introduction

Optimal process operation involves taking decisions in real-time to meet production goals, that is typically done in the context of real-time optimization (RTO). RTO, also known as daily production optimization (DPO), requires process models and real-time measurements. Initially, DPO is performed offline and manually. Bieker et. al., 2006 stated that the use of mathematical tools has shown production improvement by 2 %. Moreover, Chui et. al., 2018 reported that the use of Artificial Intelligence (AI) tools in the oil and gas industry may improve the performance up to 79 % beyond the provided analytical techniques. Thus, it is promising to utilize online and ‘smart’ RTO in day-to-day oil production operations. Due to the similarity in terms of context, we use DPO and RTO interchangeably.

With increasing energy demand, market competition, the complexity of process system, and the necessity for sustainable energy production, there is an explicit need for the oil industry to focus on resource and

energy efficiency. Thus, it is expected to obtain optimal performance as a single entity instead of separated organizations. Industrial symbiosis is an enabling technology towards sustainable energy production, where different organizations share energy and resources in a mutually beneficial manner.

However, industrial symbiosis creates additional challenges that may impede practical application. For example, DPO requires process models and real-time measurements, which the different companies or organizations may not be willing to share due to intellectual property rights, process simplification, market competitiveness, etc. Thus, there is a need to optimally allocate the shared resources with limited information sharing across the different organizations. This kind of problem may occur when we operate a subsea oil production network.

Operating subsea oil production network can be a challenging and risky task that several organizations or even operator companies cooperate to operate the production oilfield. During the exploitation period or even since exploration period, a case of remote reservoir location finding can exist. To exploit the oil, constructing a local processing platform may not be an economic option. One possible solution is by doing subsea tie-in and share common processing facilities. This solution has several subsea clusters (satellite wells) that may be operated by different organizations or companies with different local objective. Moreover, the location of the clusters can be far away such that centralizing the coordination may not be an economic option, too.

Distributed optimization methods such as the dual (Lagrangian) decomposition approach have been shown as a potential solution for this resource sharing (Wenzel et al. 2016). In this framework, the different subsystems representing different organizations, are locally optimized for a given shadow price of the shared resources, and a central coordinator updates the shadow price for the shared resource to match the supply and demand of the shared resource in this micro-market setting. This only requires sharing limited information, such as shadow prices and total resource consumption. This coordination scheme imitates the tâtonnement process, where the coordinator has the responsibility to find the equilibrium price iteratively by dynamic pricing based on responses of the subsystems (Walker 1987 and Wenzel et al. 2016). To this end, the different subproblems and the central coordinator are solved iteratively until the problem converges to a feasible and optimal solution.

However, having a central coordinator, that updates the shadow price of the shared resource, may create additional challenges that impede practical implementation. Since the shadow price is an important variable that affects the local objectives of the different subsystems, such a role may lead to impartiality issues and conflict of interest. Moreover, it may also lead to additional costs, such as technical support for the central coordinator, or cost of communication. Furthermore, if the coordinator or the communication from the central coordinator fails, then the entire system may fail to achieve system-wide optimal operation. To this end, these issues lead to profit loss.

An alternative approach to eliminate the need for a central coordinator is by directly coordinating among the different subsystems over a fixed communication network. This also further minimizes the information exchange, since each subsystem communicates data with only a few subsystems, instead of broadcasting it to the central coordinator. A similar intuitive cooperative game was implemented in interconnected Model Predictive Controller (MPC) in Maestre et al. (2011) where each MPC cooperates with the other MPCs via a communication channel to find the best input set-points for reference tracking. Cooperative game models have also been proposed to find optimal set-points or trajectories of shared resources that will systematically and gradually (via the neighborhood) obtain the optimal set points in the whole networked system (Arif et al. 2017).

Although various forms of distributed RTO/DPO with a central coordinator has been studied for optimal resource allocation problems (see Dirza et al. 2021, Gunnerud et al. 2010, Krishnamoorthy 2021, Marti et al. 2012, Wenzel et al. 2016), decentralized frameworks without a central coordinator (such as Banjac et al. 2019) has received relatively little attention.

The main contribution of this paper is a consensus-based decentralized real-time optimization, denoted as a consensus-RTO/DPO framework, applied to an oil and gas production network, where the different

for virtual subsystem 0 that store unutilized shared resources, and $A_0 \in \mathbb{R}^{m_x \times n_x}$ is a matrix that couples virtual subsystem 0 to the ‘physical’ subsystems. Thus, $\sum_{i=1}^N A_i x_i - \bar{x} \leq 0$, that represent inequality constraint.

For simplified illustration, consider a sample case where the available share resource of three physical subsystems is $\bar{x} = 2.75$, the coupling matrices are $A_0 = 1, A_1 = 1, A_2 = 2$, and $A_3 = 0$. The obtained optimal decision variables are $x_0^* = 0.25, x_1^* = 0.5, x_2^* = 2$, and $x_3^* = 2$. This result shows that, physically, total utilized shared resource is $\sum_{i=1}^3 A_i x_i = 2.5$, which is less than $\bar{x} = 2.75$. The unutilized shared resource is stored in virtual subsystem 0 where $A_0 x_0 = 0.25$.

Remark 1: In practical context, we can see Problem (1) as an optimization problem where we aim to maximize oil production rate and minimize gas lift rate consumption subject to given shared resources constraints, process models and physical constraints.

Remark 2: Each subsystem i may also have local constraints that are assumed to be locally managed by each organization and are not explicitly shown in the problem formulation (1).

Each subsystem i may also have local constraints that are assumed to be locally managed by each organization and are not explicitly shown in the problem formulation (1).

The objective of the problem (1) is to determine optimal shared resource allocation in order to achieve system-wide steady-state optimal operation in a decentralized fashion with limited information sharing over a fixed communication network. This can be described by an undirected graph $\mathcal{G} = (\mathcal{S}, \mathcal{E}, \mathcal{V})$ with the set of subsystems $\mathcal{S} = (1, 2, \dots, N)$, set of communication channel $\mathcal{E} \subseteq \mathcal{S} \times \mathcal{S}$, and an adjacency matrix $\mathcal{V} = [v_{i,j}]$. The communication channel of \mathcal{G} is denoted by $e_{i,j} = \{i, j\}$, meaning that there exists a data transmission between subsystems i and j . Since we consider an undirected graph, $e_{i,j} = e_{j,i}$. The adjacency elements associated with the communication line of the graph are positive, i.e. $e_{i,j} \in \mathcal{E} \Leftrightarrow v_{i,j} = 1$. Consequently, $e_{i,j} \notin \mathcal{E} \Leftrightarrow v_{i,j} = 0$, and we consider $v_{i,i} = 0$ for all i .

The set of neighboring connected plant of subsystem i is defined as follows.

$$\mathcal{N}_i = \{j \in \mathcal{S} : e_{i,j} = (i, j) \in \mathcal{E}\} \quad (2)$$

In other words, the set \mathcal{N}_i represents the set of subsystems that directly communicates with subsystem i . Fig. 1(b) illustrates the definition of neighboring subsystems.

The degree of the subsystem i is defined as

$$d_i = \sum_{j=1}^N v_{i,j} \quad (3)$$

which denotes the number of directly connected neighbours of subsystem i .

We assume that undirected graph G is connected. Halin (1969) provides comprehensive definition of connected graph. Thus, in order to use the proposed method, we need to look at if the graph G is connected by evaluating the rank of its Laplacian matrix, L . G is connected, if and only if $\text{rank } L(G) = N-1$, where $L = D - VL$, and $[D]_{i,i} = d_i$.

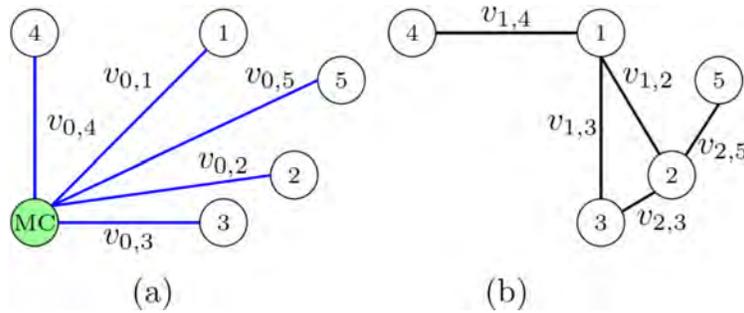


Figure 1—(a) (Centralized-) distributed RTO with central coordinator (MC), (b) Decentralized RTO, where $N_1 = \{2,3,4\}$, $N_2 = \{1,3,5\}$, $N_3 = \{1,2\}$, $N_4 = \{1\}$, and $N_5 = \{2\}$.

Methodology

In this paper, we aim to solve the optimal resource allocation problem (1) in a decentralized manner without a central coordinator using consensus-based approach. To this end, we want to propose a framework that so-called consensus-based decentralized RTO/DPO (consensus-RTO/DPO) as shown in Fig. 2 and Algorithm 1.

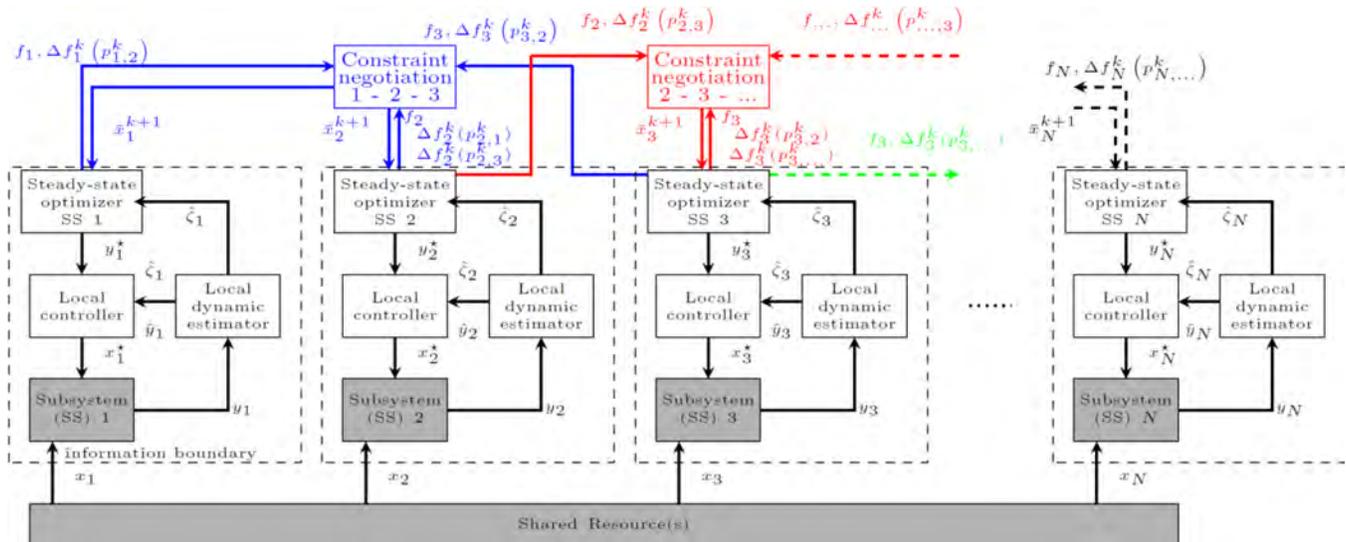


Figure 2—The schematic illustration of the proposed consensus-RTO.

Algorithm 1: Consensus-RTO/DPO Algorithm

Algorithm 1—Consensus-RTO Algorithm

- given step length δ and initial constraint for each subsystem $i, \bar{x}_i^0, i \in \mathcal{N}$
- for every sampling time k do
1. solve local RTO/DPO problem (4) to get optimal decision and expected cost values: $[x_i^{k+1}(\bar{x}_i^k), f_i(x_i^{k+1}(\bar{x}_i^k))]$.
 2. for negotiations with every neighboring subsystem do
 - a. execute Algorithm 2 to compute predicted rewards.
 - b. receive predicted rewards from neighboring subsystem, $\Delta f_j(p_{j,i}^k)$.
 - c. compute common predicted rewards, $\Delta \hat{p}_{i,j}^k = \Delta f_i(p_{i,j}^k) + \Delta f_j(p_{j,i}^k)$.
 - d. determine the best decisions $(p_{i,j}^*, p_{j,i}^*)$ based on one-step prediction using Table 1
 - e. compute fraction of constraint update as resource receiver by using Eq. (12). end

3. compute fraction of constraint update as resource distributor by using Eq. (13).
4. update constraints by using Eq. (11).

end

Table 1—Possible Decisions and One-step Prediction

$p_{i,j}^k = -p_{j,i}^k$	Δf_i	Δf_j	$\Delta \hat{g}_{i,j}^k$
0	$\hat{f}_i(x_i^{k+2}(\bar{x}_i^k)) - f_i(x_i^{k+1}(\bar{x}_i^k))$	$\hat{f}_j(x_j^{k+2}(\bar{x}_j^k)) - f_j(x_j^{k+1}(\bar{x}_j^k))$	$\Delta f_i(p_{i,j}^k) + \Delta f_j(p_{j,i}^k)$
1	$\hat{f}_i(x_i^{k+2}(\bar{x}_i^k + \delta)) - f_i(x_i^{k+1}(\bar{x}_i^k))$	$\hat{f}_j(x_j^{k+2}(\bar{x}_j^k - \delta)) - f_j(x_j^{k+1}(\bar{x}_j^k))$	$\Delta f_i(p_{i,j}^k) + \Delta f_j(p_{j,i}^k)$
-1	$\hat{f}_i(x_i^{k+2}(\bar{x}_i^k - \delta)) - f_i(x_i^{k+1}(\bar{x}_i^k))$	$\hat{f}_j(x_j^{k+2}(\bar{x}_j^k + \delta)) - f_j(x_j^{k+1}(\bar{x}_j^k))$	$\Delta f_i(p_{i,j}^k) + \Delta f_j(p_{j,i}^k)$

In Fig. 2, grey box represents the physical system and white box represents the computation block. Blue box and lines represent constraint negotiation process between subsystem 2 and its neighbor (N_2). Red box and lines represent constraint negotiation process between subsystem 3 and N_3 . Green lines represent information transmission from subsystem ‘...’ as a neighbor in a constraint negotiation process for other subsystems. As local controllers, MPC needs estimated states, labelled by \hat{y}_i and estimated disturbance, $\hat{\zeta}_i$. For PID as local controllers, \hat{y}_i which is an outputs measurement, thus dynamic estimator is unnecessary for Proportional Integral Derivative (PID) controller but important to estimate disturbance $\hat{\zeta}_i$ for the optimizer. The dashed line represents information boundary, where inside the information boundary, we have local RTO/DPO. This local RTO/DPO estimate the parameters or disturbance regularly using local dynamic estimator. The local steady-state optimizer updates its parameters and/or disturbance information using the estimated parameters and/or disturbance given by local dynamic estimator.

Remark 3: In practical context, several commercial softwares such as PROSPER, GAP and IPM by Petex, to name a few, are available to executed estimation (i.e. dynamic estimator) based on measured variables and (steady state) optimization (Bakshi et al., 2015). Thus, local RTO/DPO tools are commercially available, and this work focuses on introducing an alternative method to coordinate them. Meanwhile, for local controllers, we can consider Distributed Control Systems (DCS) in practical context.

Algorithm 2: Compute Predicted Rewards

Algorithm 2—Compute Predicted Rewards

```

for every possible decision do
    1. solve one-step prediction local RTO/DPO problem (9) get one-step-prediction optimal decision and expected cost
       values:  $\left[ x_i^{k+2}(\hat{\Delta}^{k+1}(p_{i,j}^k)), \hat{f}_i(x_i^{k+2}(\hat{\Delta}^{k+1}(p_{i,j}^k))) \right]$ 
    2. compute predicted rewards using Eq. (10).
end

```

To find local solution, consider an optimization problem of subsystem i at iteration k .

$$P_i(x_i^{k+1}(\bar{x}_i^k)) := \min_{x_i^{k+1}(\bar{x}_i^k)} f_i(x_i) \quad (4a)$$

subjects to:

$$1) \text{ shared resources constraints: } A_i x_i - \bar{x}_i^k \leq 0 \quad (4b)$$

$$2) \text{ local process model} \quad (4c)$$

$$3) \text{ local physical constraints} \quad (4d)$$

where \bar{x}_i^k and f_i are shared resources constraints and local objective of subsystem i at iteration k , respectively. Note that since \bar{x}_i^k is time varying, we denote local objective as $f_i(x_i^{k+1}(\bar{x}_i^k))$. The optimal solution of problem (4) is $x_i^{k+1}(\bar{x}_i^k)$.

For the constraint negotiation, subsystem i and its neighbour $j \in \text{Ni}$ have a common cost function as follows.

$$g_{i,j}^k(x_i(\bar{x}_i), x_j(\bar{x}_j)) = f_i(x_i(\bar{x}_i)) + f_j(x_j(\bar{x}_j)) \quad (5)$$

Obtaining optimal decision x_i^{k+1} and x_j^{k+1} respectively, we can mathematically express the constraint negotiation problem as follows.

$$\min_{\bar{x}_i, \bar{x}_j} p_i(x_i^{k+1}(\bar{x}_i^k)) + p_j(x_j^{k+1}(\bar{x}_j^k)) \quad (6a)$$

subject to:

$$\bar{x}_i + \bar{x}_j = \bar{x}_i^k + \bar{x}_j^k \quad (6b)$$

Since $x_i(\bar{x}_i)$, it is not easy to solve problem (6) directly in order to compute \bar{x}_i . In order to circumvent this issue, we discretize the decision space \bar{x}_i and \bar{x}_j and generate a look up table in order to find the best \bar{x}_i and \bar{x}_j .

Starting from initial (guess) shared resources constraints in subsystem i , labelled by \bar{x}_i^0 , and assuming one shared resources constraint, to generate the look up table of one-step prediction, we consider three discrete possible actions, $P_{i,j} = \{-1, 0, 1\}$, namely,

$$\hat{x}_{i,j}^{k+1}(p_{i,j}^k) = \begin{cases} \bar{x}_i^k & \text{if } p_{i,j}^k = 0 \\ \bar{x}_i^k + \delta & \text{if } p_{i,j}^k = 1 \\ \bar{x}_i^k - \delta & \text{if } p_{i,j}^k = -1 \end{cases} \quad (7)$$

and the condition to its neighbor is as follows,

$$\hat{x}_{i,j} = \bar{x}_i^k + \bar{x}_j^k - \hat{x}_{j,i} \quad (8)$$

where $\delta > 0$ is user defined step length. The size of δ will determine the speed of convergence and the steady-state condition's accuracy. In general, $|\mathcal{P}_{i,j}| = 3^{m_x}$. Note that we have three possible values for $\hat{x}_{i,j}$ bar given by Eq. (7) and corresponding values for $\hat{x}_{j,i}$ given by Eq. (8).

Each subsystem will solve a one step prediction with the three possible actions to compute their corresponding objective function value, by solving the following problem.

$$x_i^{k+1}(\hat{x}_{i,j}^k) = \arg \min f_i(x_i) \quad (9a)$$

subjects to:

$$1) \text{ shared resources constraints: } A_i x_i - \hat{x}_{i,j}^k \leq 0 \quad (9b)$$

$$2) \text{ local process model} \quad (9c)$$

$$3) \text{ local physical constraints} \quad (9d)$$

As we have three members in $P_{i,j}$, problem (9) is solved three times in each subsystem to generate the look up table as shown in Table 1.

Where $\Delta f_i(p_{i,j}^k)$ is the estimated ‘rewards’ that can be obtained by executing the following equation.

$$\Delta f_i(p_{i,j}^k) = \hat{f}_i(x_i^{k+2}(\hat{x}_{i,j}^{k+1}(p_{i,j}^k))) - f_i(x_i^{k+1}(\bar{x}_i^k)) \quad (10)$$

By summing up the expected rewards of each subsystems within the neighborhood, we obtain predicted common cost function difference, $\Delta \hat{g}_{i,j}^k$, that we want to minimize using the look up table. Hence, we obtain the agreed decisions combination, labelled by $(p_{i,j}^*, p_{j,i}^*)$, that give the lowest predicted common cost function.

From a game theory point of view, at each time step both subsystems are playing a cooperative game. The right most cells contain the predicted common cost functions difference for a particular combination of future local constraint. At each iteration, the option that yields a lower predicted common cost difference is chosen. Note that both subsystems share this information since they are neighbors, hence, they both choose the same combinations of actions.

Note that without any specific condition, the number of possible action combinations is 3^{d_i+1} . When the negotiated subsystems reach their shared resources constraints, the number of combinations will decrease as some of the actions may lead to an infeasible solution. Moreover, some combinations will never yield to the lowest $\Delta \hat{g}_{i,j}^k$.

Based on the agreed decisions combinations, subsystem i can update its constraints as follows.

$$\bar{x}_i^{k+1} = \bar{x}_i^k + \left(w_{g,i} + \sum_{j \in \mathcal{N}_i} w_{r,i,j} \right) \delta \quad (11)$$

where $w_{r,i,j}$ is the fraction of constraint update because d_j may be more than 1, and $w_{g,i}$ is the fraction of constraint update because d_i may be more than 1.

Fraction $w_{r,i,j}$ and $w_{g,i}$ can be computed by using eq. (12) and (13), respectively.

$$w_{r,i,j} = \frac{\Delta \hat{g}_{i,j}^k}{\Delta \hat{g}_{i,j}^k + \sum_{i \in \mathcal{N}_j} \Delta \hat{g}_{j,i}^k} p_{i,j}^* \quad (12)$$

$$w_{g,i} = \begin{cases} \frac{\bar{w}_{g,i}}{2} & \text{if } |w_{g,i}| > 1 \\ \bar{w}_{g,i} & \text{otherwise} \end{cases} \quad (13)$$

Where $\mathcal{N}_j \triangleq \{i : i \in \mathcal{N}_j, p_{i,j}^* = p_{j,i}^*\}$, and $\bar{w}_{g,i} = \max[P_{i,j}] + \min[P_{i,j}]$, and $P_{i,j} \triangleq \{p_{i,j} : j \in \mathcal{C}_i\}$.

The updated shared resources constraint of subsystem i , is then used to solve the local problem (4). This iteration will be executed in real-time fashion to minimize $\Delta \hat{g}_{i,j}^k$, until the steady-state condition is achieved, where p_i^* and $\forall i \in \mathcal{S}$ keep choosing 0 at every iteration which means that consensus has achieved i.e.

$$\sum_{(i,j) \in \mathcal{E}} \Delta \hat{g}_{i,j}^k = 0$$

By iteratively solving problem (4) and executing constraint negotiations, we can avoid the need of central coordinator. Moreover, to execute constraint negotiation, we do not need local information such as the detail models, measurements, and local constraints, across the different subsystems. The only information that

needs to be shared in the negotiation are the local objective and estimated ‘rewards’ within the neighborhood N_i only.

Local measurements y_i for each subsystem i are used for online model update and local setpoint tracking control, and this information is not shared with any of the other subsystems as shown in Fig. 2.

The model update step is performed using the dynamic model, such that one can use transient measurements and hence does not need to wait for the local subsystem to reach steady-state condition (Krishnamoorthy et al. 2018).

Any suitable parameter estimator using dynamic models can be used for the online model update by augmenting the states and the parameters/disturbances $\hat{\zeta}_i$ as explained in Simon (2006).

Each subsystem may also have local controllers such as PID controllers or Model Predictive Control (MPC) to track the setpoint provided by the local RTO/DPO, as shown in Fig. 2.

Implementation

Case Descriptions

In this section, we apply the proposed consensus-RTO approach on a subsea gas-lifted oil production well network with $N = 5$ subsystems (subsea clusters), operated by different organizations as shown in Fig. 3 and Table 2. The overall network G and its neighboring sets (N_i) is given by Fig. 1(b) and Table 2. Rank $L(G) = 4$, indicating that G is a connected graph/network. Note that, if we want to create a centralized approach in a given network shown in Fig. 1(b), it is required to install another communication channel to create an entity as a central coordinator. This installation work can be costly and undesirable especially for marginal oil production field.

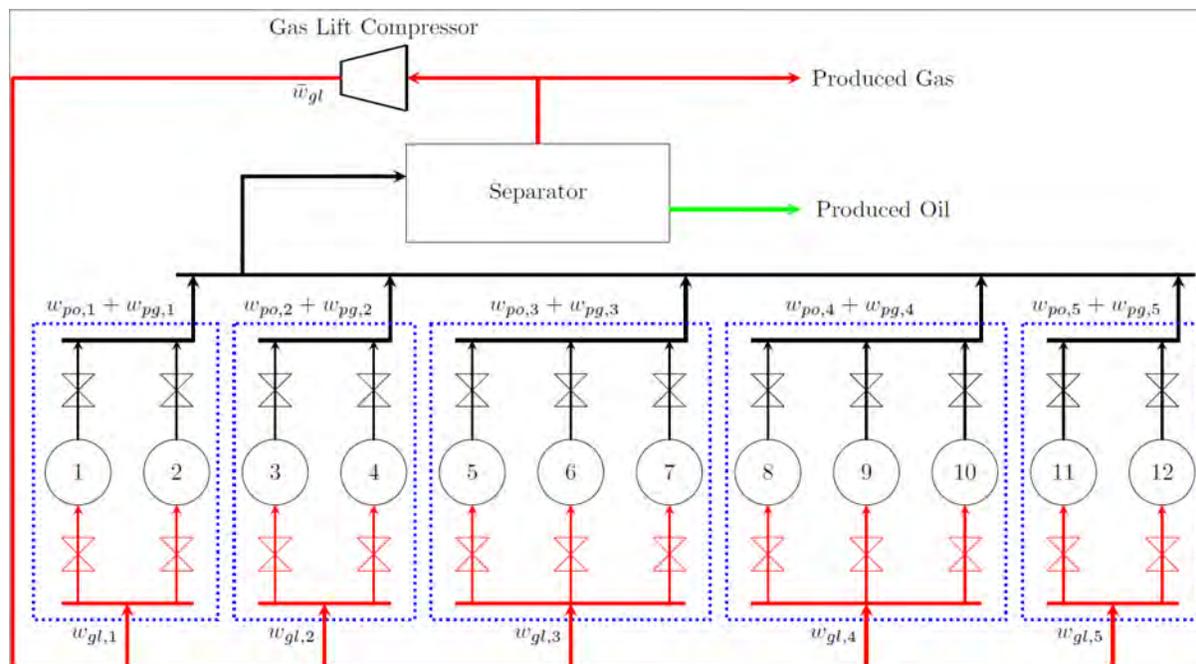


Figure 3—A simplified process diagram of a large-scale offshore field with shared gas-lift resource, equipped with constraints in gas-lift compressor.

Table 2—Subsea Clusters, Oil Production Wells, Neighboring Sets

Subsystem (Cluster)	Oil Production Well ID (wid)	Neighboring Set (N_i)
1	1,2	{2, 3, 4}
2	3,4	{1, 3, 5}
3	5,6,7	{1, 2}
4	8,9,10	{1}
5	11,12	{2}

The produced oil and gas from each subsystem gather in a common topside process facility that has the gas compression station as shown in Fig. 3. In gas lifted wells (see Fig. 4), compressed gas is injected into the well tubing in order to reduce the hydrostatic pressure losses and increase oil production. However, injecting too much compressed gas can increase the frictional pressure drop in the well tubing, which may reduce the oil production. It means that each well has a local optimum associated to maximizing its oil production. Thus, the lift gas supplied from the gas compressor is a common shared resource that must be optimally allocated among the different subsystems. The gas-lift can be part of produced gas or supplied by other common processing facilities.

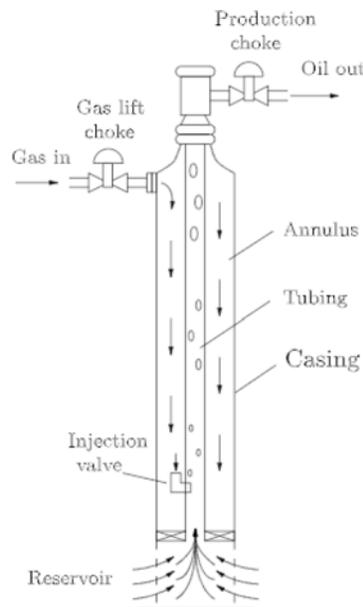


Figure 4—A schematic of a gas lift oil well.

The objective is to maximize the revenue from the oil production from each subsystem and minimize the costs associated with the gas lift compression. The lift gas w_{gl} is a shared resource that must be optimally allocated amongst the subsystems. Thus, the optimization problem for problem (1) is given as follows.

$$\min_{w_{gl,1}, \dots, w_{gl,N}} - \$_o \sum_{i=1}^N w_{po,i}(w_{gl,i}) + \$_{gl} \sum_{i=1}^N w_{gl,i} \quad (14a)$$

subjects to:

$$1) \text{ shared gas - lift constraint: } w_{gl,0} + \sum_{i=1}^N w_{gl,i} - \bar{w}_{gl} = 0 \quad (14b)$$

$$2) \text{ gas - lift oil production well model} \quad (14c)$$

3) physical constraints (14d)

where $\$o$ is the oil price, and $\$gl$ is the cost of gas compression. Total gas lift constraint is labelled by \bar{w}_{gl} . Gas-lift injection rate, $w_{gl,i}$, are the decision variables, $w_{po,i}$ is the oil production rates, which depend on $w_{gl,i}$, and $w_{gl,0}$ is decision variable for virtual subsystem 0 that can be attached individually to any physical subsystem. The local objective function is given by

$$f_i(w_{gl,i}) = -\$o w_{po,i}(w_{gl,i}) + \$gl w_{gl,i}$$

Defining $\bar{w}_{gl,i}^k$ as local shared gas lift constraint for subsystem i at iteration k , the local subproblem can be expressed as follows.

$$P_i(w_{gl,i}^{k+1}(\bar{w}_{gl,i}^k)) = \min_{w_{gl,i}^{k+1}(\bar{w}_{gl,i}^k)} f_i(w_{gl,i}) \quad (15a)$$

subject to:

$$1) \text{ local gas - lift constraint: } w_{gl,i} - \bar{w}_{gl,i}^k \leq 0 \quad (15b)$$

$$2) \text{ local gas - lift oil production well model} \quad (15c)$$

$$3) \text{ local physical constraints} \quad (15d)$$

where $w_{gl,0}^k + \sum_{i=1}^N w_{gl,i}^k - \bar{w}_{gl} = 0$.

The gas-to-oil ratio (GOR), which is a reservoir property, is a time varying disturbance for the different wells. In this simulation study, the GOR are assumed to vary as shown in Fig. 5. High GOR indicates the well that has a lighter fluid requires less amount of gas-lift rate compared to the wells with lower GOR to produce the same amount of oil. GOR is normally local information.

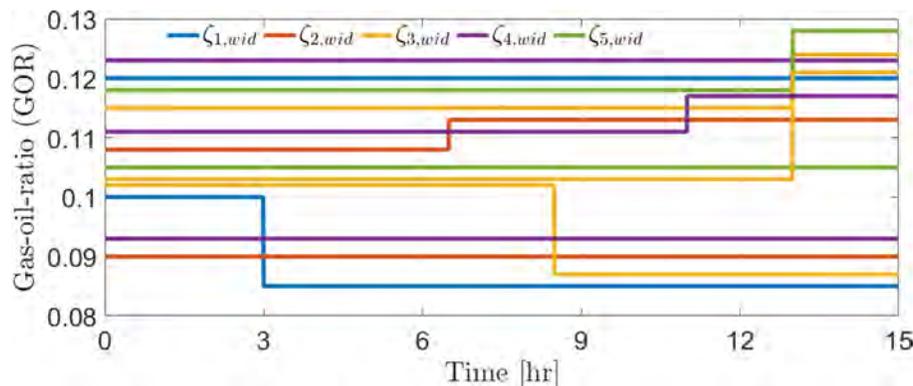


Figure 5—GOR variations. labelled by $\hat{\zeta}_{i,wid}$ of well wid in subsystem i .

Local RTO/DPO problem (15) is solved every 150 sec. We use extended kalman filter (EKF) as the local dynamic estimator to update the model parameters using transient measurements (Krishnamoorthy et al., 2018). Note that the proposed consensus-RTO/DPO framework is not just restricted to this local RTO/DPO approach, and one may instead use any other approach such as dynamic RTO.

In this paper, we use PI controllers for each well for local setpoint, which are tuned using the SIMC tuning rules (Skogestad, 2003). The controller is designed with a sampling time of 1 sec. Each well uses this controller to drive the wellhead pressure, $p_{wh,i,wid}$, to an optimal setpoint computed by the local optimizer (15).

- tNavigator (by Rock Flow Dynamics): integrated static and dynamic modelling from reservoir to surface networks.
- MBAL (by Petex): analytical studies of the reservoir.
- HYSYS (by AspenTech) and K-spice (by Kongsberg): to simulate process dynamic at processing facilities.
- Integrated Production Modeling or IPM (by Petex): to model the integrated production and processing system.

Readers are advised to contact the development company for a more comprehensive description of each software.

Results and Discussions

System-wide optimality

First, we solve the centralized optimization problem (14) to obtain ideal optimal setpoints as the baseline to measure the performance of the consensus-RTO/DPO method. During constraint negotiation, we consider $\delta = 0.005$ and the possible actions combination is only 3 in this case. If the proposed approach can achieve similar near-optimal steady-state condition to the baseline, then the proposal satisfies the goal of this work in terms of system-wide optimal performance.

Fig. 6(a) shows the simulation results comparing the ideal optimum obtained by solving problem (14), and the consensus-RTO. The absolute errors between the ideal optimum and consensus-RTO are shown in Fig. 6(b) and 6(d) for total oil production rate and total gas lift injection rate respectively, which indicate that the proposed method is able to converge to the ideal optimum at steady-state with insignificant absolute error. Moreover, Fig. 6(c) shows that the gas lift rate of each subsystem satisfies the negotiated gas-lift constraints. Note that starting from time 13 hour (due to change of GOR), $\sum_{i=1}^N w_{gl,i} - \bar{w}_{gl} \leq 0$. Therefore, $w_{gl,0} \geq 0$. In addition, Fig. 6(d) also shows that $\delta = 0.005$ is good enough user choice to achieve relatively insignificant absolute error. Step length δ plays crucial role in this method. Too large step length may lead to significant absolute error even though the entire network can converge faster. Too small step length may lead to slower convergence rate for the entire network even though the absolute error can be insignificant. The speed of convergence, in fact, does not solely depend on step length, but also the algebraic connectivity of the graph.

Fig. 7(a) and 7(b) show that the total common cost function differences have reached 0,

$$\sum_{(i,j) \in \mathcal{E}} \Delta \hat{\rho}_{i,j}^k = 0$$

indicating that no gas-lift allocation transfer anymore amongst immediate connected clusters, meaning that the consensus has reached every time consensus-RTO converges to the ideal optimum at steady-state.

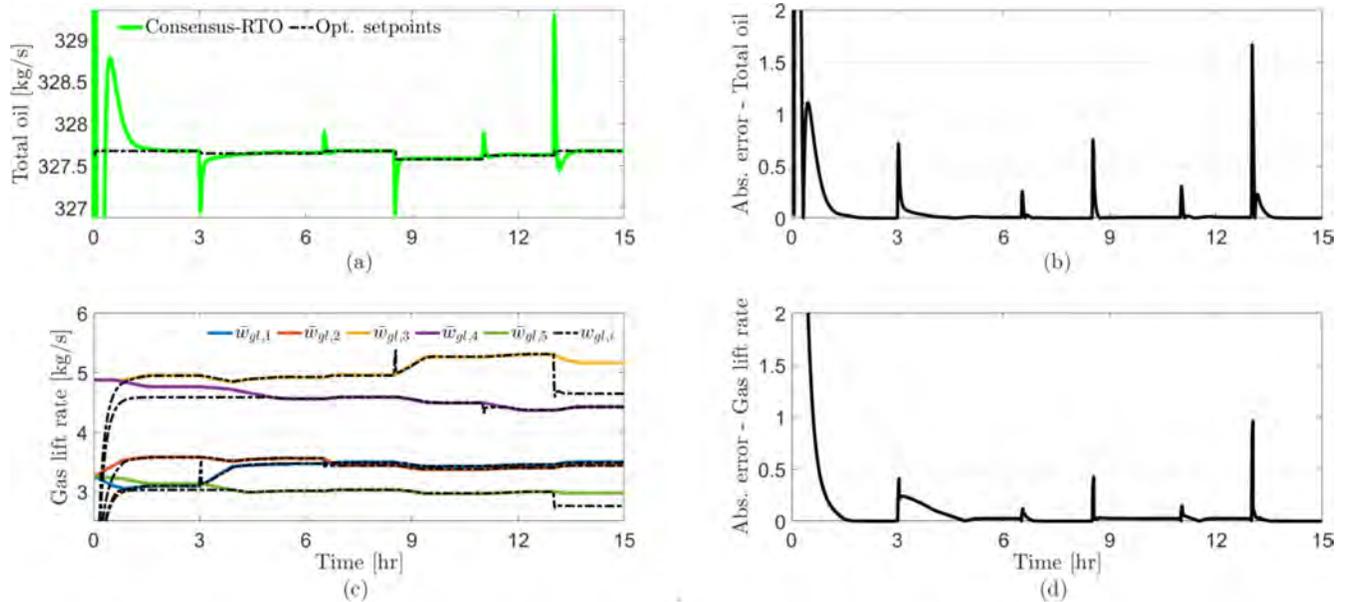


Figure 6—Simulation results showing the performance of the consensus-RTO framework can reach optimal steady-state setpoints, along with the absolute error between them (total produced oil in top-right subplot, and total gas lift rate in bottom-right subplot). The bottom-left subplot shows all negotiated gas-lift constraints and gas-lift rate from each cluster ($N=5$).

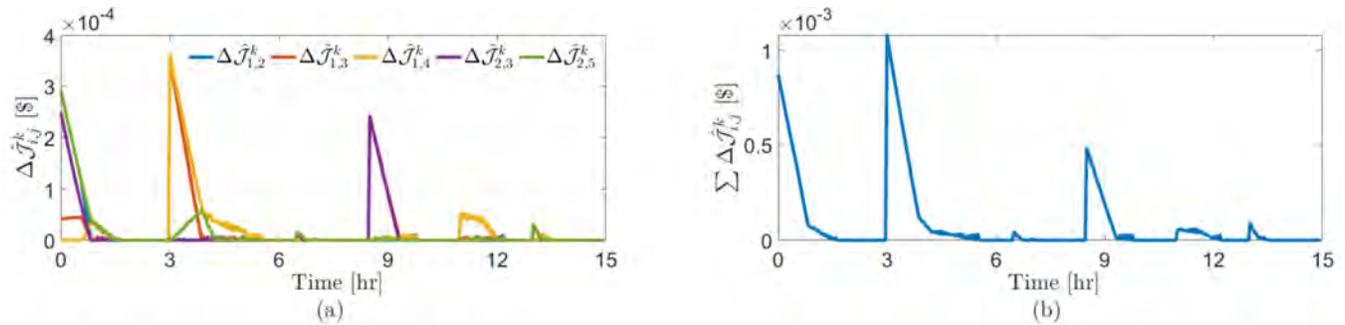


Figure 7—The left subplot shows common cost function differences between two connected subsystems. The right subplot shows total common cost function differences of the entire network, that stay at 0 every time consensus-RTO converges to the ideal optimum at steady-state.

Robustness

The proposed method can achieve system-wide optimal performance if and only if G is a connected graph/network. Losing the rank $L(G)$ due to communication failure creates isolated subsystem(s). This condition leads to profit loss. In this part, we compare the following case:

- Case 1: Consensus-RTO method with G as shown in Fig. 1(b).
- Case 2: Consensus-RTO method with G as shown in Fig. 1(b) and communication failure happens between subsystem 1 and 3 ($v_{1,3} = 0$).
- Case 3: Consensus-RTO method with G as shown in Fig. 1(b) and communication failure happens between subsystem 1 and 3 ($v_{1,3} = 0$), and subsystem 1 and 2 ($v_{1,2} = 0$).
- Case 4: (Centralized-) Distributed RTO method with G as shown in Fig. 1(a) and communication failure happens between central coordinator and subsystem 1 ($v_{0,1} = 0$), and subsystem 4 ($v_{0,4} = 0$).

Note that both case 3 and 4 have subsystem 1 and 4 are isolated from most of the other subsystems i.e., subsystem 2,3 and 5. At time 3-hour, disturbance occurs in subsystem 4.

Fig. 8 shows that Case 2 is still able to reach the same performance as Case 1 because eventhough communication failure happens between subsystem 1 and 3, network G loses no rank in this case. However,

Case 2 takes more time to converge, which is expected. It is also interesting to observe that when other 3 subsystems ‘lose’ subsystem 1, and 4, Case 3 is still able to produce more oil than Case 4. This benefit is obtained because unlike Consensus-RTO, (Centralized-) Distributed RTO framework does not provide communication channel between subsystem 1 and 4. Having communication channel between isolated subsystems allows a new network with another system-wide optimization problem. Thus, in terms of robustness, Consensus-RTO outperforms (Centralized-) Distributed RTO in some cases.

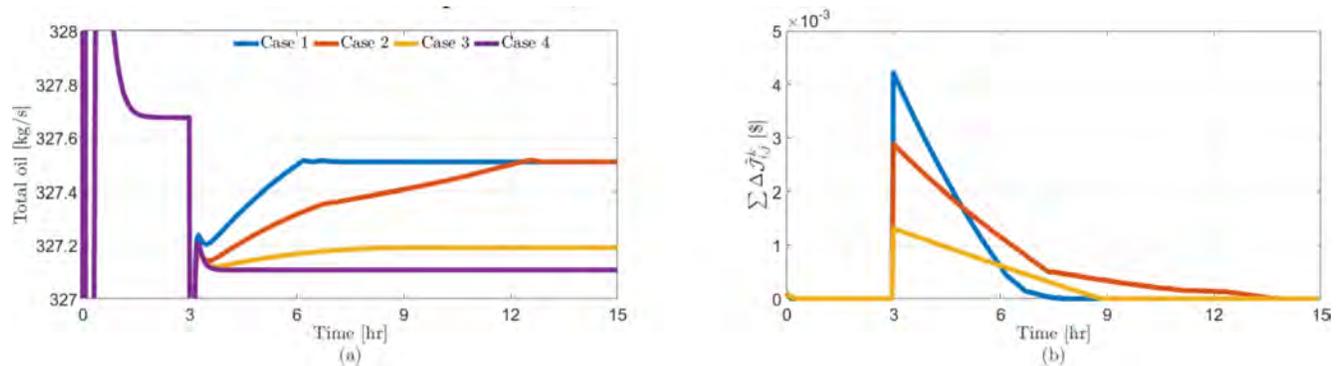


Figure 8—The left subplot shows the production rate in the aforementioned cases. The right subplot shows total common cost function differences of the entire network for case 1,2, and 3.

Speed of Convergence

One interesting open issue in the class of system-wide optimization problem is convergence rate. This is a natural issue since having several subsystems with limited information sharing is fundamentally slowing down the process to find the optimal set point for the entire network. This issue not only occurs in Decentralized RTO but also in Centralized-Distributed RTO. There are several interesting works that have proposed methods to speed up the convergence rate. To name a few, [Wenzel et al., 2020](#) proposed a quadratic approximation for coupling constraints, and [Rahman et al., 2020](#) proposed a gain adaptation to improve the convergence rate.

Conclusions

In this paper, we presented a consensus-RTO/DPO strategy using constraint negotiation for optimal resource allocation with limited information sharing applied to a large-scale subsea oil production system, where we showed that the proposed approach can be an alternative to eliminate the need for central coordinator. The simulation results show that the proposed strategy can converge to optimal steady state setpoint. The consensus-RTO/DPO framework also allows the negotiation among subsystems within connected neighborhood and enables system-wide optimal operation. Moreover, this approach has relatively no additional cost, making it possibly attractive for marginal oil production field. Interesting research directions are first, to improve the optimality of this approach as mathematically the performance of consensus-RTO/DPO is near-optimal performance and depending on user defined step size, and second, to improve the rate of convergence.

Acknowledgement

The authors gratefully acknowledge the financial support from SUBPRO, which is financed by the Research Council of Norway, major industry partners and NTNU.

References

- Aamo, O.M., Eikrem, G.O., Siahaan, H.B., and Foss, B. 2005. "Observer design for multiphase flow in vertical pipes with gas-lift – theory and experiments." *Journal of Process Control* **15**: 247 – 257.

Appendix A

Simplified Model

The model equations can be divided into simple gas-lift system model and simple oil production well model.

Simple gas-lift system model

The mass balance equation is as follows.

$$\dot{m}_{gai} = w_{gl,i} - w_{iv,i} \quad (\text{A.1})$$

where m_{gai} is the mass of gas each well inside the annulus from gas-lift choke valve up to injection valve, $w_{gl,i}$ is gas mass flow rate of each well injected to the annulus, and $w_{iv,i}$ is gas mass flow rate of each injected to well tubing.

The mass flow rate equation is as follows.

$$w_{iv,i} = c_{iv,i} \sqrt{\rho_{ai,i} \max\{0, (p_{ai,i} - p_{wi,i})\}} \quad (\text{A.2})$$

where $C_{iv,i}$ is the valve coefficient of injection valve, $\rho_{ai,i}$ is gas mass density in annulus of each well, $p_{ai,i}$ is gas pressure inside annulus, and $p_{wi,i}$ is injection pressure inside well tubing.

The gas density is defined as follows.

$$\rho_{ai,i} = \frac{M_w}{RT_{aj}} p_{ai,i} \quad (\text{A.3})$$

where M_w is the molecular weight, R is Reynold number, and T_{aj} is the temperature of the gas in annulus in each well.

The pressure is defined as follows.

$$p_{ai,i} = \left(\frac{RT_{aj}}{M_w} + gH_{aj} \right) \frac{m_{gai}}{\pi (r_{ai,i}^2 - r_{wi,i}^2) L_{aj}} \quad (\text{A.4})$$

where $r_{ai,i} = \frac{D_{aj}}{2}$ and $r_{wi,i} = \frac{D_{wi,i}}{2}$, D_{aj} and $D_{wi,i}$ are the diameter of gas-lift supply and annulus respectively, L_{aj} is the length of annulus, g is the gravity force, and H_{aj} is the height of the annulus.

Simple oil production well model

The mass balance equations are as follows.

$$\dot{m}_{gt,i} = w_{rg,i} + w_{iv,i} - w_{pg,i} \quad (\text{A.5a})$$

$$\dot{m}_{ot,i} = w_{ro,i} - w_{po,i} \quad (\text{A.5b})$$

where $m_{gt,i}$ is the mass of gas inside well tubing, $m_{ot,i}$ is the mass of oil inside well tubing, $w_{rg,i}$ is gas mass flow rate from reservoir, $w_{ro,i}$ is oil mass flow rate from reservoir, $w_{pg,i}$ is produced gas mass flow rate, and $w_{po,i}$ is produced oil mass flow rate.

The mass flow rate equations are as follows.

$$w_{pc,i} = c_{pc,i} \sqrt{\rho_{mix,i} \max\{0, (p_{wh,i} - p_m)\}} \quad (\text{A.6a})$$

$$w_{rg,i} = GOR_i w_{ro,i} \quad (\text{A.6b})$$

$$w_{pg,i} = \frac{m_{gt,i}}{m_{gt,i} + m_{ot,i}} w_{pc,i} \quad (\text{A.6c})$$

$$w_{ro,i} = PI_i (p_{res,i} - p_{bh,i}) \quad (\text{A.6d})$$

$$w_{po,i} = \frac{m_{ot,i}}{m_{gt,i} + m_{ot,i}} \cdot w_{pc,i} \quad (\text{A.6e})$$

where $w_{pc,i}$ is the produced hydrocarbon (both oil and gas) mass flow rate, $c_{pc,i}$ is the valve coefficient of production choke valve, $\rho_{mix,i}$ is mixed mass density in the well tubing, $p_{res,i}$ is reservoir pressure of each well, $p_{bh,i}$ is bottomhole pressure of each well, $p_{wh,i}$ is wellhead pressure of each well, p_m is manifold pressure, GOR_i is gas-to-oil ratio of each well, and PI_i is well productivity index.

The mass density is defined as follows.

$$\rho_{mix,i} = \frac{m_{gt,i} + m_{ot,i} - \rho_{o,i} \pi r_{bh,i}^2 L_{bh,i}}{\pi r_{wi,i}^2 L_{w,i}} \quad (\text{A.7})$$

where $\rho_{o,i}$ is oil density, $r_{bh,i} = \frac{D_{bh,i}}{2}$, and $D_{bh,i}$ is the diameter of bottomhole, and $L_{bh,i}$ and $L_{w,i}$ are the length of bottomhole, and well tubing respectively.

The pressures are defined as follows.

$$p_{wh,i} = \frac{R T_{w,i}}{M_w} \frac{m_{gt,i}}{\pi r_{wi,i}^2 L_{w,i} + \pi r_{bh,i}^2 L_{bh,i} - \frac{m_{ot,i}}{\rho_{o,i}}} - \Delta p_{fric,w,i} \quad (\text{A.8a})$$

$$p_{wi,i} = p_{wh,i} + \rho_{mix,i} g H_{w,i} \quad (\text{A.8b})$$

$$p_{bh,i} = p_{wi,i} + \rho_{o,i} g H_{bh,i} \quad (\text{A.8c})$$

where $T_{w,i}$ is the temperature in well tubing, $\Delta p_{fric,w,i}$ is pressure drop due to friction in well tubing, $H_{w,i}$ is the height of well tubing from well head to injection point, and $H_{bh,i}$ is the height from injection point to the bottomhole.

As seen from (A.1) - (A.8), the process model can be represented as a semi-explicit DAE system. This model was also compared with high fidelity dynamic simulator such as OLGa which ensures that the models used are representative of the real system (Codas et al., 2016).

Some important assumptions that we consider are that the oil density $\rho_{o,i}$ is relatively constant, the mass of oil computed for mixture density excludes the mass of oil in the bottomhole, and we assume that the impact of water produced is relatively insignificant. Similar simplified modelings can also be found in Aamo et al., 2005, Eikrem et al., 2008, and Krishnamoorthy et al., 2018, to name a few.