United States Patent Application Publication 20250264849
Kind Code A1
Publication Date August 21, 2025
Inventor(s) Lu; Joseph

SYSTEMS AND METHODS FOR MANAGING AN INDUSTRIAL PROCESS IN A PROCESS PLANT

Abstract

Various embodiments described herein relate to management of an industrial process in a process plant. In this regard, a status of an order of one or more orders is determined based at least on component spent characteristics. The one or more orders are associated with production of an industrial process product by the industrial process. A flag is set to a first value if the status of the order is a first status. Upon receiving an optimization request to optimize the industrial process with respect to another order, first notification(s) are rendered on a display. Then, the flag is set to a second value if the status of the order is a second status. Further, second notifications are rendered on the display, demand data associated with the orders is updated, and control signal is transmitted to a controller to optimize the industrial process with respect to another order.

Inventors: **Lu; Joseph (Glendale, AZ)**

Applicant: **HONEYWELL INTERNATIONAL INC.** (Charlotte, NC)

Family ID: **1000007953751**

Appl. No.: **18/444774**

Filed: **February 19, 2024**

Publication Classification

Int. Cl.: **G05B13/04** (20060101)

U.S. Cl.:

CPC **G05B13/041** (20130101); **G05B13/048** (20130101);

Background/Summary

TECHNICAL FIELD

[0001] The present disclosure relates generally to industrial process control systems, and more particularly to systems and methods for managing industrial processes in a process plant.

BACKGROUND

[0002] Processing facilities or plants are often managed using industrial process control systems. A typical processing facility may be composed of multiple operations that may operate independently or in combination. For example, an oil and gas refining system may include a continuous process section and a batch process section. The continuous process section of the refinery is configured to generate one or more blending components that are further utilized by the batch process section to produce refined products. To this end, the batch process section may be concatenated to the continuous process section of the refinery. In some cases, the batch process section may disrupt the continuous process section of the refinery. Also, in some cases, situations such as asset downtime, ad hoc orders or requests to produce refined products, industrial process control system downtime, maintenance operations, and/or the like in the facility may disrupt for example, operations of the continuous process section and the batch process section. In this regard, initialization and planning of operations in the facility becomes challenging.

SUMMARY

[0003] The details of some embodiments of the subject matter described in this specification are set forth in the accompanying drawings and the description below. Other features, aspects, and advantages of the subject matter will become apparent from the description, the drawings, and the claims.

[0004] In accordance with an embodiment of the current disclosure, a method for managing an industrial process in a process plant is described herein. The method comprises determining a status of a first order of one or more orders based at least on component spent characteristics such that the first order is associated with production of an industrial process product by the industrial process. The method further comprises setting a flag to a first value if the status of the first order corresponds to a first status. The flag indicates the status of an order of the one or more orders. Further, the method comprises receiving an optimization request to optimize the industrial process with respect to a second order of the one or more orders such that the second order is associated with production of the industrial process product by the industrial process. In response to receiving the optimization request, the method comprises rendering on a display one or more first notifications based on the first value. Furthermore, the method comprises setting the flag to a second value if the status of the first order corresponds to a second status. In response to setting the flag to the second value, the method comprises rendering on the display, one or more second notifications based on the second value, updating demand data associated with the one or more orders based on the component spent characteristics and the optimization request, and transmitting a control signal based on the demand data to a controller configured to optimize the industrial process with respect to the second order.

[0005] In accordance with another embodiment of the current disclosure, a system for managing an industrial process in a process plant is described herein. The system comprises a processor and a memory communicatively coupled to the processor, wherein the memory comprises one or more instructions which when executed by the processor, cause the processor to determine a status of a first order of one or more orders based at least on component spent characteristics such that the first order is associated with production of an industrial process product by the industrial process. The processor is further configured to set a flag to a first value if the status of the first order corresponds to a first status. The flag indicates the status of an order of the one or more orders. The processor is also further configured to receive an optimization request to optimize the industrial process with respect to a second order of the one or more orders such that the second order is associated with production of the industrial process product by the industrial process. In response to receiving the optimization request, the processor is configured to render on a display one or more first notifications based on the first value. Then, the flag is set to a second value if the status of the first order corresponds to a second status. In response to setting the flag to the second value, the processor is configured to render on the display, one or more second notifications based on the second value, update demand data associated with the one or more orders based on the component spent characteristics and the optimization request, and transmit a control signal based on the demand data to a controller configured to optimize the industrial process with respect to the second order.

[0006] In accordance with yet another embodiment of the current disclosure, a non-transitory, computer-readable storage medium having instructions stored thereon and executable by one or more processors is described herein. In this regard, the instructions when executed by one or more processors cause the one or more processors to determine a status of a first order of one or more orders based at least on component spent characteristics such that the first order is associated with production of an industrial process product by the industrial process. Further, a flag is set to a first value if the status of the first order corresponds to a first status. The flag indicates the status of an order of the one or more orders. Then, the one or more processors receive an optimization request to optimize the industrial process with respect to a second order of the one or more orders such that the second order is associated with production of the industrial process product by the industrial process. In response to receiving the optimization request, the one or more processors render on a display one or more first notifications based on the first value. The one or more processors set the flag to a second value if the status of the first order corresponds to a second status. In response to setting the flag to the second value, the one or more processors render on the display, one or more second notifications based on the second value, update demand data associated with the one or more orders based on the component spent characteristics and the optimization request, and transmit a control signal based on the demand data to a controller configured to optimize the industrial process with respect to the second order.

[0007] The above summary is provided merely for purposes of providing an overview of one or more exemplary embodiments described herein so as to provide a basic understanding of some aspects of the disclosure. Accordingly, it will be appreciated that the above-described embodiments are merely examples and should not be construed to narrow the scope or spirit of the disclosure in any way. It will be appreciated that the scope of the disclosure encompasses many potential embodiments in addition to those here summarized, some of which are further explained in the following description and its accompanying drawings.

[0008] Additional objects and advantages of the disclosed embodiments will be set forth in part in the description that follows, and in part will be apparent from the description, or may be learned by practice of the disclosed embodiments. The objects and advantages of the disclosed embodiments will be realized and attained by means of the elements and combinations particularly pointed out in the appended claims.

[0009] It is to be understood that both the foregoing general description and the following detailed description are exemplary and explanatory only and are not restrictive of the disclosed embodiments, as claimed.

Description

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] The description of the illustrative embodiments can be read in conjunction with the accompanying figures. It will be appreciated that for simplicity and clarity of illustration, elements illustrated in the figures have not necessarily been drawn to scale. For example, the dimensions of some of the elements are exaggerated relative to other elements. Embodiments incorporating teachings of the present disclosure are shown and described with respect to the figures presented herein, in which:

[0011] FIG. **1** illustrates an exemplary networked computing system environment, in accordance with one or more embodiments described herein;

[0012] FIG. **2** illustrates a schematic block diagram of a framework of an IoT platform of the networked computing system, in accordance with one or more embodiments described herein;

[0013] FIG. **3**A illustrates a planning model, in accordance with one or more embodiments described herein;

[0014] FIG. **3**B illustrates a model predictive control (MPC) model, in accordance with one or more embodiments described herein;

[0015] FIG. **4** illustrates a system that provides management of industrial process optimization related to batch operations, in accordance with one or more embodiments described herein;

[0016] FIG. **5** illustrates a system that provides a cascaded MPC architecture for plantwide control and optimization, in accordance with one or more embodiments described herein;

[0017] FIG. **6** illustrates an industrial process, in accordance with one or more embodiments described herein;

[0018] FIG. **7** illustrates a system that provides an exemplary industrial process computer system, in accordance with one or more embodiments described herein;

[0019] FIG. **7**A illustrates a system for controlling an industrial process, in accordance with one or more embodiments described herein;

[0020] FIG. **7**B illustrates a system for controlling an industrial process, in accordance with one or more embodiments described herein;

[0021] FIG. **7**C illustrates a user interface rendering demand data, in accordance with one or more embodiments described herein;

[0022] FIG. **7**D illustrates an initialization plan for initializing an asynchronous update scheme, in accordance with one or more embodiments described herein;

[0023] FIG. **8**A illustrates a graphical representation that depicts an initial industrial demand vector for m-number of days, in accordance with one or more embodiments described herein;

[0024] FIG. **8**B illustrates a graphical representation that depicts an updated demand vector, in accordance with one or more embodiments described herein;

[0025] FIG. **9** illustrates a system that provides another cascaded MPC architecture for enterprise-wide control and optimization, in accordance with one or more embodiments described herein;

[0026] FIG. **10** illustrates a system that provides an enterprise supply chain with transport links, in accordance with one or more embodiments described herein;

[0027] FIG. **11** illustrates a flow diagram for managing industrial process optimization related to batch operations, in accordance with one or more embodiments described herein;

[0028] FIG. **12** illustrates a flow diagram for a method for controlling an industrial process, in accordance with one or more embodiments described herein;

[0029] FIG. **13** illustrates a flow diagram for managing an industrial process in a process plant, in accordance with one or more embodiments described herein; and

[0030] FIG. **14** illustrates a functional block diagram of a computer that may be configured to execute techniques described in accordance with one or more embodiments described herein.

DETAILED DESCRIPTION

[0031] Reference will now be made in detail to embodiments, examples of which are illustrated in the accompanying drawings. In the following detailed description, numerous specific details are set forth in order to provide a thorough understanding of the various described embodiments. However, it will be apparent to one of ordinary skill in the art that the various described embodiments may be practiced without these specific details. In other instances, well-known methods, procedures, components, circuits, and networks have not been described in detail so as not to unnecessarily obscure aspects of the embodiments. The term "or" is used herein in both the alternative and conjunctive sense, unless otherwise indicated. The terms “illustrative,” “example,” and “exemplary” are used to be examples with no indication of quality level. Like numbers refer to like elements throughout.

[0032] The phrases “in an embodiment,” “in one embodiment,” “according to one embodiment,” and the like generally mean that the particular feature, structure, or characteristic following the phrase can be included in at least one embodiment of the present disclosure and can be included in more than one embodiment of the present disclosure (importantly, such phrases do not necessarily refer to the same embodiment).

[0033] The word “exemplary” is used herein to mean “serving as an example, instance, or illustration.” Any implementation described herein as “exemplary” is not necessarily to be construed as preferred or advantageous over other implementations.

[0034] If the specification states a component or feature “can,” “may,” “could,” “should,” “would,” “preferably,” “possibly,” “typically,” “optionally,” “for example,” “often,” or “might” (or other such language) be included or have a characteristic, that particular component or feature is not required to be included or to have the characteristic. Such component or feature can be optionally included in some embodiments, or it can be excluded.

[0035] In one or more embodiments, the present disclosure provides for an “Internet-of-Things” or “IoT” platform for industrial process control and/or industrial process optimization that uses real-time accurate models and/or real-time control for sustained peak performance of an enterprise and/or an enterprise process. The IoT platform is an extensible platform that is portable for deployment in any cloud or data center environment for providing an enterprise-wide, top to bottom control of processes and/or assets. Further, the IoT platform of the present disclosure supports end-to-end capability to execute models against process data and/or to translate the output into actionable insights and/or real-time control, as detailed in the following description.

[0036] Processing facilities are often managed using industrial process control systems. For certain types of industrial processes such as, for example, oil and gas processes, a batch operation can be performed in addition to one or more other industrial processes (e.g., one or more other continuous industrial processes). In one example, a batch operation can blend different blending components to form a refinery product. However, a batch operation can disrupt one or more other industrial processes and/or one or more optimization operations related to a processing facility. For example, a batch operation can be configured with an on-and-off functionality associated with a starting time and/or duration that is generally not predictable. As such, a batch operation can cause one or more other industrial processes and/or one or more optimization operations related to a processing facility to be performed in an inefficient manner (e.g., a cyclic manner) with respect to operation of the processing facility. Also, at some instances, situations such as asset downtime, ad hoc orders or requests to produce refined products, orders unsynchronized with demand, downtime of operational technology (OT) systems or information technology (IT) applications, maintenance operations, and/or the like may also disrupt the continuous and batch operations in the processing facility. In this regard, data associated with the operations may be absent or it may be required to collect the data afresh leading to disruptions in the operations. It is therefore advantageous to optimize a batch operation with respect to one or more other industrial processes and/or one or more optimization operations related to a processing facility in order to achieve optimal performance and/or optimal efficiency for one or more industrial processes.

[0037] Thus, to address these and/or other issues, management of an industrial process related to continuous and batch operations in a processing facility is provided. In one or more embodiments, harmonization between plantwide optimization and one or more batch operations is provided. In one or more embodiments, optimization between one or more continuous operations and one or more batch operations in a hybrid plant or processing facility is provided. In various embodiments, management of industrial process optimization provides one or more optimal recipes (e.g., optimal batch blending recipes, optimal batch-blending plantwide recipes, etc.) for the industrial process. In one or more embodiments, statuses of one or more operations in the processing facility is monitored and tracked so that status of each operation is known even during situations such as downtime, maintenance operations, etc., in the processing facility. This facilitates optimal planning of operations in the processing facility without interruptions in order to achieve optimal performance and/or optimal efficiency for one or more industrial processes. This also aids synchronization of the continuous and batch operations in the processing facility seamlessly. In certain embodiments, the industrial process includes batch product blending for an oil refinery. In certain embodiments, the industrial process includes filter washing processes in a chemical refinery (e.g., an alumina refinery, etc.) or a lubricant plant. However, it is to be appreciated that, in certain embodiments, the industrial process is a different type of industrial process.

[0038] In various embodiments, a virtual content tracker is strategically inserted into a continuous control process or an optimization process (e.g., a plant optimizer) for an industrial process or industrial facility to monitor spent inventory for a batch process. In one or more embodiments, the virtual content tracker is a discontinuous system configured for monitoring spent inventory related to a batch process. In one or more embodiments, the virtual content tracker is employed with an asynchronous update scheme that is configured to determine how batch operations or batch productions should be updated when a continuous optimizer (e.g., a plantwide optimizer) receives new input at a pre-specified frequency, or at a user request. Also, in various embodiments, the virtual content tracker is configured to monitor and track status of each operation so that the status of each operation is made available. Further, in various embodiments described herein, initialization of the asynchronous update scheme employed in the virtual content tracker and working of the plantwide optimizer is based on the status of each operation so that the asynchronous update scheme and the plantwide optimizer operate without any bumps or interruptions.

[0039] As such, by employing one or more techniques disclosed herein, batch disruptions with respect to an industrial process is minimized or eliminated (e.g., as if the batch operations had been converted into continuous operations for the industrial process). Also, by employing one or more techniques disclosed herein, interruptions associated with: initialization of the asynchronous update scheme and working of the plantwide optimizer can be eliminated though undesired situations occur in the processing facility. Additionally, by employing one or more techniques disclosed herein, smooth plantwide control and/or optimization solutions related to the batch operations are provided (e.g., zig-zagged optimization solutions related to the batch operations are minimized or eliminated). Moreover, by employing one or more techniques disclosed herein, industrial process performance and/or industrial process efficiency is optimized. In various embodiments, an amount of time and/or an amount of processing related to an industrial process is reduced. Additionally, in one or more embodiments, performance of a processing system (e.g., a control system) associated with an industrial process is improved by employing one or more techniques disclosed herein. For example, in one or more embodiments, a number of computing resources, a number of a storage requirements, and/or number of errors associated with a processing system (e.g., a control system) for an industrial process is reduced by employing one or more techniques disclosed herein.

[0040] FIG. **1** illustrates an exemplary networked computing system environment **100**, according to the present disclosure. As shown in FIG. **1**, networked computing system environment **100** is organized into a plurality of layers including a cloud layer **105**, a network layer **110**, and an edge layer **115**. As detailed further below, components of the edge **115** are in communication with components of the cloud **105** via network **110**.

[0041] In various embodiments, network **110** is any suitable network or combination of networks and supports any appropriate protocol suitable for communication of data to and from components of the cloud **105** and between various other components in the networked computing system environment **100** (e.g., components of the edge **115**). According to various embodiments, network **110** includes a public network (e.g., the Internet), a private network (e.g., a network within an organization), or a combination of public and/or private networks. According to various embodiments, network **110** is configured to provide communication between various components depicted in FIG. **1**. According to various embodiments, network **110** comprises one or more networks that connect devices and/or components in the network layout to allow communication between the devices and/or components. For example, in one or more embodiments, the network **110** is implemented as the Internet, a wireless network, a wired network (e.g., Ethernet), a local area network (LAN), a Wide Area Network (WANs), Bluetooth, Near Field Communication (NFC), or any other type of network that provides communications between one or more components of the network layout. In some embodiments, network **110** is implemented using cellular networks, satellite, licensed radio, or a combination of cellular, satellite, licensed radio, and/or unlicensed radio networks.

[0042] Components of the cloud **105** include one or more computer systems **120** that form a so-called "Internet-of-Things" or "IoT" platform **125**. It should be appreciated that "IoT platform" is an optional term describing a platform connecting any type of Internet-connected device and should not be construed as limiting on the types of computing systems useable within IoT platform **125**. In particular, in various embodiments, computer systems **120** includes any type or quantity of one or more processors and one or more data storage devices comprising memory for storing and executing applications or software modules of networked computing system environment **100**. In one embodiment, the processors and data storage devices are embodied in server-class hardware, such as enterprise-level servers. For example, in an embodiment, the processors and data storage devices comprise any type or combination of application servers, communication servers, web servers, super-computing servers, database servers, file servers, mail servers, proxy servers, and/virtual servers. Further, the one or more processors are configured to access the memory and execute processor-readable instructions, which when executed by the processors configures the processors to perform a plurality of functions of the networked computing system environment **100**.

[0043] Computer systems **120** further include one or more software components of the IoT platform **125**. For example, in one or more embodiments, the software components of computer systems **120** include one or more software modules to communicate with user devices and/or other computing devices through network **110**. For example, in one or more embodiments, the software components include one or more modules **141**, models **142**, engines **143**, databases **144**, services **145**, and/or applications **146**, which may be stored in/by the computer systems **120** (e.g., stored on the memory), as detailed with respect to FIG. **2** below. According to various embodiments, the one or more processors are configured to utilize the one or more modules **141**, models **142**, engines **143**, databases **144**, services **145**, and/or applications **146** when performing various methods described in this disclosure.

[0044] Accordingly, in one or more embodiments, computer systems **120** execute a cloud computing platform (e.g., IoT platform **125**) with scalable resources for computation and/or data storage and may run one or more applications on the cloud computing platform to perform various computer-implemented methods described in this disclosure. In some embodiments, some of the modules **141**, models **142**, engines **143**, databases **144**, services **145**, and/or applications **146** are combined to form fewer modules, models, engines, databases, services, and/or applications. In some embodiments, some of the modules **141**, models **142**, engines **143**, databases **144**, services **145**, and/or applications **146** are separated into separate, more numerous modules, models, engines, databases, services, and/or applications. In some embodiments, some of the modules **141**, models **142**, engines **143**, databases **144**, services **145**, and/or applications **146** are removed while others are added.

[0045] The computer systems **120** are configured to receive data from other components (e.g., components of the edge **115**) of networked computing system environment **100** via network **110**. Computer systems **120** are further configured to utilize the received data to produce a result.

According to various embodiments, information indicating the result is transmitted to users via user computing devices over network **110**. In some embodiments, the computer systems **120** is a server system that provides one or more services including providing the information indicating the received data and/or the result(s) to the users. According to various embodiments, computer systems **120** are part of an entity which include any type of company, organization, or institution that implements one or more IoT services. In some examples, the entity is an IoT platform provider.

[0046] Components of the edge **115** include one or more enterprises **160***a*-**160***n* each including one or more edge devices **161***a*-**161***n* and one or more edge gateways **162***a*-**162***n*. For example, a first enterprise **160***a* includes first edge devices **161***a* and first edge gateways **162***a*, a second enterprise **160***b* includes second edge devices **161***b* and second edge gateways **162***b*, and an nth enterprise **160***n* includes nth edge devices **161***n* and nth edge gateways **162***n*. As used herein, enterprises **160***a*-**160***n* represent any type of entity, facility, or vehicle, such as, for example, processing facilities, industrial plants, oil and gas facilities (e.g., oil refineries), chemical processing facilities (e.g., metal refinery, alumina refinery, etc.), lubricant industrial plants, manufacturing plants, buildings, warehouses, real estate facilities, laboratories, companies, divisions, aircrafts, spacecrafts, automobiles, ships, boats, military vehicles, or any other type of entity, facility, and/or vehicle that includes any number of local devices.

[0047] According to various embodiments, the edge devices **161***a*-**161***n* represent any of a variety of different types of devices that may be found within the enterprises **160***a*-**160***n*. Edge devices **161***a*-**161***n* are any type of device configured to access network **110**, or be accessed by other devices through network **110**, such as via an edge gateway **162***a*-**162***n*. According to various embodiments, edge devices **161***a*-**161***n* are “IoT devices” which include any type of network-connected (e.g., Internet-connected) device. For example, in one or more embodiments, the edge devices **161***a*-**161***n* include sensors, units, storage tanks, air handler units, fans, actuators, valves, pumps, ducts, processors, computers, vehicle components, cameras, displays, doors, windows, security components, HVAC components, factory equipment, refinery equipment, and/or any other devices that are connected to the network **110** for collecting, sending, and/or receiving information. Each edge device **161***a*-**161***n* includes, or is otherwise in communication with, one or more controllers for selectively controlling a respective edge device **161***a*-**161***n* and/or for sending/receiving information between the edge devices **161***a*-**161***n* and the cloud **105** via network **110**. With reference to FIG. **2**, in one or more embodiments, the edge **115** include operational technology (OT) systems **163***a*-**163***n* and information technology (IT) applications **164***a*-**164***n* of each enterprise **161***a*-**161***n*. The OT systems **163***a*-**163***n* include hardware and software for detecting and/or causing a change, through the direct monitoring and/or control of industrial equipment (e.g., edge devices **161***a*-**161***n*), assets, processes, and/or events. The IT applications **164***a*-**164***n* includes network, storage, and computing resources for the generation, management, storage, and delivery of data throughout and between organizations.

[0048] The edge gateways **162***a*-**162***n* include devices for facilitating communication between the edge devices **161***a*-**161***n* and the cloud **105** via network **110**. For example, the edge gateways **162***a*-**162***n* include one or more communication interfaces for communicating with the edge devices **161***a*-**161***n* and for communicating with the cloud **105** via network **110**. According to various embodiments, the communication interfaces of the edge gateways **162***a*-**162***n* include one or more cellular radios, Bluetooth, WiFi, near-field communication radios, Ethernet, or other appropriate communication devices for transmitting and receiving information. According to various embodiments, multiple communication interfaces are included in each gateway **162***a*-**162***n* for providing multiple forms of communication between the edge devices **161***a*-**161***n*, the gateways **162***a*-**162***n*, and the cloud **105** via network **110**. For example, in one or more embodiments, communication are achieved with the edge devices **161***a*-**161***n* and/or the network **110** through wireless communication (e.g., WiFi, radio communication, etc.) and/or a wired data connection (e.g., a universal serial bus, an onboard diagnostic system, etc.) or other communication modes, such as a local area network (LAN), wide area network (WAN) such as the Internet, a telecommunications network, a data network, or any other type of network.

[0049] According to various embodiments, the edge gateways **162**_a_-**162**_n_ also include a processor and memory for storing and executing program instructions to facilitate data processing. For example, in one or more embodiments, the edge gateways **162**_a_-**162**_n_ are configured to receive data from the edge devices **161**_a_-**161**_n_ and process the data prior to sending the data to the cloud **105**. Accordingly, in one or more embodiments, the edge gateways **162**_a_-**162**_n_ include one or more software modules or components for providing data processing services and/or other services or methods of the present disclosure. With reference to FIG. **2**, each edge gateway **162**_a_-**162**_n_ includes edge services **165**_a_-**165**_n_ and edge connectors **166**_a_-**166**_n_. According to various embodiments, the edge services **165**_a_-**165**_n_ include hardware and software components for processing the data from the edge devices **161**_a_-**161**_n_. According to various embodiments, the edge connectors **166**_a_-**166**_n_ include hardware and software components for facilitating communication between the edge gateway **162**_a_-**162**_n_ and the cloud **105** via network **110**, as detailed above. In some cases, any of edge devices **161**_a-n_, edge connectors **166**_a-n_, and edge gateways **162**_a-n_ have their functionality combined, omitted, or separated into any combination of devices. In other words, an edge device and its connector and gateway need not necessarily be discrete devices.

[0050] FIG. **2** illustrates a schematic block diagram of framework **200** of the IoT platform **125**, according to the present disclosure. The IoT platform **125** of the present disclosure is a platform for plantwide optimization that uses real-time accurate models and/or real-time data to deliver intelligent actionable recommendations and/or real-time control for sustained peak performance of the enterprise **160**_a_-**160**_n_. The IoT platform **125** is an extensible platform that is portable for deployment in any cloud or data center environment for providing an enterprise-wide, top to bottom view, displaying the status of processes, assets, people, and safety. Further, the IoT platform **125** supports end-to-end capability to execute digital twins against process data and to translate the output into actionable insights, using the framework **200**, detailed further below.

[0051] As shown in FIG. **2**, the framework **200** of the IoT platform **125** comprises a number of layers including, for example, an IoT layer **205**, an enterprise integration layer **210**, a data pipeline layer **215**, a data insight layer **220**, an application services layer **225**, and an applications layer **230**. The IoT platform **125** also includes a core services layer **235** and an extensible object model (EOM) **250** comprising one or more knowledge graphs **251**. The layers **205-235** further include various software components that together form each layer **205-235**. For example, in one or more embodiments, each layer **205-235** includes one or more of the modules **141**, models **142**, engines **143**, databases **144**, services **145**, applications **146**, or combinations thereof. In some embodiments, the layers **205-235** are combined to form fewer layers. In some embodiments, some of the layers **205-235** are separated into separate, more numerous layers. In some embodiments, some of the layers **205-235** are removed while others may be added.

[0052] The IoT platform **125** is a model-driven architecture. Thus, in certain embodiments, the extensible object model **250** communicates with each layer **205-230** to contextualize site data of the enterprise **160**_a_-**160**_n_ using an extensible object model (or “asset model”) and knowledge graphs **251** where the equipment (e.g., edge devices **161**_a_-**161**_n_) and processes of the enterprise **160**_a_-**160**_n_ are modeled. The knowledge graphs **251** of EOM **250** are configured to store the models in a central location. The knowledge graphs **251** define a collection of nodes and links that describe real-world connections that enable smart systems. As used herein, a knowledge graph **251**: (i) describes real-world entities (e.g., edge devices **161**_a_-**161**_n_) and their interrelations organized in a graphical interface; (ii) defines possible classes and relations of entities in a schema; (iii) enables interrelating arbitrary entities with each other; and (iv) covers various topical domains. In other words, the knowledge graphs **251** define large networks of entities (e.g., edge devices **161**_a_-**161**_n_), semantic types of the entities, properties of the entities, and relationships between the entities. Thus, the knowledge graphs **251** describe a network of “things” that are relevant to a specific domain or to an enterprise or organization. Knowledge graphs **251** are not limited to abstract concepts and relations, but can also contain instances of objects, such as, for example, documents and datasets. In some embodiments, the knowledge graphs **251** include resource description framework (RDF) graphs. As used herein, a "RDF graph" is a graph data model that formally describes the semantics, or meaning, of information. The RDF graph also represents metadata (e.g., data that describes data). According to various embodiments, knowledge graphs **251** also include a semantic object model. The semantic object model is a subset of a knowledge graph **251** that defines semantics for the knowledge graph **251**. For example, the semantic object model defines the schema for the knowledge graph **251**.

[0053] As used herein, EOM **250** is a collection of application programming interfaces (APIs) that enables seeded semantic object models to be extended. For example, the EOM **250** of the present disclosure enables a customer's knowledge graph **251** to be built subject to constraints expressed in the customer's semantic object model. Thus, the knowledge graphs **251** are generated by customers (e.g., enterprises or organizations) to create models of the edge devices **161***a*-**161***n* of an enterprise **160***a*-**160***n*, and the knowledge graphs **251** are input into the EOM **250** for visualizing the models (e.g., the nodes and links).

[0054] The models describe the assets (e.g., the nodes) of an enterprise (e.g., the edge devices **161***a*-**161***n*) and describe the relationship of the assets with other components (e.g., the links). The models also describe the schema (e.g., describe what the data is), and therefore the models are self-validating. For example, in one or more embodiments, the model describes the type of sensors mounted on any given asset (e.g., edge device **161***a*-**161***n*) and the type of data that is being sensed by each sensor. According to various embodiments, a key performance indicator (KPI) framework is used to bind properties of the assets in the extensible object model **250** to inputs of the KPI framework. Accordingly, the IoT platform **125** is an extensible, model-driven end-to-end stack including: two-way model sync and secure data exchange between the edge **115** and the cloud **105**, metadata driven data processing (e.g., rules, calculations, and aggregations), and model driven visualizations and applications. As used herein, "extensible" refers to the ability to extend a data model to include new properties/columns/fields, new classes/tables, and new relations. Thus, the IoT platform **125** is extensible with regards to edge devices **161***a*-**161***n* and the applications **146** that handle those devices **161***a*-**161***n*. For example, when new edge devices **161***a*-**161***n* are added to an enterprise **160***a*-**160***n* system, the new devices **161***a*-**161***n* will automatically appear in the IoT platform **125** so that the corresponding applications **146** understand and use the data from the new devices **161***a*-**161***n*.

[0055] In some cases, asset templates are used to facilitate configuration of instances of edge devices **161***a*-**161***n* in the model using common structures. An asset template defines the typical properties for the edge devices **161***a*-**161***n* of a given enterprise **160***a*-**160***n* for a certain type of device. For example, an asset template of a pump includes modeling the pump having inlet and outlet pressures, speed, flow, etc. The templates may also include hierarchical or derived types of edge devices **161***a*-**161***n* to accommodate variations of a base type of device **161***a*-**161***n*. For example, a reciprocating pump is a specialization of a base pump type and would include additional properties in the template. Instances of the edge device **161***a*-**161***n* in the model are configured to match the actual, physical devices of the enterprise **160***a*-**160***n* using the templates to define expected attributes of the device **161***a*-**161***n*. Each attribute is configured either as a static value (e.g., capacity is 1000 BPH) or with a reference to a time series tag that provides the value. The knowledge graph **250** can automatically map the tag to the attribute based on naming conventions, parsing, and matching the tag and attribute descriptions and/or by comparing the behavior of the time series data with expected behavior.

[0056] In certain embodiments, the modeling phase includes an onboarding process for syncing the models between the edge **115** and the cloud **105**. For example, in one or more embodiments, the onboarding process includes a simple onboarding process, a complex onboarding process, and/or a standardized rollout process. The simple onboarding process includes the knowledge graph **250** receiving raw model data from the edge **115** and running context discovery algorithms to generate the model. The context discovery algorithms read the context of the edge naming conventions of the edge devices **161***a***-161***n* and determine what the naming conventions refer to. For example, in one or more embodiments, the knowledge graph **250** receives “TMP” during the modeling phase and determine that “TMP” relates to “temperature.” The generated models are then published. In certain embodiments, the complex onboarding process includes the knowledge graph **250** receiving the raw model data, receiving point history data, and receiving site survey data. According to various embodiments, the knowledge graph **250** then uses these inputs to run the context discovery algorithms. According to various embodiments, the generated models are edited and then the models are published. The standardized rollout process includes manually defining standard models in the cloud **105** and pushing the models to the edge **115**.

[0057] The IoT layer **205** includes one or more components for device management, data ingest, and/or command/control of the edge devices **161***a***-161***n*. The components of the IoT layer **205** enable data to be ingested into, or otherwise received at, the IoT platform **125** from a variety of sources. For example, in one or more embodiments, data is ingested from the edge devices **161***a***-161***n* through process historians or laboratory information management systems. The IoT layer **205** is in communication with the edge connectors **165***a***-165***n* installed on the edge gateways **162***a***-162***n* through network **110**, and the edge connectors **165***a***-165***n* send the data securely to the IoT platform **205**. In some embodiments, only authorized data is sent to the IoT platform **125**, and the IoT platform **125** only accepts data from authorized edge gateways **162***a***-162***n* and/or edge devices **161***a***-161***n*. According to various embodiments, data is sent from the edge gateways **162***a***-162***n* to the IoT platform **125** via direct streaming and/or via batch delivery. Further, after any network or system outage, data transfer will resume once communication is re-established and any data missed during the outage will be backfilled from the source system or from a cache of the IoT platform **125**. According to various embodiments, the IoT layer **205** also includes components for accessing time series, alarms and events, and transactional data via a variety of protocols.

[0058] The enterprise integration layer **210** includes one or more components for events/messaging, file upload, and/or REST/OData. The components of the enterprise integration layer **210** enable the IoT platform **125** to communicate with third party cloud applications **211**, such as any application(s) operated by an enterprise in relation to its edge devices. For example, the enterprise integration layer **210** connects with enterprise databases, such as guest databases, customer databases, financial databases, patient databases, etc. The enterprise integration layer **210** provides a standard application programming interface (API) to third parties for accessing the IoT platform **125**. The enterprise integration layer **210** also enables the IoT platform **125** to communicate with the OT systems **163***a***-163***n* and IT applications **164***a***-164***n* of the enterprise **160***a***-160***n*. Thus, the enterprise integration layer **210** enables the IoT platform **125** to receive data from the third-party applications **211** rather than, or in combination with, receiving the data from the edge devices **161***a***-161***n* directly.

[0059] The data pipeline layer **215** includes one or more components for data cleansing/enriching, data transformation, data calculations/aggregations, and/or API for data streams. Accordingly, in one or more embodiments, the data pipeline layer **215** pre-processes and/or performs initial analytics on the received data. The data pipeline layer **215** executes advanced data cleansing routines including, for example, data correction, mass balance reconciliation, data conditioning, component balancing and simulation to ensure the desired information is used as a basis for further processing. The data pipeline layer **215** also provides advanced and fast computation. For example, in one or more embodiments, cleansed data is run through enterprise-specific digital twins. According to various embodiments, the enterprise-specific digital twins include a reliability advisor containing process models to determine the current operation and the fault models to trigger any early detection and determine an appropriate resolution. According to various embodiments, the digital twins also include an optimization advisor that integrates real-time economic data with real-time process data, selects the right feed for a process, and determines optimal process conditions and product yields.

[0060] According to various embodiments, the data pipeline layer **215** employs models and templates to define calculations and analytics. Additionally, or alternatively, according to various embodiments, the data pipeline layer **215** employs models and templates to define how the calculations and analytics relate to the assets (e.g., the edge devices **161**_a_-**161**_n_). For example, in an embodiment, a fan template defines fan efficiency calculations such that every time a fan is configured, the standard efficiency calculation is automatically executed for the fan. The calculation model defines the various types of calculations, the type of engine that should run the calculations, the input and output parameters, the preprocessing requirement and prerequisites, the schedule, etc. According to various embodiments, the actual calculation or analytic logic is defined in the template, or it may be referenced. Thus, according to various embodiments, the calculation model is employed to describe and control the execution of a variety of different process models. According to various embodiments, calculation templates are linked with the asset templates such that when an asset (e.g., edge device **161**_a_-**161**_n_) instance is created, any associated calculation instances are also created with their input and output parameters linked to the appropriate attributes of the asset (e.g., edge device **161**_a_-**161**_n_).

[0061] According to various embodiments, the IoT platform **125** supports a variety of different analytics models including, for example, curve fitting models, regression analysis models, first principles models, empirical models, engineered models, user-defined models, machine learning models, built-in functions, and/or any other types of analytics models. Fault models and predictive maintenance models will now be described by way of example, but any type of models may be applicable.

[0062] Fault models are used to compare current and predicted enterprise **160**_a_-**160**_n_ performance to identify issues or opportunities, and the potential causes or drivers of the issues or opportunities. The IoT platform **125** includes rich hierarchical symptom-fault models to identify abnormal conditions and their potential consequences. For example, in one or more embodiments, the IoT platform **125** drill downs from a high-level condition to understand the contributing factors, as well as determining the potential impact a lower-level condition may have. There may be multiple fault models for a given enterprise **160**_a_-**160**_n_ looking at different aspects such as process, equipment, control, and/or operations. According to various embodiments, each fault model identifies issues and opportunities in their domain and can also look at the same core problem from a different perspective. According to various embodiments, an overall fault model is layered on top to synthesize the different perspectives from each fault model into an overall assessment of the situation and point to the true root cause.

[0063] According to various embodiments, when a fault or opportunity is identified, the IoT platform **125** provides recommendations about optimal corrective actions to take. Initially, the recommendations are based on expert knowledge that has been pre-programmed into the system by process and equipment experts. A recommendation services module presents this information in a consistent way regardless of source, and supports workflows to track, close out, and document the recommendation follow-up. According to various embodiments, the recommendation follow-up is employed to improve the overall knowledge of the system over time as existing recommendations are validated (or not) or new cause and effect relationships are learned by users and/or analytics.

[0064] According to various embodiments, the models are used to accurately predict what will occur before it occurs and interpret the status of the installed base. Thus, the IoT platform **125** enables operators to quickly initiate maintenance measures when irregularities occur. According to various embodiments, the digital twin architecture of the IoT platform **125** employs a variety of modeling techniques. According to various embodiments, the modeling techniques include, for example, rigorous models, fault detection and diagnostics (FDD), descriptive models, predictive

maintenance, prescriptive maintenance, process optimization, and/or any other modeling technique.
[0065] According to various embodiments, the rigorous models are converted from process design simulation. In this manner, in certain embodiments, process design is integrated with feed conditions. Process changes and technology improvement provide business opportunities that enable more effective maintenance schedule and deployment of resources in the context of production needs. The fault detection and diagnostics include generalized rule sets that are specified based on industry experience and domain knowledge and can be easily incorporated and used working together with equipment models. According to various embodiments, the descriptive models identify a problem and the predictive models determines possible damage levels and maintenance options. According to various embodiments, the descriptive models include models for defining the operating windows for the edge devices **161***a*-**161***n*.
[0066] Predictive maintenance includes predictive analytics models developed based on rigorous models and statistic models, such as, for example, principal component analysis (PCA) and partial least square (PLS). According to various embodiments, machine learning methods are applied to train models for fault prediction. According to various embodiments, predictive maintenance leverages FDD-based algorithms to continuously monitor individual control and equipment performance. Predictive modeling is then applied to a selected condition indicator that deteriorates in time. Prescriptive maintenance includes determining an optimal maintenance option and when it should be performed based on actual conditions rather than time-based maintenance schedule. According to various embodiments, prescriptive analysis selects the right solution based on the company's capital, operational, and/or other requirements. Process optimization is determining optimal conditions via adjusting set-points and schedules. The optimized set-points and schedules can be communicated directly to the underlying controllers, which enables automated closing of the loop from analytics to control.
[0067] The data insight layer **220** includes one or more components for time series databases (TDSB), relational/document databases, data lakes, blob, files, images, and videos, and/or an API for data query. According to various embodiments, when raw data is received at the IoT platform **125**, the raw data is stored as time series tags or events in warm storage (e.g., in a TSDB) to support interactive queries and to cold storage for archive purposes. According to various embodiments, data is sent to the data lakes for offline analytics development. According to various embodiments, the data pipeline layer **215** accesses the data stored in the databases of the data insight layer **220** to perform analytics, as detailed above.
[0068] The application services layer **225** includes one or more components for rules engines, workflow/notifications, KPI framework, insights (e.g., actionable insights), decisions, recommendations, machine learning, and/or an API for application services. The application services layer **225** enables building of applications **146***a-d*. The applications layer **230** includes one or more applications **146***a-d* of the IoT platform **125**. For example, according to various embodiments, the applications **146***a-d* includes a buildings application **146***a*, a plants application **146***b*, an acro application **146***c*, and other enterprise applications **146***d*. According to various embodiments, the applications **146** includes general applications **146** for portfolio management, asset management, autonomous control, and/or any other custom applications. According to various embodiments, portfolio management includes the KPI framework and a flexible user interface (UI) builder. According to various embodiments, asset management includes asset performance, asset health, and/or asset predictive maintenance. According to various embodiments, autonomous control includes plantwide optimization, energy optimization and/or predictive maintenance. As detailed above, according to various embodiments, the general applications **146** is extensible such that each application **146** is configurable for the different types of enterprises **160***a*-**160***n* (e.g., buildings application **146***a*, plants application **146***b*, aero application **146***c*, and other enterprise applications **146***d*).
[0069] The applications layer **230** also enables visualization of performance of the enterprise **160***a*-

160_n_. For example, dashboards provide a high-level overview with drill downs to support deeper investigations. Recommendation summaries give users prioritized actions to address current or potential issues and opportunities. Data analysis tools support ad hoc data exploration to assist in troubleshooting and process improvement.

[0070] The core services layer **235** includes one or more services of the IoT platform **125**. According to various embodiments, the core services **235** include data visualization, data analytics tools, security, scaling, and monitoring. According to various embodiments, the core services **235** also include services for tenant provisioning, single login/common portal, self-service admin, UI library/UI tiles, identity/access/entitlements, logging/monitoring, usage metering, API gateway/dev portal, and the IoT platform **125** streams.

[0071] FIG. **3**A represents a planning model **300** according to one or more embodiments of the disclosure. FIG. **3**B represents a model predictive controller (MPC) model **350** according to one or more embodiments of the disclosure.

[0072] The planning model **300** and the MPC model **350** are an example of a multiscale model pair employed in certain embodiments to solve multi-level problems in a cascaded MPC architecture. In certain embodiments, the planning model **300** and/or the MPC model **350** are employed to support a cascaded MPC approach in an industrial process system (e.g., an industrial process control system, an industrial process control and automation system, etc.). In certain embodiments, the planning model **300** is a yield-based planning model.

[0073] As shown in FIG. **3**A, the planning model **300** identifies multiple units **302**, which generally operate to convert one or more input streams **304** of feed materials into one or more output streams **306** of processed materials. In this example, the units **302** denote components in an oil and gas refinery that convert a single input stream **304** (crude oil) into multiple output streams **306** (different refined oil/gas products). Various intermediate components **308** are created by the units **302**, and one or more storage tanks **310** could be used to store one or more of the intermediate components **308**. As shown in FIG. **3**B, the MPC model **350** identifies multiple components **352** of a single unit. Various valves and other actuators **354** can be used to adjust operations within the unit, and various APCs and other controllers **356** can be used to control the actuators within the unit.

[0074] In general, the planning model **300** analyzes an entire plant (or portion thereof) with a “bird's eye” view and thus represents individual units on a coarse scale. In one or more embodiments, the planning model **300** focuses on the inter-unit steady-state relationships pertaining to unit productions, product qualities, material and energy balances, and manufacturing activities inside the plant. In one or more embodiments, the planning model **300** is composed of process yield models and product quality properties. In one or more embodiments, the planning model **300** is constructed from a combination of various sources, such as planning tools, scheduling tools, yield validation tools and/or historical operating data. In one or more embodiments, the MPC model **350**, however, represents at least one unit on a finer scale. In one or more embodiments, the MPC model **350** focuses on the intra-unit dynamic relationships between controlled variables (CVs), manipulated variables (MVs), and disturbance variables (DVs) pertaining to the safe, smooth, and efficient operation of a unit. The time scales of the two models **300**, **350** are also different. In one or more embodiments, time horizon of the MPC model **350** ranges from minutes to hours, while a time horizon of the planning model **300** ranges from days to months. Note that a “controlled variable” generally denotes a variable whose value is controlled to be at or near a setpoint or within a desired range, while a “manipulated variable” generally denotes a variable that is adjusted in order to alter the value of at least one controlled variable. A “disturbance variable” generally denotes a variable whose value can be considered but not controlled or adjusted.

[0075] In one or more embodiments, the planning model **300** excludes non-production-related or non-economically related variables, such as pressures, temperatures, tank levels, and valve openings within each unit. For example, in one or more embodiments, the planning model **300** reduces a process unit to one or several material or energy yield vectors. The MPC model **350**, on the other hand, typically includes all of the operating variables for control purposes in order to help ensure the safe and effective operation of a unit. As a result, in one or more embodiments, the MPC model **350** includes many more variables for a unit compared to the planning model **300**. As a non-limiting example, the MPC model **350** for a Fluidized Catalytic Cracking Unit (FCCU) of an oil refinery contains approximately 100 CVs (outputs) and **40** MVs (inputs). In certain embodiments, the planning model **300** of the same unit determines key causal relationships between the feed quality and operating modes (as inputs) and the FCCU product yield and quality (as outputs). As a non-limiting example, the planning model **300** includes three or four inputs and ten outputs.

[0076] FIG. **4** illustrates a system **400** according to one or more described features of one or more embodiments of the disclosure. In one or more embodiments, the system **400** facilitates harmonizing control/optimization solutions under disruptive batch operations say, in a process plant. The system **400** includes the planning model **300**, current component inventory **402**, spent components **404**, products **406** being blended, and the one or more output streams **306** of processed materials. In one or more embodiments, the spent components **404** are associated with an asynchronous update scheme and/or a virtual content tracker. Also, in one or more embodiments, the system **400** initializes the asynchronous update scheme based on corresponding statuses of one or more operations in the process plant. Furthermore, in one or more embodiments, the spent components **404** are implemented between continuous control associated with the planning model **300** (e.g., one or more optimization processes associated with the planning model **300**) and the products **406** being blended. In one or more embodiments, the current component inventory **402** is employed for one or more batch operations (e.g., in a batch process) associated with the products **406** being blended. In one or more embodiments, the products **406** being blended are blending components (e.g., intermediate components) that are combined to create the one or more output streams **306** of processed materials. In one or more embodiments, the one or more output streams **306** correspond to final products. In one or more embodiments, the spent components **404** monitor content in the current component inventory **402** and/or an amount of material in the current component inventory **402**. Additionally, or alternatively, in one or more embodiments, the spent components **404** monitor content in the products **406** being blended and/or an amount of material in the products **406** being blended. In this regard, in some embodiments, the spent components **404** tracks a status of an operation (or an order) based on the amount of material in the current component inventory **402** and/or the amount of material in the products **406** being blended.

[0077] In certain embodiments, if a batch operation is completed, the spent components **404** update an inventory count associated with the current component inventory **402**. Furthermore, in certain embodiments, if a batch operation is completed, the spent components **404** additionally or alternatively resets a virtual content tracker associated with the current component inventor **402** and/or the products **406** being blended to zero. In certain embodiments, if demand associated with the current component inventory **402** (e.g., a number of requests for product associated with the current component inventory **402**) is altered, the spent components **404** update demand data for the current component inventory **402**. In certain embodiments, if demand associated with a continuous industrial process has a receding horizon formulation, the spent components **404** reset a new receding time horizon to reflect time remaining for fulfilling the demand and/or for completing a task associated with the continuous industrial process. As such, a receding horizon solution for an industrial process can be converted to an asynchronous receding horizon solution.

[0078] As an example, for N physical blenders, Vs is comprised of N columns of spent component inventories:

[00001]$v_s = V_{s,1} + V_{s,2} + \text{ .Math. } + V_{s,N}$ [0079] where, for each column, “V.sub.s, i” represents the spent components **404** for blending the i-th product in a batch operation. In an embodiment, “V.sub.s, i” is determined by measuring and/or accumulating component draws for blending the i-th product since the start of a batch operation. In another embodiment, “V.sub.s, i” is determined by

measuring volume (or a level) in the i-th product tank and back-calculating the spent components **404**.

[0080] In certain embodiments, an asynchronous update scheme associated with the spent components **404** is executed based on a schedule for an industrial process. In certain embodiments, an asynchronous update scheme associated with the spent components **404** is executed based on one or more conditions being satisfied for an industrial process. Also, in some embodiments, an asynchronous update scheme associated with the spent components **404** is provided with a tracking flag to indicate a status of an order based on the amount of material in the current component inventory **402** and/or the amount of material in the products **406** being blended.

[0081] In one embodiment, a future component vector c.sub.m for optimization of an industrial process corresponds to:

[00002] $c_m = \int_0^{m\text{ - ts}} G_F * \text{udt} + V_0 + V_s = \int_t^{\text{Ts}*} G_F * \text{udt} + V_0 + V_s$ [0082] where t.sub.s is an elapsed time since a previous asynchronous update scheme, and Ts* is the TimeStamp of the previous asynchronous update scheme+m. Gr corresponds to a Laplace transfer function for a continuous process (e.g., a continuous optimization process) related to an industrial process. For example, in an embodiment, GF corresponds to a Laplace transfer function for a continuous portion of an industrial plant (e.g., a processing plant, a refinery, etc.). In certain embodiments, GF corresponds to a model partition for component flow rates (e.g., a model related to an effect on component flows). Further details are provided by U.S. patent application Ser. No. 14/336,888 filed Jul. 21, 2014 and U.S. patent application Ser. No. 17/084,157 filed Oct. 29, 2020, the entire contents of each of which are expressly incorporated by reference herein in their entirety. In certain embodiments, if any batch blend is completed, a product amount associated with the batch blend is subtracted from Pm and a corresponding “V.sub.s, i” is reset to zero. In one embodiment, if demand is changed either in by a certain amount or timing is altered (e.g., since a previous asynchronous update scheme), the new demand is added to Pm, t.sub.s is reset to zero, and m is set to a new value.

[0083] In certain embodiments, an asynchronous update scheme is tuned. For instance, for a time period (t, t+m) where ‘t’ is the present time, if all product deliveries within the time period are confirmed, the time period (t, t+m) is defined as a confirmed period. In one embodiment, the internal optimization time horizon, m1, is shorter than or equal to m. In general, the shorter m1 is, the more quickly the current inventory V0 will be depleted to a low limit.

[0084] In one embodiment, the confirmed demand vector Pm is modified to a modified demand vector Pm1=(m1/m)*Pm. Furthermore, the future component vector cm is altered to a revised cml that corresponds to:

[00003] $c_{m1} = \int_0^{m1\text{ - ts}} G_F * \text{udt} + V_0 + V_s = \int_t^{\text{Ts1}*} G_F * \text{udt} + V_0 + V_s$ [0085] where T.sub.s1* is the TimeStamp of the last the previous asynchronous update scheme+m1.

[0086] Whereas, in another embodiment, a future component vector c.sub.m for optimization of an industrial process corresponds to:

[00004] $c_m = \int_0^{m\text{ - Ets}} G_F * u\text{dt} + V_0 + V_s = \int_t^{Ts} G_F * u\text{dt} + V_0 + V_s$ [0087] where E.sub.ts corresponds to elapsed time since the last asynchronous update scheme run, and T.sub.s corresponds to demand end time to produce one or more optimal components c.sub.m. Additionally, in response to a change in demand for an industrial process product, the confirmed demand vector Pm is modified to a modified demand vector P.sub.m1= ((T.sub.s1−T.sub.0)/m.sub.0)*P.sub.m, such that m.sub.0=T.sub.s−T.sub.0, where T.sub.s1 is end of an optimization horizon (OH) at time t and T.sub.0 corresponds to time at which a last asynchronous update scheme was performed. Also, the future component vector c.sub.m is altered to a revised c.sub.m1 that corresponds to:

[00005] $c_{m1} = \int_0^{\text{OH}} G_F * u\text{dt} + V_0 + V_s = \int_t^{\text{Ts1}} G_F * u\text{dt} + V_0 + V_s$ [0088] where T.sub.s1 is end of the optimization horizon (OH) at time t.

[0089] FIG. **5** illustrates a system **500** according to one or more described features of one or more embodiments of the disclosure. The system **500** includes an optimizer controller **502** and one or more multivariable MPC controllers **504***a-n*. For example, in one or more embodiments, the optimizer controller **502** is a primary controller (e.g., a master MPC controller) and the one or more multivariable MPC controllers **504***a-n* are one or more secondary controllers. In one or more embodiments, the optimizer controller **502** and the one or more multivariable MPC controllers **504***a-n* represent respective computing devices. For example, in one or more embodiments, the optimizer controller **502** and the one or more multivariable MPC controllers **504***a-n* each include one or more processing devices and one or more memories for storing instructions and data used, generated, or collected by the one or more processing devices. In one or more embodiments, the optimizer controller **502** and the one or more multivariable MPC controllers **504***a-n* each also include at least one network interface such as one or more Ethernet interfaces or one or more wireless transceivers. In one or more embodiments, an industrial process control system associated with the optimizer controller **502** and the one or more multivariable MPC controllers **504***a-n* includes one or more sensors, one or more actuators, one or more other controllers, one or more servers, one or more operator stations, one or more networks, and/or one or more other components.
[0090] In one or more embodiments, the optimizer controller **502** and the one or more multivariable MPC controllers **504***a-n* are configured as a cascaded MPC architecture for plantwide control and optimization to facilitate plantwide optimization as part of an automatic control and automation system. In one or more embodiments, the optimizer controller **502** is configured to use a planning model (e.g., the planning model **300**) as a seed model. In one or more embodiments, the optimizer controller **502** performs plantwide economic optimization using one or more optimization processes to control production inventories, manufacturing activities, or product qualities inside a plant. In one or more embodiments, the optimizer controller **502** is cascaded on top of the one or more multivariable MPC controllers **504***a-n*. In certain embodiments, the one or more multivariable MPC controllers **504***a-n* represent controllers at the unit level (Level 3) of a system, and each multivariable MPC controller from the one or more multivariable MPC controllers **504***a-n* provides the optimizer controller **502** with one or more respective operating state and/or respective constraints. As such, in one or more embodiments, a plantwide optimization solution provided by the optimizer controller **502** accounts for unit-level operating constraints from the one or more multivariable MPC controllers **504***a-n*. Jointly, the MPC cascade associated with the optimizer controller **502** and the one or more multivariable MPC controllers **504***a-n* simultaneously provides both decentralized controls (such as at the unit level) and centralized plantwide optimization (such as at the plant level) in a single consistent control system. The phrases “plantwide optimization” or “plantwide control” refers to optimization or control of multiple units in an industrial facility, regardless of whether those multiple units represent every single unit in the industrial facility.
[0091] In one or more embodiments, the MPC cascade solution associated with the optimizer controller **502** and the one or more multivariable MPC controllers **504***a-n* enables embedded real-time planning solutions to honor lower-level operating constraints. By cross-leveraging both planning and control models online, the MPC cascade solution associated with the optimizer controller **502** and the one or more multivariable MPC controllers **504***a-n* provides for a “reduced-horizon” form of planning optimization within a closed-loop control system in real-time. For instance, in one or more embodiments, the optimizer controller **502** is configured to provide reduced-horizon planning optimization. In one or more embodiments, the MPC cascade solution associated with the optimizer controller **502** and the one or more multivariable MPC controllers **504***a-n* is employed to automatically perform just-in-time production plans via the one or more multivariable MPC controllers **504***a-n.*
[0092] In one or more embodiments, a planning model (e.g., the planning model **300**) determines a

region for optimization and the MPC cascade solution associated with the optimizer controller **502** and the one or more multivariable MPC controllers **504***a-n* coordinates units and manages an industrial plant to provide optimal operating conditions for the industrial plant. In one or more embodiments, each of the one or more multivariable MPC controllers **504***a-n* comprise an embedded economic optimizer to facilitate providing optimal operating conditions for the industrial plant. In one or more embodiments, each of the one or more multivariable MPC controllers **504***a-n* additionally or alternatively provide multivariable control functions to facilitate providing optimal operating conditions for the industrial plant. In one or more embodiments, the optimizer controller **502** employs a planning model (e.g., the planning model **300**) to provide an initial steady-state gain matrix and/or relevant model dynamics using operating data of the industrial plant. In one or more embodiments, the optimizer controller **502** is configured to control product inventories, manufacturing activities, and/or product qualities within the industrial plant. In one or more embodiments, the optimizer controller **502** provides real-time planning optimization for the industrial plant. In one or more embodiments, the one or more multivariable MPC controllers **504***a-n* provide the optimizer controller **502** with future predictions and/or operating constraints for each unit of the industrial plant.

[0093] FIG. **6** illustrates an industrial process **602** according to one or more described features of one or more embodiments of the disclosure. In one or more embodiments, the industrial process **602** is an industrial process for an industrial plant. Furthermore, in one or more embodiments, the industrial process **602** produces one or more industrial process products (e.g., the one or more output streams **306** of processed materials). The industrial process **602** includes a continuous optimization subprocess **604**, an asynchronous update subprocess **606** and a discontinuous batch operation subprocess **608**. In one or more embodiments, the continuous optimization process **604** is configured for optimizing continuous conversion of one or more input streams **304** of feed materials (e.g., petroleum stocks) into the current component inventory **402** (e.g., blending components). In one or more embodiments, the continuous optimization process **604** is associated with the optimizer controller **502** and/or the one or more multivariable MPC controllers **504***a-n*. Additionally, in one or more embodiments, the discontinuous batch operation subprocess **608** is configured for non-continuous conversion of the current component inventory **402** (e.g., blending components) into the one or more output streams **306** of processed materials (e.g., refinery products). For example, in one or more embodiments, at least a portion of the discontinuous batch operation subprocess **608** occurs in a batch fashion. In certain embodiments, at least a portion of the discontinuous batch operation subprocess **608** occurs in a continuous fashion or assembly-line fashion during an interval of time that is different than an interval of time associated with the continuous optimization process **604**. In one or more embodiments, the asynchronous update subprocess **606** is configured for harmonizing the discontinuous batch operation subprocess **608** with respect to the continuous optimization process **604**. For example, in one or more embodiments, the asynchronous update subprocess **606** is configured to determine how one or more batch operations associated with the discontinuous batch operation subprocess **608** should be updated when the continuous optimization process **604** receives new input at a pre-specified frequency.

[0094] FIG. **7** illustrates a system **700** that provides an exemplary environment according to one or more described features of one or more embodiments of the disclosure. According to an embodiment, the system **700** includes an industrial process computer system **702** to facilitate a practical application of industrial process control and/or industrial process simulation for an industrial process **602**. In one or more embodiments, the industrial process computer system **702** provides for management of industrial process optimization related to batch operations. In certain embodiments, the industrial process computer system **702** facilitates a practical application of machine learning technology to facilitate management of industrial process optimization related to batch operations. In one or more embodiments, the industrial process computer system **702** analyzes real-time industrial process data related to the industrial process **602** to provide optimization, cost savings, and/or improved efficiency for the industrial process **602**. In one or more embodiments, the industrial process computer system **702** is implemented via a controller (e.g., the optimizer controller **502** or another controller in communication with the optimizer controller **502**).

[0095] In an embodiment, the industrial process computer system **702** is a server system (e.g., a server device) that facilitates a cloud-based industrial control platform. In one or more embodiments, the industrial process computer system **702** is a device with one or more processors and a memory. In one or more embodiments, the industrial process computer system **702** is a computer system from the computer systems **120**. For example, in one or more embodiments, the industrial process computer system **702** is implemented via the cloud **105**. The industrial process computer system **702** is also related to one or more technologies, such as, for example, industrial technologies, process plant technologies, oil and gas technologies, petrochemical technologies, refinery technologies, supply chain analytics technologies, sensor technologies, Internet of Things (IoT) technologies, enterprise technologies, smart building technologies, connected building technologies, connected asset technologies, connected edge-device technologies, HVAC technologies, modeling technologies, energy optimization technologies, predictive maintenance technologies, asset performance management technologies, data analytics technologies, digital transformation technologies, cloud computing technologies, cloud database technologies, server technologies, network technologies, wireless communication technologies, machine learning technologies, artificial intelligence technologies, digital processing technologies, electronic device technologies, computer technologies, aircraft technologies, cybersecurity technologies, navigation technologies, asset visualization technologies, procurement technologies, and/or one or more other technologies.

[0096] Moreover, the industrial process computer system **702** provides an improvement to one or more technologies such as industrial technologies, process plant technologies, oil and gas technologies, petrochemical technologies, refinery technologies, supply chain analytics technologies, sensor technologies, IoT technologies, enterprise technologies, smart building technologies, connected building technologies, connected asset technologies, connected edge-device technologies, HVAC technologies, modeling technologies, energy optimization technologies, predictive maintenance technologies, asset performance management technologies, data analytics technologies, digital transformation technologies, cloud computing technologies, cloud database technologies, server technologies, network technologies, wireless communication technologies, machine learning technologies, artificial intelligence technologies, digital processing technologies, electronic device technologies, computer technologies, aircraft technologies, cybersecurity technologies, navigation technologies, asset visualization technologies, procurement technologies, and/or one or more other technologies. In an implementation, the industrial process computer system **702** improves performance of an industrial plant. For example, in one or more embodiments, the industrial process computer system **702** optimizes one or more industrial processes for an industrial plant, improves processing efficiency of one or more controllers of an industrial plant, reduces power consumption of one or more controllers of an industrial plant, optimizes energy usage related to one or more controllers of an industrial plant, etc. Additionally, or alternatively, in another implementation, the industrial process computer system **702** improves performance of a computing device (e.g., a server, a controller, a processor, etc.). For example, in one or more embodiments, the industrial process computer system **702** improves processing efficiency of a computing device, reduces power consumption of a computing device, improves quality of data provided by a computing device, etc.

[0097] The industrial process computer system **702** includes an asynchronous update component **704**, a virtual content component **706** and/or a control component **708**. Additionally, in certain embodiments, the industrial process computer system **702** includes a processor **710** and/or a memory **712**. In certain embodiments, one or more aspects of the industrial process computer system **702** (and/or other systems, apparatuses and/or processes disclosed herein) constitute executable instructions embodied within a computer-readable storage medium (e.g., the memory **712**). For instance, in an embodiment, the memory **712** stores computer executable component and/or executable instructions (e.g., program instructions). Furthermore, the processor **710** facilitates execution of the computer executable components and/or the executable instructions (e.g., the program instructions). In an example embodiment, the processor **710** is configured to execute instructions stored in the memory **712** or otherwise accessible to the processor **710**.

[0098] The processor **710** is a hardware entity (e.g., physically embodied in circuitry) capable of performing operations according to one or more embodiments of the disclosure. Alternatively, in an embodiment where the processor **710** is embodied as an executor of software instructions, the software instructions configure the processor **710** to perform one or more algorithms and/or operations described herein in response to the software instructions being executed. In an embodiment, the processor **710** is a single core processor, a multi-core processor, multiple processors internal to the industrial process computer system **702**, a remote processor (e.g., a processor implemented on a server), and/or a virtual machine. In certain embodiments, the processor **710** is in communication with the memory **712**, the asynchronous update component **704**, the virtual content component **706** and/or the control component **708** via a bus to, for example, facilitate transmission of data among the processor **710**, the memory **712**, the asynchronous update component **704**, the virtual content component **706** and/or the control component **708**. The processor **710** may embodied in a number of different ways and can, in certain embodiments, includes one or more processing devices configured to perform independently. Additionally, or alternatively, in one or more embodiments, the processor **710** includes one or more processors configured in tandem via a bus to enable independent execution of instructions, pipelining of data, and/or multi-thread execution of instructions.

[0099] The memory **712** is non-transitory and includes, for example, one or more volatile memories and/or one or more non-volatile memories. In other words, in one or more embodiments, the memory **712** is an electronic storage device (e.g., a computer-readable storage medium). The memory **712** is configured to store information, data, content, one or more applications, one or more instructions, or the like, to enable the industrial process computer system **702** to carry out various functions in accordance with one or more embodiments disclosed herein. As used herein in this disclosure, the term “component,” “system,” and the like, is a computer-related entity. For instance, “a component,” “a system,” and the like disclosed herein is either hardware, software, or a combination of hardware and software. As an example, a component is, but is not limited to, a process executed on a processor, a processor, circuitry, an executable component, a thread of instructions, a program, and/or a computer entity.

[0100] In an embodiment, the industrial process computer system **702** (e.g., the asynchronous update component **704** and/or the processor **710** of the industrial process computer system **702**) receives a request **720**. In one or more embodiments, the request **720** is an optimization request to optimize an industrial process (e.g., the industrial process **602**) that produces one or more industrial process products. In one or more embodiments, the request **720** is associated with a continuous optimization subprocess (e.g., the continuous optimization subprocess **604**). Furthermore, in one or more embodiments, the request **720** is generated by a controller (e.g., the optimizer controller **502** and/or the one or more multivariable MPC controllers **504***a-n*). Alternatively, in one or more embodiments, the request **720** is generated by a user device (e.g., in response to user input provided via a user interface of the user device). The user device is a mobile computing device, a smartphone, a tablet computer, a mobile computer, a desktop computer, a laptop computer, a workstation computer, a wearable device, a virtual reality device, an augmented reality device, or another type of user device located remote from the industrial process computer system **702**. In one or more embodiments, the request **720** is received in response to a determination that time data for the industrial process **602** (e.g., a schedule for the industrial process) satisfies a defined criterion. In one or more embodiments, the request **720** is received in response to a determination that a condition (e.g., an industrial process event) associated with the industrial process **602** satisfies a defined criterion. Also, in one or more embodiments, the request **720** received at the industrial process computer system **702** can correspond to one or more orders to produce one or more industrial process products or one or more output streams **306**.

[0101] In one or more embodiments, the industrial process **602** is associated with components of the edge **115** such as, for example, one or more enterprises **160***a*-**160***n*. In one or more embodiments, the industrial process **602** is associated with one or more edge devices from the one or more edge devices **161***a*-**161***n*. In one or more embodiments, the industrial process computer system **702** is in communication with one or more portions of an industrial plant (e.g., the optimizer controller **502** and/or the one or more multivariable MPC controllers **504***a*-*n*) via the network **110**. For example, in certain embodiments, the industrial process computer system **702** receives data via the network **110** and/or transmits data via the network **110**. In certain embodiments, the industrial process **602** incorporates encryption capabilities to facilitate encryption of one or more portions of data received via the network **110** and/or one or more portions of data transmitted via the network **110**. In one or more embodiments, the network **110** is a Wi-Fi network, a Near Field Communications (NFC) network, a Worldwide Interoperability for Microwave Access (WiMAX) network, a personal area network (PAN), a short-range wireless network (e.g., a Bluetooth® network), an infrared wireless (e.g., IrDA) network, an ultra-wideband (UWB) network, an induction wireless transmission network, and/or another type of network.

[0102] In one or more embodiments, the request **720** includes inventory data. In one or more embodiments, the inventory data is indicative of an inventory level for one or more feed products associated with one or more blending components of a batch operation subprocess (e.g., the discontinuous batch operation subprocess **608**). For instance, in one or more embodiments, the inventory data includes information associated with the current component inventory **402** and/or the products **406** being blended. In certain embodiments, the inventory data includes information associated with component flow rate and/or component flow quality for the current component inventory **402** and/or the products **406** being blended. Also, in one or more embodiments, the inventory data can indicate a status of an order based on the information associated with the current component inventory **402** and/or the products **406** being blended. In one or more embodiments, the request **720** additionally or alternatively includes an industrial process identifier that indicates an identity of the industrial process **602** and/or one or more assets associated with the industrial process **602**. The industrial process identifier in the request **720** is a digital code such as, for example, a machine-readable code, a combination of numbers and/or letters, a string of bits, a barcode, a Quick Response (QR) code, or another type of identifier for the industrial process **602**. Furthermore, the asset identifier in the request **720** facilitates identification of the industrial process **602**.

[0103] The asynchronous update component **704** employs the request **720** and/or information related to the request **720** to facilitate optimization of one or more portions of the industrial process **602**. For example, in one or more embodiments, asynchronous update component **704** employs the request **720** and/or information related to the request **720** to facilitate harmonizing a discontinuous batch operation subprocess of the industrial process (e.g., the discontinuous batch operation subprocess **608**) with respect to a continuous optimization process of the industrial process (e.g., the continuous optimization process **604**). In one or more embodiments, in response to the request **720**, the asynchronous update component **704** determines component spent characteristics for one or more blending components of a batch operation subprocess (e.g., the discontinuous batch operation subprocess **608**). For example, in an embodiment, the asynchronous update component **704** determines content of material used by one or more blending components of one or more batch operations associated with the batch operation subprocess. Additionally, or alternatively, in an embodiment, the asynchronous update component **704** determines an amount of material used by one or more blending components of one or more batch operations associated with the batch operation subprocess. In one or more embodiments, the asynchronous update component **704** determines the component spent characteristics based on industrial process product characteristics for the industrial process product. In one or more embodiments, the asynchronous update component **704** determines the component spent characteristics based on monitoring an amount of material included in the one or more blending components. In one or more embodiments, the asynchronous update component **704** determines the component spent characteristics by predicting one or more component spent characteristics (e.g., using machine learning). For example, in one or more embodiments, the asynchronous update component **704** performs a machine learning process with respect to industrial process data related to the industrial process **602**. In certain embodiments, the asynchronous update component **704** determines at least a portion of the component spent characteristics based on relationships, correlations and/or predictions between aspects of industrial process data related to the industrial process **602**. In one or more embodiments, the asynchronous update component **704** determines the component spent characteristics based on one or more virtual spent components for the one or more blending components.

[0104] In one or more embodiments, the virtual content component **706** updates, based on the component spent characteristics and/or the inventory data, demand data for one or more feed products employed to produce the one or more blending components. In one or more embodiments, the demand data is related to demand associated with the current component inventory **402** and/or the products **406** being blended. In one or more embodiments, the demand data is related to demand associated with the one or more output streams **306** of processed materials (e.g., the refinery products). In one or more embodiments, the demand data includes a maximum demand value (e.g., a maximum demand value plus a predefined demand value) and/or a minimum demand value (e.g., a minimum demand value) for respective product types such as, for example, regular gasoline, premium gasoline, jet gasoline, diesel gasoline, etc. In one or more embodiments, the demand data includes future demand data. For example, in one or more embodiments, the demand data represents predicted demand for a product or product grade for a next M number of days, where M is an integer. In certain embodiments, the demand data accounts for supply chain characteristics of a product such as, for example, multiple liftings and/or deliveries of a product. In one or more embodiments, in response to a change in demand for one or more industrial process products (e.g., the current component inventory **402** and/or the products **406** being blended), the virtual content component **706** updates one or more portions of the demand data.

[0105] In one or more embodiments, the control component **708** is configured to generate control data **722** based on the demand data determined by the virtual content component **706**. In one or more embodiments, the control data **722** includes one or more control signals configured based on the demand data. In one or more embodiments, the control data **722** includes one or more setpoints for one or more assets. A setpoint is an asset setting (e.g., an asset setpoint) for a respective asset. For instance, in one or more embodiments, the one or more setpoints include one or more voltage values, one or more current values, one or more switch states, and/or one or more other configuration settings for respective assets. In one or more embodiments, the control component **708** transmits the control data **722** (e.g., the one or more control signals) to a controller configured for optimization associated with the industrial process **602**. In one or more embodiments, the controller corresponds to the optimizer controller **502** and/or the one or more multivariable MPC controllers **504***a-n*. In one or more embodiments, the control component **708** alters one or more portions of an optimization process (e.g., the continuous optimization subprocess **604**) based on the control data **722** (e.g., the one or more control signals). For example, in one or more embodiments, the control component **708** tunes an interval of time for the optimization associated with the industrial process **602** based on the demand data. In one or more embodiments, the control component **708** additionally or alternatively generates one or more batch blending recipes for one or more batch processes (e.g., the discontinuous batch operation subprocess **608**) based on the demand data. In one or more embodiments, in response to determining that a batch operation subprocess (e.g., the discontinuous batch operation subprocess **608**) satisfies a defined criterion, the control component **708** additionally or alternatively updates one or more portions of the inventory data. For example, in response to determining that a batch operation subprocess is complete (e.g., one or more subprocesses of the discontinuous batch operation subprocess **608** is complete), the control component **708** additionally or alternatively updates one or more portions of the inventory data. In one or more embodiments, in response to determining that a batch operation subprocess (e.g., the discontinuous batch operation subprocess **608**) satisfies a defined criterion, the control component **708** additionally or alternatively updates one or more portions of the component spent characteristics. For example, in response to determining that a batch operation subprocess is complete (e.g., one or more subprocesses of the discontinuous batch operation subprocess **608** is complete), the control component **708** additionally or alternatively updates one or more portions of the component spent characteristics. Also, in one or more embodiments, the control component **708** is configured to initialize the asynchronous update scheme associated with the asynchronous update component **704** and/or the virtual content component **706**. Per this aspect, the initialization is based at least on the component spent characteristics, the inventory data, and/or the demand data. Further, in this regard, the control component **708** via the one or more control signals controls the asynchronous update component **704**, the virtual content component **706**, and/or the optimizer controller **502** based on the initialization. An exemplary initialization plan is described in more details in accordance with FIG. **7**D of the present disclosure.

[0106] In certain embodiments, the control component **708** generates a user-interactive electronic interface that renders a visual representation of the demand data. For example, in an embodiment, the control component **708** configures a visualization (e.g., a dashboard visualization for display via a user interface of a computing device). In an embodiment, the user-interactive electronic interface facilitates presentation and/or interaction with one or more portions of the demand data. In one or more embodiments, the user-interactive electronic interface renders one or more interactive media elements via a set of pixels. An example user interface rendering demand data is described in more details in accordance with FIG. **7**C of the current disclosure. In another embodiment, the control component **708** transmits, to a computing device, one or more notifications associated with the demand data. In an exemplary embodiment, a notification advises that one or more portions of the industrial process **602** is operating inefficiently. In another embodiment, the control component **708** performs another type of action associated with the application services layer **225**, the applications layer **230**, and/or the core services layer **235** based on the demand data.

[0107] FIG. **7**A illustrates a system **700**A according to one or more described features of one or more embodiments of the disclosure. FIG. **7**B illustrates a system **700**B according to one or more described features of one or more embodiments of the disclosure. In one or more embodiments, the system **700**A and the system **700**B are used for controlling the industrial process **602** (of FIG. **6**). In one or more embodiments, each of the system **700**A and the system **700**B may correspond to the system **400** of FIG. **4** that includes the planning model **300**, the current component inventory **402**, the spent components **404**, the products **406** being blended, and the one or more output streams **306** of processed materials.

[0108] The system **700**A may also correspond to the system **700** of FIG. **7** that includes the memory **712** (of FIG. **7**) including one or more instructions and the processor **710** (of FIG. **7**), which is communicatively coupled with the memory **712**, such as the processor **710** of the system **700**A is configured to control the industrial process **602** that includes a continuous process and a batch operation.

[0109] In one or more implementation, the processor **702** is configured to receive an optimization request in response to a determination that time data for the industrial process satisfies a defined criterion. Firstly, the processor **710** is configured to determine the time data (e.g., two days) to optimize the industrial process **602** and produce the industrial process product. Thereafter, if the time data (e.g., two days) for the industrial process **602** satisfies the defined criterion, then the processor **710** is configured to receive the optimization request to optimize the industrial process **602** (e.g., a schedule for the industrial process **602**) that produces the industrial process product, such as to fulfill at least one order from the one or more orders. Optionally, the optimization request may be received from a user device.

[0110] In operation, the processor **710** is configured to determine a first time period to produce the industrial process product to fulfill one or more orders. In an implementation, the first time period corresponds to a total time that is required to produce the industrial process product through the batch operation to fulfill the one or more orders. In one example, the batch operation is performed to blend different blending components (e.g., the current component inventory **402**) to form a refinery product, such as the products **406**. However, the batch operation can disrupt one or more other industrial processes and/or one or more optimization operations related to a processing facility. For example, the batch operation can be performed with the first time period that includes an on-and-off functionality associated with a starting time and/or duration that is generally not predictable. Optionally, the starting time may also be referred to as a time at which a last asynchronous update scheme was performed, and the duration that is generally not predictable may also be referred to as a demand end time of the batch operation. Optionally, the first time period may be represented by T.sub.1st (or m.sub.0), such as a time at which a last asynchronous update scheme was performed may be represented by To, and the demand end time may be represented by T.sub.s for the batch operation. In an example, the first time period referred to as a demand window, which can vary based on the one or more orders for the batch operation, such as can vary from 10 days to 12 days, 10 days to 15 days, 12 days to 16 days, and the like. Furthermore, the batch operation can cause one or more other industrial processes and/or one or more optimization operations related to a processing facility to be performed in an inefficient manner (e.g., a cyclic manner) with respect to operation of the processing facility. It is therefore advantageous to optimize the batch operation with respect to one or more other industrial processes and/or one or more optimization operations related to a processing facility to achieve optimal performance and/or optimal efficiency for one or more industrial processes.

[0111] The processor **710** is further configured to determine an optimization time period, such as the optimization time period is less than the first time period. In other words, in response to the received optimization request, the processor **710** is configured to determine the optimization time period to optimize the industrial process **602** that produces the industrial process product, such as to fulfill at least one order from the one or more orders. In an example, the optimization time period may be represented by optimization horizon (OH), which can be set (i.e., can be increased or decreased) by a user. Moreover, the optimization time period is used by the processor **710** to produce the industrial process product based on the received optimization request to fulfill at least one order from the one or more orders. For example, if the optimization request is received by the processor **710** to produce one barrel of regular gas to fulfill at least one order, then the processor **710** is configured to determine the optimization time period to produce the one barrel of the regular gas to fulfill the at least one order. Therefore, the optimization time period (i.e., OH or m.sub.1) may also be referred to as a desired demand window to fulfil the received optimization request. Furthermore, the optimization time period is less than the first time period. For example, a time period between a time ‘t’ and a time T.sub.s1 corresponds to the optimization time period (i.e., OH=T.sub.s1−t) or the desired demand window. Moreover, if the first time period between the last asynchronous update scheme To and the demand end time T.sub.s is for sixteen days, then the optimization time period between the time ‘t’ and the time T.sub.s1 can be for two days, which is less than the first time period. In addition, a time period between the demand end time T.sub.s and the time ‘t’ corresponds to a remaining demand window to fulfill remaining orders. In an example, if the first time period (i.e., the demand window) to fulfill the one or more orders is for 12 days, the optimization time period (i.e., the desired demand window) to fulfill the at least one order is for 2 days, then the remaining demand window to fulfil the remaining orders is for 10 days. In one or more implementation, if all the industrial product orders based on the received optimization request within the first time period are confirmed, then the first time period (i.e., T.sub.0, T.sub.s) is defined as a confirmed period. Furthermore, the processor **710** is configured to select a maximum (or sufficiently large) value for the T.sub.s(m), such as the first time period (i.e., T.sub.0, T.sub.s) remains confirmed to fulfill the one or more orders. Optionally, a user can enter all the confirmed orders to maximize the T.sub.s. Therefore, at any time, ‘t’, the controller **710** can perform plantwide control and optimization (PWO), which can have infinite options of corresponding optimization horizon (OH)=(t, T.sub.s1), as long as T.sub.s1≤T.sub.s.

[0112] In response to determination of the optimization time period, the processor **710** is further configured to update industrial process product characteristics based on at least the first time period and the optimization time period. In an implementation, the industrial process product characteristics are related to the one or more output streams **306** of processed materials. For example, the industrial process product characteristics are associated with an amount of production of at least one or more of: typical refinery products, regular gas, refinery fuel gas, liquid petroleum gas (LPG), regular gasoline, premium gasoline, solvents, aviation fuels, diesels, heating oils, lube oils, greases, asphalts, industrial fuels, refinery fuel oil, and the like. In an example, the industrial process product characteristics for the regular gas are updated based on the first time period and the optimization time period based on the current component inventory **402**. Optionally, updating the industrial process product characteristics for the regular gas corresponds to determination of total amount of production of the regular gas within the first time period (e.g., within 10 days) and required production of the regular gas within the optimization time period (e.g., within 2 days) based on the received optimization request. Similarly, the industrial process product characteristics can be updated for other output streams based on the received optimization request and based on the first time period and the optimization time period.

[0113] Furthermore, in response to determination of the optimization time period, the processor **710** is further configured to update blending component characteristics indicative of availability of one or more blending components for the batch process based on the updated industrial process product characteristics, the first time period, and the optimization time period. In an implementation, the availability of the one or more blending products for the batch process may be represented by N-number of blending products. In such implementation, the current component inventory **402** is employed for one or more batch operations (e.g., a batch process) associated with the one or more blending products, such as the products **406** being blended. Moreover, the blending component characteristics are indicative of availability the current component inventory **402** for blending the products **406** (e.g., intermediate components) to create the one or more output streams **306** of processed materials. In addition, the blending component characteristics are updated based on the updated industrial process product characteristics, the first time period, and the optimization time period. Optionally, updated industrial process product characteristics for the regular gas corresponds to number of current component inventory required from the current component inventory **402** to produce the regular gas within the optimization time period (e.g., within 2 days) based on the received optimization request. Therefore, in such case, the blending component characteristics are beneficial to determine the production of one or more number of blending products from the N-number of blending products based on the updated industrial process product characteristics, based on the first time period (e.g., within 10 days), and also based on the optimization time period (e.g., within 2 days).

[0114] The processor **710** is further configured to optimize the industrial process **602** based on the received optimization request to produce the industrial process product (e.g., to produce the regular gas) to fulfill an order from the one or more orders, such as based on the availability of the one or more blending components for the batch process. Furthermore, the processor **710** is configured to determine if the industrial process product (e.g., the regular gas) for the order of the one or more orders is produced or not. Thereafter, in response to completing the production of the industrial process product for the order of the one or more orders, the processor **710** is configured to determine component spent characteristics associated with the one or more blending products. In an implementation, the processor **710** is firstly configured to determine and identify finished batches (e.g., the discontinuous batch operation subprocess **608**) of the order from the one or more orders by determining the component spent characteristics associated with the one or more blending products. For example, in an embodiment, the processor **710** is configured to determine content of material used by one or more blending components of one or more batch operations associated with the batch operation process. In an example, at least one blended product can be represented by an empty cartoon (as shown in FIG. **7**B), which depicts that the corresponding blending products is blended for producing the industrial process product for the order of the one or more orders. Optionally, the component spent characteristics can be associated with at least two or more than two blending components, which depicts utilization of the corresponding two or more than two blending components to produce the blending products. Beneficially as compared to conventional approaches, the plant wide control and optimization remains unchanged during or after the batch blending.

[0115] In one or more embodiments, the processor **710** is further configured to determine the component spent characteristics associated with the one or more blending products based on industrial process product characteristics for the industrial process product or one or more virtual spent components for the one or more blending products. Moreover, the component spent characteristics are associated with an asynchronous update scheme and/or a virtual content tracker. For example, in an embodiment, the asynchronous update scheme is used to determine content of material used by the one or more blending products of one or more batch operations associated with the production of the industrial process product for the order of the one or more orders. In one or more embodiments, processor **710** is configured to determine component spent characteristics associated with the one or more blending components through the asynchronous update scheme and based on one or more virtual spent components for the one or more blending products. Also, in one or more embodiments, the system **700**A for example, via the processor **710** is configured to initialize the asynchronous update scheme associated with the asynchronous update component **704** and/or the virtual content component **706**. Further, in one or more embodiments, the system **700**A is configured to manage the industrial process **602** (alternatively, one or more portions of the industrial process **602**) based at least on the initialization of the asynchronous update scheme. An exemplary initialization plan is described in more details in accordance with FIG. **7**D of the present disclosure. In an implementation, for the current component inventory **402**, if N denote the total number of blending products (or blends), both in progress and completed, since a reference time To, then for M (M<N) number of physical blenders, a matrix V.sub.s is a collection of N column vectors of the virtual spent components, such as:

[00006] $V_s = v_{s,1} + v_{s,2} + \text{.Math.} + v_{s,N}$

[0116] Each column vector "v.sub.s,i" represents virtual spent components required for blending the i.sup.th blending product in a batch process. Optionally, such blending can be either in progress or has been completed. In an example, a first set of virtual spent components (e.g., v.sub.s,1) from a first column of the matrix V.sub.s are required for blending a 1.sup.st blending product from the products **406**. In another example, a second set of virtual spent components (e.g., v.sub.s,2) from a second column of the matrix V.sub.s are required for blending a 2.sup.nd blending product from the products **406**, and the like. Optionally, one or more set of virtual spent components from one or more columns of the matrix V.sub.s can be utilized for blending the i.sup.th blending product from the products **406** in a batch process for production of the industrial process product for the order of the one or more orders without limiting the scope of the present disclosure. In an example, the one or more virtual spent components are selected from the current component inventory **402** based on the received optimization request, such as to produce the requested industrial process product.

[0117] In one or more embodiments, the processor **710** is further configured to utilize the asynchronous update scheme to determine the component spent characteristics associated with the one or more blending products based on the industrial process product characteristics to produce the industrial process product. In one or more embodiments, the asynchronous update scheme is used by the processor **710** to determine the component spent characteristics based on monitoring an amount of material (e.g., gases, hydrocarbons) included in the one or more blending components, such as material included in the current component inventory **402**. Additionally, or alternatively, in an embodiment, the asynchronous update scheme is used by the processor **710** to determine an amount of material used by the one or more blending products (e.g., the product **406**) of one or more batch operations associated with the production of the industrial process product for the order of the one or more orders. In one or more embodiments, the asynchronous update scheme is used by the processor **710** to determine the component spent characteristics based on the industrial process product characteristics for the industrial process product for the order of the one or more orders. In one or more embodiments, the asynchronous update scheme is used by the processor **710** to determine the component spent characteristics by predicting one or more component spent characteristics (e.g., using machine learning). For example, in one or more embodiments, the asynchronous update scheme is used by the processor **710** to perform a machine learning process with respect to industrial process data related to the industrial process. In certain embodiments, the asynchronous update scheme is used by the processor **710** to determine at least a portion of the component spent characteristics based on relationships, correlations and/or predictions between aspects of industrial process data related to the industrial process.

[0118] In one or more embodiments, the processor **710** is further configured to utilize the virtual content tracker to track the component spent characteristics associated with the one or more blending products based on the optimization time period. In various embodiments, the virtual content tracker is strategically inserted into a continuous control process or an optimization process (e.g., a plant optimizer) for the industrial process **602** or industrial facility to monitor spent inventory for a batch process. In one or more embodiments, the virtual content tracker is a discontinuous system configured for monitoring spent inventory related to a batch process. In one or more embodiments, the virtual content tracker is employed with the asynchronous update scheme that is configured to determine how batch operations or batch productions should be updated when a continuous optimizer (e.g., a plantwide optimizer) receives new input at a pre-specified frequency, or at a user request. Optionally, the virtual content tracker is utilized to simultaneously track the optimization time period as well as the component spent characteristics associated with the current component inventory **402**, such as for the order of the one or more orders. As a result, the virtual content tracker is beneficial to fulfil the demand and/or for completing a task associated with the continuous industrial process based on the received optimization request.

[0119] In one or more embodiments, the processor **710** is further configured to determine the component spent characteristics based on monitoring content of material included in the one or more blending components. In an example, the processor **710** is further configured to monitor content of material (e.g., flammable mixture of gases) included in the current component inventory **402** to produce the regular gas. In another example, the processor **710** is further configured to monitor content of material (e.g., percentage of hydrocarbons) included in the current component inventory **402** to produce the diesel based on the received request. Optionally, the content in a blending component (e.g., a product tank) can be transferred to another blending component when the asynchronous update scheme runs. For example, during the production of the natural gas, the content from the blending component can be transferred to another blending component to fulfil the order.

[0120] The processor **710** is further configured to determine the completion of the production of the industrial process product for the order of the one or more orders. In addition, the processor **710** is configured to determine the utilization of the one or more virtual spent components for blending the i.sup.th blending product from the products **406** in a batch process of the production of the industrial process product for the order and within the optimization time period. For example, the second set of virtual spent components (e.g., v.sub.s,2) from the second column of the matrix V.sub.s are selected (as shown in FIG. **7**A within a box) for blending the 2.sup.nd blending product from the products **406**, which is also selected (as shown within another box), as shown in FIG. **7**A. Thereafter, the processor **710** is configured to optimize the industrial process **602** for blending the 2.sup.nd blending product from the products **406** and to produce the industrial process product. Furthermore, the processor **710** is configured to determine that the 2.sup.nd blending product is blended to produce the industrial process product to fulfill one or more orders within the optimization time period. In an example, the 2.sup.nd blending product can be blended to produce the regular gas. In another example, the 2.sup.nd blending product can be blended to produce diesel, and the like. Thereafter, the processor **710** is configured to determine the utilization of the second set of virtual spent components (e.g., v.sub.s,2) from the second column of the matrix V.sub.s, which are used for blending the 2.sup.nd blending product. Optionally, the processor **710** may be configured to utilize the asynchronous update scheme to determine the utilization of the second set of virtual spent components (e.g., v.sub.s,2) from the second column of the matrix V.sub.s. In an implementation, the asynchronous update scheme can be executed on a schedule or based on demand, such as there is no need to synchronize the asynchronous update scheme with the plantwide control and optimization execution schedule.

[0121] The processor **710** is further configured to update the demand data for the one or more feed products utilized for blending the one or more blending products based on the component spent characteristics. In an implementation, the processor **710** is configured to determine the utilization of the one or more feed products, for example to determine the utilization of the crude oil **302** into the current component inventory **402** (e.g., blending components). In one or more embodiments, utilization of the one or more feed products is measured by determining the one or more blending components, the component spent characteristics, and then back-calculating the product feed products. In an implementation, utilization of the one or more feed products is measured by measuring utilization of each column vector “v.sub.s,i” of the matrix V.sub.s. Optionally, utilization of each column vector “v.sub.s, i” can be measured in at least two different ways, either by measuring and accumulating the blending component that are used for blending the i.sup.th product order or by measuring the volume (or level) in the one or more blending product and then back-calculating the virtual spent components. In an example, the blending component that are used for blending the i.sup.th product order can be measured and accumulated by measuring the utilization of the crude oil **302** into the current component inventory **402** (e.g., blending components). Similarly. The volume (or level) in the one or more blending product can be measured by determining the virtual spent components spent while blending the one or more blending product. For example, if the second set of virtual spent components (e.g., v.sub.s,2) from the second column of the matrix V.sub.s are utilized for blending the 2.sup.nd blending product, then the processor **710** is configured to determine the volume of the products **406** based on the determination of the utilization of the second set of virtual spent components (e.g., v.sub.s,2). Thereafter, the processor **710** is configured to update the demand data for the one or more feed products, such as by updating one or more blending products and by updating the virtual spent component in the matrix V.sub.s. In an implementation, the processor **710** is configured to update the industrial process product produced by deleting the blended orders in a confirmed demand vector (P.sub.m), such as by deleting the 2.sup.nd blending product from the products **406**, as shown in FIG. **7**B by a blank tank number ‘2’. An exemplary implementation of updating the demand data is further shown and described in FIG. **8**B. Similarly, the processor **710** is configured to reset the corresponding spent components, “v.sub.s,i” to zero, such as deleting the second set of virtual spent components (e.g., v.sub.s,2) from the second column of the matrix V.sub.s that are utilized for blending the 2.sup.nd blending product as shown in FIG. **7**B by a blank set of components in second row. By virtue of updating the demand data, the processor **710** provides one or more technical improvements such as, but not limited to, extended functionality for a computing device and/or improving accuracy of data provided to a computing device. In addition, the processor **710** provides an improved product demand horizon without compromising the control and optimization responsiveness of the plantwide control and optimization.

[0122] In one or more embodiments, the processor **710** is further configured to update the demand data for the one or more feed products utilized for blending the one or more blending products based on the component spent characteristics and in response to updating the demand data, the processor **710** is further configured to reset the first time period. In other words, if a batch for the production of the industrial process product for the order of the one or more orders is completed, then the processor **710** is configured to update corresponding downstream inventory and reset the first time period. In an implementation, the processor **710** is configured to utilize the asynchronous update scheme to reset the first time period. By virtue of resetting the first time period, the virtual content tracker associated with the current component inventory **402** is also reset. In an implementation, by virtue of updating the demand data for the one or more feed products, the processor **710** is further configured to revise a future minimum component vector c.sub.m for the plantwide control and optimization, as shown below:

[00007] $c_m = \int_0^{m\text{ - Ets}} G_F * u\,\mathrm{dt} + V_0 + V_s = \int_t^{T_s} G_F * u\,\mathrm{dt} + V_0 + V_s$ [0123] where E.sub.ts corresponds to elapsed time since the last asynchronous update scheme run, and T.sub.s is the demand end time (i.e., optimal components cm will be produced by Ts).

[0124] In one or more embodiments, the processor **710** is further configured to update the demand data by updating one or more portions of the demand data in response to a change in demand for the industrial process product. For example, if an updated optimization request is received with the change in demand for the industrial process product (e.g., the current component inventory **402** and/or the products **406** being blended), then the processor **710** is configured to utilize the virtual content component **706** to update one or more portions of the demand data based on the updated optimization request. In an implementation, if there is change in demand for the industrial process product (i.e., any confirmed demand is changed since the last asynchronous update scheme), then the processor **710** is configured to update the confirmed demand vector (P.sub.m) to include the new/updated orders. Thereafter, the processor **710** is configured to reset the elapsed time (E.sub.ts) to zero for remaining one or more orders and set the first time period (T.sub.s*) to a new value for remaining one or more orders, which is different from the previous first time period.

[0125] In an implementation, the processor **710** is further configured to update the demand data by updating the confirmed demand vector (P.sub.m), the component vector c.sub.m, and component mass balance equation in the plantwide control and optimization formulation. In an example, the confirmed demand vector (P.sub.m) is updated and revised to P.sub.m1=((T.sub.s1−T.sub.0)/m.sub.0)*P.sub.m, where m.sub.0=T.sub.s−T.sub.0, where T.sub.s1 is the end of optimization horizon (OH) at time t. In an implementation, if the processor **710** is configured to track the spent components from the matrix V.sub.s for the production of the industrial process product for an order of the one or more orders, then the processor **710** is configured to update the component vector c.sub.m to c.sub.m1 and also update component mass balance equation as shown below:

[00008]
$$R.p_{m1}^{\mathrm{before}} = R.p_{m1}^{\mathrm{after}} + v_{s,2} \le \int_0^{\mathrm{OH}} G_F\,\mathrm{udt} + V_0 + V_s^{\mathrm{after}} + v_{s,2}$$
$$R.p_{m1}^{\mathrm{after}} \le c_{m1} = \int_0^{\mathrm{OH}} G_{\mathrm{Fudt}} + V_o + V_s^{\mathrm{after}}$$

[0126] Where, OH=(t, T.sub.s1), u=conjoint MVs, and R denote a recipe matrix:

[00009]
$$R = \begin{bmatrix} r_{1,1} & r_{1,2} & .Math. & r_{1,np} \\ r_{2,1} & r_{2,2} & .Math. & r_{2,np} \\ .Math. & & & .Math. \\ r_{nc,1} & r_{nc,2} & .Math. & r_{nc,np} \end{bmatrix}$$
$$\sum_{i=1}^{nc} r_{i,j} \equiv 1$$

[0127] The i-th column in the R is a fractional recipe for blending the i-th product.
[0128] Moreover, the updated component mass balance equation (i.e., after blending) is same as that of an original equation (i.e., before blending). Furthermore, the matrix V.sub.s is used to track each blend that has been used by the blending components, thus the matrix V.sub.s can be used to track both the blend amount and corresponding recipe deviations. Furthermore, the remainder of the spent components become “reusable” components for the future demand, which may potentially introduce a small bias to the solution.
[0129] In another implementation, if the processor **710** is configured to track the blended products to produce the industrial process product for an order of the one or more orders, then the processor **710** is configured to update the component vector c.sub.m to c.sub.m2 and update the component mass balance equation as shown below:
[00010]
$$R(p_{m1}^{after} - \mathrm{sum}(V_s^{after})^T) = R(p_{m1}^{after} - \mathrm{sum}(V_s^{after})^T) \le c_{m2} = \int_0^{OH} G_F\,u\,dt + V_0$$
$$R\,.\ .Math.\ p_{m1} - \mathrm{sum}(V_s)^T\ .Math.\ \le c_{m2} = \int_0^{OH} G_F\,u\,dt + V_0$$
[0130] Here, to prevent any reverse flows (c.sub.m2<0). Furthermore, ColSum (V.sub.s) is used by the processor **710** to track all blends, both in-progress and completed since the last asynchronous update scheme run, which are then excluded from the plantwide control and optimization. Beneficially, the processor **710** can remove the possibility of a small bias that could potentially be introduced in previous equation related to tracking spent components. Moreover, in such implementation, if any in-progress blend is canceled, then the processor **710** can update the confirmed demand vector (i.e., P.sub.m1) and reuse the affected one as a starting heel for a future blend of the same/similar grade. In an example, if production disturbance is too noisy, then such implementation can be implemented to buffer the disturbances before rejecting the products.
[0131] In such implementation, when the processor **710** is configured to track the blended products to produce the industrial process product for an order of the one or more orders, then the confirmed demand vector (P.sub.m) can be updated and revised from P.sub.m to P.sub.m2. Moreover, the processor **710** is configured to update the component vector c.sub.m to c.sub.m2 and update component mass balance equation more precisely, as shown below:
[00011]
$$R\,.\,p_{m2} \le c_{m2} = \int_0^{OH} G_F\,u\,dt + V_0$$
$$p_{m2} = \frac{OH}{T_s - t}\ .Math.\ .Math.\ p_m - \mathrm{sum}(V_s)^T\ .Math.\ = \frac{T_{S1} - t}{T_s - t}\ .Math.\ .Math.\ p_m - \mathrm{sum}(V_s)^T\ .Math.$$
[0132] Here, P.sub.m2=[OH/(T.sub.s−t)] *abs (P.sub.m−sum (V.sub.s)′). Moreover, in such implementation, an if the accumulative number of blended products in elapsed window (T.sub.0, t), is different from an “expected amount”, due to disturbances, then an auto production catch-up in the remaining time window (t, T.sub.s), which is beneficial to reject undesired disturbance.
[0133] The processor **710** is further configured to transmit a control signal to control the flow of the feed component in the continuous process. The transmitted control signal provides one or more technical improvements such as, but not limited to, providing a varied experience for a computing device. In one or more embodiments, the processor **710** is further configured to generate one or more batch blending recipes based on the updated demand data. In various embodiments, the

management of the industrial process optimization provides one or more optimal batch blending recipes (e.g., optimal recipes, optimal batch-blending plantwide recipes, etc.) for the industrial process.

[0134] The processor **710** of the system **700**A is used for controlling the industrial process with improved product demand horizon without compromising the control and optimization responsiveness of the plantwide control and optimization. The processor **710** is beneficial to control the plantwide control and optimization to operate with a time horizon suitable for mitigating different types of disturbances during the production of the industrial process product to fulfill one or more orders. Therefore, the processor **710** liberates the plantwide control and optimization from the constraint of the product demand horizon, transforming the plantwide control and optimization into an improved system, which can be used for batch blending oil refineries.

[0135] FIG. **7**C illustrates a user interface rendering demand data, in accordance with one or more embodiments described herein. An example demand data **700**C is rendered via user interface of a computing device (which is described in accordance with FIG. **7** of the current disclosure). The demand data **700**C, in some embodiments, corresponds to demand data described in accordance with FIG. **7** of the current disclosure. The demand data **700**C described herein comprises one or more fields such as, but not limited to order # or order identifier **702**C, product **704**C, order name **706**C, target volume **708**C, heel volume **710**C, heel properties **712**C, status **714**C, and/or order updated details **716**C. In one or more embodiments, the order identifier **702**C corresponds to number or any identifier that identifies an order (or a request) to produce one or more industrial process products. Further, in one or more embodiments, the product **704**C corresponds to the one or more industrial process products that is to be produced by the industrial process **602** based on the corresponding order identifier **702**C. Also, in some embodiments, the product **704**C corresponds to one or more output streams **306** which is produced using one or more blending components. And, in one or more embodiments, the order name **706**C of the demand data **700**C comprises details of, for example, a customer or a user who places an order for the corresponding product **704**C. Further, in one or more embodiments, the target volume **708**C corresponds to a volume or a quantity of the product **704**C that is to be produced by the industrial process **602**. In this regard, in some embodiments, the target volume **708**C is indicated by customer or user as in the order name **706**C who places an order for the corresponding product **704**C. Furthermore, in one or more embodiments, the demand data **700**C comprises the heel volume **710**C that indicates a volume (or a quantity) of one or more blending components in current component inventory **402**. Also, in some embodiments, the heel volume **710**C indicates an inventory level for one or more feed products associated with the one or more blending components. Then, in one or more embodiments, the heel properties **712**C of the demand data **700**C indicates one or more properties of the heel volume **710**C.

[0136] Furthermore, in one or more embodiments, the status **714**C indicates a status of an order associated with corresponding order # or order identifier **702**C. In some embodiments, the status **714**C can be indicated based on a time and a date of receipt of an order and based on amount of material in the current component inventory **402** and/or amount of material in products **406** being blended. Then, in one or more embodiments, the order updated details **716**C of the demand data **700**C comprises a time and a date (or timestamp) at which the status **714**C of a corresponding order was last updated. For example, in response to receipt of an order as a fresh order, the order updated details **716**C comprises a timestamp at which the order has been received and the status **714**C can correspond to “new”. Further, in the same example, based on amount of material in the current component inventory **402** and/or amount of material in products **406** being blended for production of industrial process product of the order, the status **714**C can be updated to, for instance as “in progress”. In this regard, a timestamp at which the status **714**C of the order is moved to “in progress” is updated in the order updated details **716**C. Then, in the same example, upon completing production of industrial process product of the order, the status **714**C can be updated to, for instance as "completed". In this regard, a timestamp at which the status **714**C of the order is moved to "completed" is updated in the order updated details **716**C. Yet in one or more embodiments, at least some statuses of the status **714**C can be indicated by a tracking flag to indicate a status of an order. Accordingly, in one or more embodiments, the demand data **700**C described herein provides relevant information of one or more orders that is associated with production of one or more industrial process products. In this regard, the demand data **700**C is utilized for instance, by processor **710** or control component **708** to initialize an asynchronous update scheme which is described in more details in accordance with FIG. **7**D of the current disclosure. It is to be understood that one or more values represented in the demand data **700**C described herein are exemplary only and the values are not limited to the values represented in the demand data **700**C.

[0137] FIG. **7**D illustrates an initialization plan for initializing an asynchronous update scheme, in accordance with one or more embodiments described herein. In one or more embodiments, the exemplary initialization plan **700**D is utilized by system **400** or industrial process computer system **702** to initialize the asynchronous update scheme (which is described in accordance with, for instance, FIG. **4**, FIG. **7**, and FIG. **7**A of the current disclosure). In addition to the initialization plan **700**D, the system **400** or industrial process computer system **702** also utilizes demand data **700**C to initialize the asynchronous update scheme. The initialization plan **700**D described herein comprises one or more fields such as, but not limited to flag **702**D, one or more cases **704**D, order origin **706**D, one or more notifications **708**D, and/or optimizer state **710**D. In one or more embodiments, the flag **702**D corresponds to a tracking flag. In this regard, the flag **702**D indicates a status of an order of one or more orders which are associated with production of one or more industrial process products. Also, the status indicated by the flag **702**D is based on the amount of material in the current component inventory **402** and/or the amount of material in the products **406** being blended for production of an industrial process product indicated by the corresponding order(s). In some embodiments, the amount of material in the current component inventory **402** and/or the amount of material in the products **406** being blended is determined using real-time industrial process data. Accordingly, based on the amount of material in the current component inventory **402** and/or the amount of material in the products **406** being blended, the flag **702**D is dynamically updated. In this regard, the flag **702**D is set to an appropriate value to indicate a relevant status of the one or more orders. For example, upon beginning a blending operation for an order (say, a first order) to produce an industrial process product, a first status is assigned to the order if, for instance, an amount of material in product(s) being blended lies between 1-99% of an amount of material in a completed or finished industrial process product. And the flag **702**D is set to a first value. Then, upon completing the blending operation for the order, a second status is assigned to the order if, for instance, an amount of material in product(s) being blended is equal to an amount of material in a completed or finished industrial process product. And the flag **702**D is then set to a second value. Per this aspect, the first status corresponds to, for example, "in media res" meaning that the blending operation for production of industrial process product of the order is in progress and the first value corresponds to 0.5. Whereas, the second status corresponds to, for example, "ab initio" meaning that the blending operation for production of industrial process product of the order is completed and the second value corresponds to 1. Also, the amount of material in a completed or finished industrial process product can correspond to target volume **708**C. It is to be understood that the statuses and values described herein are exemplary only and there can be other intermediate status(es) and value(s) which can be added or modified by say, an operator in the process plant based on one or more requirements in the process plant. Also, in one or more embodiments, the value to which the flag **702**D is set to along with the relevant status is renderable on a user interface as well. For example, for orders where blending operation for production of industrial process products is in progress, the status "in media res" along with the first value of 0.5 is rendered against, for instance, order identifier **702**C or status **714**C of the corresponding orders on the user interface. Also, in some embodiments, a value to which the flag **702**D is set to can also be indicated by one or more icons and/or other identifiers as well.

[0138] In one or more embodiments, based on the order origin **706**D and the flag **702**D, the one or more cases **704**D are defined. The one or more cases **704**D facilitate initialization and utilization of the asynchronous update scheme without any interruptions/bumps and allow the control component **708** to generate relevant control data **722** as well. The one or more cases **704**D described herein are defined based on one or more situations observed in the process plant. In one or more embodiments, the order origin **706**D described herein indicates an origin or a source of an order of the one or more orders. In this regard, the order origin **706**D described herein can correspond to either “demand data” or “orphan order”. Said alternatively, if one or more details of an order of the one or more orders is present in the demand data **700**C, then the order origin **706**D corresponds to “demand data” for the order. Whereas, if one or more details of an order of the one or more orders is absent in the demand data **700**C, then the order origin **706**D corresponds to “orphan order” for the order. For instance, at times, it may so happen that an order may be received as an ad hoc request or an instant order for which an operator may start a blending operation without updating details of the order in the demand data **700**C. In another instance, an order may not be synchronized with existing orders in the demand data **700**C. Also, in another instance, the optimizer controller **502** can be in OFF state due to downtime, maintenance operations, etc. As a result of this, an operator may be unable to update details of an order in the demand data **700**C or industrial process data may go unobserved. Yet in another instance, the optimizer controller **502** (alternatively, plantwide optimization solution) can be newly installed in the process plant such that the optimizer controller **502** has no history or data associated with one or more blending operations in the process plant. Accordingly, one or more orders received during the aforementioned exemplary situations serve as “orphan orders”. Also, it is to be understood that the aforementioned instances are exemplary only and there can be other similar instances that are associated with “orphan orders” as well.

[0139] Further, in one or more embodiments, based on the flag **702**D and the order origin **706**D, the one or more notifications **708**D as appropriate are determined for the corresponding cases **704**D. For example, for a case of the one or more cases **704**D, if the flag **702**D is set to a value of 0.5 and the order origin **706**D corresponds to “demand data” for an order, then the one or more notifications **708**D correspond to providing an alert or an action, for example, to an operator to turn the optimizer controller **502** to WARM state only and wait for a current batch or order to be completed. In another example, for a case of the one or more cases **704**D, if the flag **702**D is set to a value of 0.5 and the order origin **706**D corresponds to “orphan order” for an order, then the one or more notifications **708**D correspond to providing an alert or an action, for example, to an operator to turn the optimizer controller **502** to WARM state only and wait for a current batch or order to be completed. In WARM state, the optimizer controller **502** is configured to collect data related to one or more operations associated with the industrial process **602**. Further, in another example, for a case of the one or more cases **704**D, if the flag **702**D is set to a value of 1 and the order origin **706**D corresponds to “demand data” for an order, then the one or more notifications **708**D correspond to providing an alert or an action, for example, to an operator to turn the optimizer controller **502** to ON state. Yet in another example, if the flag **702**D is set to a value of 1 and the order origin **706**D corresponds to “orphan order” for an order, then the one or more notifications **708**D correspond to providing an alert or an action, for example, to an operator to: update details of the orphan order in the demand data **700**C and turn the optimizer controller **502** to ON state. In ON state, the optimizer controller **502** is configured to optimize one or more operations associated with the industrial process **602** in addition to collection of data associated with one or more operations.

[0140] Also, in one or more embodiments, the one or more notifications **708**D are rendered on the user interface (as visual notifications) to provide an alert or an action, for example, to an operator in the process plant. Per this aspect, the operator can undertake appropriate actions to facilitate smooth operations in the process plant. In some embodiments, the one or more notifications **708**D can correspond to audio notifications as well. Further, in some embodiments, the one or more notifications **708**D as appropriate are rendered on the user interface upon receiving an optimization request at the industrial process computer system **702** as well. In this regard, the optimization request corresponds to a request to optimize an industrial process (e.g., the industrial process **602** that produces one or more industrial process products) with respect to an order having the corresponding order #**702**C. For example, in an embodiment, when an order (say, a first order) is in progress (or is being produced by the industrial process **602**), if an optimization request is received with respect to another order (say, a second order) which is newly received and for which blending operation is to be initiated, then the one or more notifications **708**D (say, first notifications) rendered correspond to providing an alert or an action, for example, to an operator to turn the optimizer controller **502** to WARM state only and wait for a current batch or the first order to be completed. Then, in the same embodiment, when the first order is completed, the one or more notifications **708**D (say, second notifications) rendered correspond to providing an alert or an action, for example, to an operator to turn the optimizer controller **502** to ON state and/or to update details of the first order (if the first order is an “orphan order”) in the demand data **700**C. In this regard, the industrial process computer system **702** updates the demand data **700**C based on the details of the first order, component spent characteristics, and/or the optimization request. Further, in this regard, the industrial process computer system **702** transmits a control signal **722** based on the demand data **700**C to the optimizer controller **502** to optimize the industrial process **602** with respect to another order i.e., the second order. In this regard, the industrial process **602** may be optimized as described in accordance with, for instance, FIG. **4**, FIG. **7**, and FIG. **7**A of the current disclosure. Further, in one or more embodiments, the initialization plan **700**D comprises optimizer state **710**D that indicates an operational status of the optimizer controller **502**. The optimizer state **710**D described herein is based at least on the flag **702**D. In this regard, the optimizer state **710**D corresponds to “WARM” state if the flag **702**D is set to a value of 0.5 and “ON” state if the flag **702**D is set to a value of 1. Also, in one or more embodiments, the optimizer state **710**D is rendered on the user interface as well. Thus, irrespective of whether the order origin **706**D corresponds to “demand data” or “orphan order”, the initialization plan **700**D facilitates initialization and utilization of the asynchronous update scheme without any interruptions/bumps.

[0141] FIG. **8**A illustrates a graphical representation **800**A that depicts an initial industrial demand vector for m-number of days, in accordance with one or more embodiments described herein. With reference to FIG. **8**A there is shown the graphical representation **800**A that includes an X-axis **802** that depicts an industrial product type and a Y-axis **804** that depicts a corresponding confirmed demand vector (P.sub.m) required to be produced in m-number of days. For example, a first bar **806**A depicts a minimum and a maximum amount of a confirmed demand vector (P.sub.m) for a regular gas (in barrels) that is required to be produced in m-number of days. There is further shown a delta value of a confirmed demand vector (Δp.sub.1) that is required to be produced based on an optimization requested received by the processor **710**. Similarly, a second bar **808** depicts a minimum and a maximum amount of another confirmed demand vector (P.sub.m) for a premium gas (in barrels) that is required to be produced in m-number of days. In addition, a third bar **810** depicts a minimum and a maximum amount of yet another confirmed demand vector (P.sub.m) for a jet oil (in barrels) that is required to be produced in m-number of days. Furthermore, a fourth bar **812** depicts a minimum and a maximum amount of another confirmed demand vector (P.sub.m) for diesel (in barrels) that is required to be produced in m-number of days. Similarly, the graphical representation **800**A can includes one or more bars to depicts multiple industrial products to be produced in m-number of days. Optionally, each bar could have one or more multiple industrial products to be produced after completion of m-number of days.

[0142] FIG. **8**B illustrates a graphical representation **800**B that depicts an updated demand vector, in accordance with one or more embodiments described herein. With reference to FIG. **8**B there is shown the graphical representation **800**B that includes the X-axis **802** that depicts an industrial product type and the Y-axis **804** that depicts a corresponding updated demand vector (P.sub.m) required to be produced in m-number of days. In an implementation, the processor **710** is configured to determine if the production of the industrial process product for an order of the one or more orders is completed or not based on the received optimization request. Furthermore, if the production of the industrial process product for an order of the one or more orders completed, then the processor **710** is configured to update the demand data for the one or more feed products utilized for blending the one or more blending products based on the component spent characteristics. In an implementation, the processor **710** is configured to update the demand data by updating the confirmed demand vector (P.sub.m) for the corresponding industrial process product. For example, the processor **710** is configured to determine that the regular gas is produced for the order of the one or more orders is completed based on the received optimization request. Thereafter, the processor **710** is configured to update the corresponding demand vector (P.sub.m), as shown by a bar **806**B that depicts an updated minimum and demand vector (P.sub.m) maximum amount of the updated demand vector (P.sub.m) for the regular gas (in barrels) that is required to be produced in m-number of days. The bar **806**B is different from the first bar **806**A.

[0143] FIG. **9** illustrates a system **900** according to one or more described features of one or more embodiments of the disclosure. The system **900** includes an enterprise primary controller **902** and a set of systems **500***a-n*. Each system from the set of systems **500***a-n* corresponds to the system **500**. For example, each system from the set of systems **500***a-n* includes the optimizer controller **502** and the one or more multivariable MPC controllers **504***a-n*. In an embodiment, the system **900** is a tier-3 enterprise-wide optimizer for a supply chain associated with one or more industrial processes. For example, in one or more embodiments, the system **900** is employed to optimize transfer of material between industrial plants where respective industrial plants are associated with respective a respective system **500** from the set of systems **500***a-n*. For example, in an embodiment, the system **500***a* is associated with a first industrial plant, the system **500***b* is associated with a second industrial plant, and the system **500***n* is associated with an nth industrial plant. In certain embodiments, the system **900** is employed to improve coordination between multiple refineries and/or multiple fuel/crude depots within a network. In certain embodiments, the system **900** is employed to increase supply chain agility, optimize inventories within an enterprise, and/or reduce inventories within an enterprise. In certain embodiments, the system **900** is employed to optimize transfer or cross-shipping of intermediate materials/parts between industrial plants. In one or more embodiments, the enterprise primary controller **902** is configured in a similar manner as the optimizer controller **502**.

[0144] FIG. **10** illustrates a system **1000** according to one or more described features of one or more embodiments of the disclosure. In one or more embodiments, the system **1000** provides an enterprise supply chain with transport links. In one or more embodiments, the system **1000** provides tracking of inventory-in-transit. The system **1000** includes an industrial plant **1002** and an industrial plant **1004**. For T shipments/transfers, V.sub.s is comprised of T columns of Inventory-in-Transit:

[00012] $V_s = V_{s,1} + V_{s,2} + \quad + V_{s,T}$ [0145] where each column V.sub.s,i represents the Inventory-in-Transit for the i-th shipment/transfer between the industrial plant **1002** and the industrial plant **1004**. In one or more embodiments, this shipment/transfer between the industrial plant **1002** and the industrial plant **1004** is either on route or has been received. If any shipment is received from the industrial plant **1002**, an asynchronous updating scheme associated with the inventory-in-transit adds the shipment associated with the industrial plant **1002** to the industrial plant **1004** inventory, subtracts the shipment associated with the industrial plant **1002** from the future demand P.sub.m, and reset a corresponding V.sub.s, i to zero. If there is any newly confirmed future demand (e.g., since a previous asynchronous updating scheme), the new demand is added to P.sub.m.

[0146] FIG. **11** illustrates a method **1100** for managing industrial process optimization related to batch operations, in accordance with one or more embodiments described herein. The method **1100** is associated with the industrial process computer system **702**, for example. For instance, in one or more embodiments, the method **1100** is executed at a device (e.g., the industrial process computer system **702**) with one or more processors and a memory. In one or more embodiments, the method **1100** facilitates asynchronous management of an industrial process **602** by monitoring product inventory provided by an optimization subprocess and component spent characteristics of an industrial process **602** product provided by a batch operation subprocess. In one or more embodiments, the method **1100** begins at block **1102** that receives (e.g., by the asynchronous update component **704**) an optimization request to optimize an industrial process **602** that produces an industrial process product. The optimization request to optimize the industrial process **602** provides one or more technical improvements such as, but not limited to, facilitating interaction with a computing device, extended functionality for a computing device and/or improving accuracy of data provided to a computing device.
[0147] At block **1104**, it is determined whether the optimization request is processed. If no, block **1104** is repeated to determine whether the optimization request is processed. If yes, the method **1100** proceeds to block **1106**. In response to the optimization request, block **1106** determines (e.g., by the asynchronous update component **704**) component spent characteristics for one or more blending components of a batch operation subprocess. The determining the component spent characteristics provides one or more technical improvements such as, but not limited to, extended functionality for a computing device and/or improving accuracy of data provided to a computing device. In certain embodiments, the determining the one or more setpoints comprises determining the one or more setpoints from asset strategy data associated with an operation strategy for the asset.
[0148] The method **1100** also includes a block **1108** that, based on the component spent characteristics and inventory data indicative of an inventory level for one or more feed products associated with the one or more blending components, updates (e.g., by the asynchronous update component **704**) demand data for the one or more feed products. The updating the demand data provides one or more technical improvements such as, but not limited to, extended functionality for a computing device and/or improving accuracy of data provided to a computing device.
[0149] The method **1100** also includes a block **1110** that transmits (e.g., by the control component **708**) a control signal configured based on the demand data to a controller configured for optimization associated with the industrial process **602** that produces the industrial process product. The transmitting the control signal provides one or more technical improvements such as, but not limited to, providing a varied experience for a computing device.
[0150] In certain embodiments, the receiving the optimization request comprising receiving the inventory data via the optimization request. In certain embodiments, the receiving the optimization request comprising receiving the optimization request from the controller. In certain embodiments, the receiving the optimization request comprising receiving the optimization request in response to a determination that time data for the industrial process **602** satisfies a defined criterion. In certain embodiments, the receiving the optimization request comprising receiving the optimization request in response to a determination that a condition associated with the industrial process **602** satisfies a defined criterion. In certain embodiments, the receiving the optimization request comprising receiving the optimization request from a user device.
[0151] In certain embodiments, the determining the component spent characteristics comprising determining the component spent characteristics based on industrial process product characteristics for the industrial process product. In certain embodiments, the determining the component spent characteristics comprising monitoring content of material included in the one or more blending components. In certain embodiments, the determining the component spent characteristics comprising monitoring an amount of material included in the one or more blending components. In

certain embodiments, the determining the component spent characteristics comprising determining the component spent characteristics based on one or more virtual spent components for the one or more blending components.
[0152] In certain embodiments, the method **1100** additionally or alternatively includes generating one or more batch blending recipes based on the demand data. In certain embodiments, the method **1100** additionally or alternatively includes updating one or more portions of the inventory data in response to determining that the batch operation subprocess satisfies a defined criterion. In certain embodiments, the method **1100** additionally or alternatively includes updating one or more portions of the component spent characteristics in response to determining that the batch operation subprocess satisfies a defined criterion. In certain embodiments, the method **1100** additionally or alternatively includes updating one or more portions of the demand data in response to a change in demand for the industrial process product. In certain embodiments, the method **1100** additionally or alternatively includes tuning an interval of time for the optimization associated with the industrial process **602** based on the demand data.
[0153] FIG. **12** illustrates a flow chart for a method **1200** for controlling an industrial process, in accordance with one or more embodiments described herein. The method **1200** is associated with the industrial process computer system **702**, for example. For instance, in one or more embodiments, the method **1200** for controlling an industrial process is executed at a device (e.g., the industrial process computer system **702**) with the processor **710** and the memory **712** comprising one or more instructions.
[0154] In one or more embodiments, the method **1200** begins by the asynchronous update component **704**, where an optimization request is received to optimize the industrial process **602** that produces an industrial process product. The optimization request to optimize the industrial process **602** provides one or more technical improvements such as, but not limited to, facilitating interaction with a computing device, extended functionality for a computing device and/or improving accuracy of data provided to a computing device.
[0155] In one or more implementation, the method **1200** comprises, receiving an optimization request in response to a determination that time data for the industrial process satisfies a defined criterion. In an implementation, the method **1200** comprises, determining, by the processor **710**, the time data (e.g., two days) for optimizing the industrial process **602** and for producing the industrial process product. Thereafter, if the time data (e.g., two days) for the industrial process **602** satisfies the defined criterion, then the method **1200** comprises, receiving the optimization request, by the processor **710**, for optimizing the industrial process **602** (e.g., a schedule for the industrial process **602**) that produces the industrial process product, such as to fulfill at least one order from the one or more orders. Optionally, the optimization request may be received from a user device.
[0156] At block **1202**, the method **1200** comprises, determining a first time period to produce the industrial process product to fulfill one or more orders. In an implementation, the first time period corresponds to a total time that is required for producing the industrial process product through the batch operation to fulfill the one or more orders. In one example, the batch operation is performed to blend different blending components (e.g., the current component inventory **402**) to form a refinery product, such as the products **406**. However, the batch operation can disrupt one or more other industrial processes and/or one or more optimization operations related to a processing facility. For example, the batch operation can be performed with the first time period that includes an on-and-off functionality associated with a starting time and/or duration that is generally not predictable. Optionally, the starting time may also be referred to as a time at which a last asynchronous update scheme was performed, and the duration that is generally not predictable may also be referred to as a demand end time of the batch operation. Optionally, the first time period may be represented by T.sub.1st (or m.sub.0), such as a time at which a last asynchronous update scheme was performed may be represented by T.sub.0, and the demand end time may be represented by T.sub.s for the batch operation. In an example, the first time period referred to as a

demand window, which can vary based on the one or more orders for the batch operation, such as can vary from 10 days to 12 days, 10 days to 15 days, 12 days to 16 days, and the like. Furthermore, the batch operation can cause one or more other industrial processes and/or one or more optimization operations related to a processing facility to be performed in an inefficient manner (e.g., a cyclic manner) with respect to operation of the processing facility. It is therefore advantageous to optimize the batch operation with respect to one or more other industrial processes and/or one or more optimization operations related to a processing facility to achieve optimal performance and/or optimal efficiency for one or more industrial processes.

[0157] At block **1204**, the method **1200** comprises, determining an optimization time period, wherein the optimization time period is less than the first time period. In other words, in response to the received optimization request, the method **1200** comprises, determining, by the processor **710**, the optimization time period for optimizing the industrial process **602** for producing the industrial process product, such as to fulfill at least one order from the one or more orders. In an example, the optimization time period may be represented by optimization horizon (OH), which can be set (i.e., can be increased or decreased) by a user. Moreover, the optimization time period is used by the processor **710** for producing the industrial process product based on the received optimization request to fulfill at least one order from the one or more orders. For example, if the optimization request is received by the processor **710** for producing one barrel of regular gas to fulfill at least one order, then the method **1200** comprises, determining, by the processor **710**, the optimization time period for producing the one barrel of the regular gas to fulfill the at least one order. Therefore, the optimization time period (i.e., OH or m.sub.1) may also be referred to as a desired demand window to fulfil the received optimization request. Furthermore, the optimization time period is less than the first time period. For example, a time period between a time ‘t’ and a time T.sub.s1 corresponds to the optimization time period (i.e., OH=T.sub.s1−t) or the desired demand window. Moreover, if the first time period between the last asynchronous update scheme To and the demand end time T.sub.s is for sixteen days, then the optimization time period between the time ‘t’ and the time T.sub.s1 can be for two days, which is less than the first time period. In addition, a time period between the demand end time T.sub.s and the time ‘t’ corresponds to a remaining demand window to fulfill remaining orders. In an example, if the first time period (i.e., the demand window) to fulfill the one or more orders is for 12 days, the optimization time period (i.e., the desired demand window) to fulfill the at least one order is for 2 days, then the remaining demand window to fulfil the remaining orders is for 10 days. In one or more implementation, if all the industrial product orders based on the received optimization request within the first time period are confirmed, then the first time period (i.e., T.sub.0, T.sub.s) is defined as a confirmed period. Furthermore, the method **1200** comprises, selecting, by the processor **710**, a maximum (or sufficiently large) value for the T.sub.s(m), such as the first time period (i.e., T.sub.0, T.sub.s) remains confirmed to fulfill the one or more orders. Optionally, a user can enter all the confirmed orders to maximize the T.sub.s. Therefore, at any time, ‘t’, the method **1200** can be used for performing plantwide control and optimization (PWO), which can have infinite options of corresponding optimization horizon (OH)=(t, T.sub.s1), as long as T.sub.s1≤T.sub.s.

[0158] At block **1206**, the method **1200** comprises, verifying if the optimization time period is determined or not. In an example, if the optimization time period is not determined, then the method **1200** comprises repeating the block **1204**. However, if the optimization time period is determined, then at block **1208**, the method **1200** comprises, updating, in response to determining the optimization time period, industrial process product characteristics based on at least the first time period and the optimization time period. In an implementation, the industrial process product characteristics are related to the one or more output streams **306** of processed materials. For example, the industrial process product characteristics are associated with an amount of production of at least one or more of: typical refinery products, regular gas, refinery fuel gas, liquid petroleum gas (LPG), regular gasoline, premium gasoline, solvents, aviation fuels, diesels, heating oils, lube oils, greases, asphalts, industrial fuels, refinery fuel oil, and the like. In an example, the method **1200** comprises, updating the industrial process product characteristics for the regular gas based on the first time period and the optimization time period based on the current component inventory **402**. Optionally, updating the industrial process product characteristics for the regular gas corresponds to determination of total amount of production of the regular gas within the first time period (e.g., within 10 days) and required production of the regular gas within the optimization time period (e.g., within 2 days) based on the received optimization request. Similarly, the industrial process product characteristics can be updated for other output streams based on the received optimization request and based on the first time period and the optimization time period.

[0159] At block **1210**, the method **1200** comprises, updating, in response to determining the optimization time period, blending component characteristics indicative of availability of one or more blending components for the batch process based on the updated industrial process product characteristics, the first time period, and the optimization time period. In an implementation, the availability of one or more blending products for the batch process may be represented by N-number of blending products. In such implementation, the current component inventory **402** is employed for one or more batch operations (e.g., a batch process) associated with the one or more blending products, such as the products **406** being blended. Moreover, the blending component characteristics are indicative of availability the current component inventory **402** for blending the products **406** (e.g., intermediate components) to create the one or more output streams **306** of processed materials. In addition, the blending component characteristics are updated based on the updated industrial process product characteristics, the first time period, and the optimization time period. Optionally, updated industrial process product characteristics for the regular gas corresponds to number of current component inventory required from the current component inventory **402** for producing the regular gas within the optimization time period (e.g., within 2 days) based on the received optimization request. Therefore, in such case, the blending component characteristics are beneficial to determine the production of one or more number of blending products from the N-number of blending products based on the updated industrial process product characteristics, based on the first time period (e.g., within 10 days), and also based on the optimization time period (e.g., within 2 days).

[0160] The method **1200** further comprises, optimizing, by the processor **710**, the industrial process **602** based on the received optimization request for producing the industrial process product (e.g., to produce the regular gas) to fulfill an order from the one or more orders, such as based on the availability of the one or more blending components for the batch process. Thereafter, at block **1212**, the method **1200** comprises, determining, by the processor **710**, if the production of the industrial process product for an order of the one or more orders is completed or not. In an implementation, if the production is not completed, then the block **1212** is repeated. However, if the production of the industrial process product for an order of the one or more orders, then at block **1214**, in response to completing the production of the industrial process product for an order of the one or more orders, the method **1200** comprises, determining, by the processor, component spent characteristics associated with the one or more blending products. In an implementation, the method **1200** comprises, determining and identifying, by the processor, finished batches (e.g., the discontinuous batch operation subprocess **608**) of the order from the one or more orders by determining the component spent characteristics associated with the one or more blending products. For example, in an embodiment, the method **1200** comprises, determining, by the processor **710**, content of material used by one or more blending components of one or more batch operations associated with the batch operation process. In an example, at least one blended product can be represented by an empty cartoon (as shown in FIG. 7B), which depicts that the corresponding blending products is blended for producing the industrial process product for the order of the one or more orders. Optionally, the component spent characteristics can be associated with at least two or more than two blending components, which depicts utilization of the corresponding two or more than two blending components to produce the blending products. Beneficially as compared to conventional approaches, the plant wide control and optimization remains unchanged during or after the batch blending.

[0161] In one or more embodiments, determining the component spent characteristics includes determining the component spent characteristics based on industrial process product characteristics for the industrial process product or one or more virtual spent components for the one or more blending products. Moreover, the one or more virtual spent components are associated with an asynchronous update scheme and/or a virtual content tracker. For example, in an embodiment, the asynchronous update scheme is used for determining the content of material used by the one or more blending products of one or more batch operations associated with the production of the industrial process product for the order of the one or more orders. In one or more embodiments, the method **1200** comprises, determining, by the processor **710**, component spent characteristics associated with the one or more blending components through the asynchronous update scheme and based on one or more virtual spent components for the one or more blending products. In an implementation, for the current component inventory **402**, if N denote the total number of blending products (or blends), both in progress and completed, since a reference time To, then for M (M<N) number of physical blenders, a matrix V.sub.s is a collection of N column vectors of the virtual spent components, as shown below:

[00013] $V_s = v_{s,1} + v_{s,2} + \text{.Math.} + v_{s,N}$

[0162] Each column vector "v.sub.s,1" represents virtual spent components required for blending the i.sup.th blending product in a batch process. Optionally, blending can be either in progress or has completed. In an example, a first set of virtual spent components (e.g., v.sub.s,1) from a first column of the matrix V.sub.s are required for blending a 1.sup.st blending product from the products **406**. In another example, a second set of virtual spent components (e.g., v.sub.s,2) from a second column of the matrix V.sub.s are required for blending a 2.sup.nd blending product from the products **406**, and the like. Optionally, one or more set of virtual spent components from one or more columns of the matrix V.sub.s can be utilized for blending the i.sup.th blending product from the products **406** in a batch process for production of the industrial process product for the order of the one or more orders without limiting the scope of the present disclosure. In an example, the one or more virtual spent components are selected from the current component inventory **402** based on the received optimization request, such as to produce the required industrial process product.

[0163] In one or more embodiments, the method **1200** further comprising, utilizing the asynchronous update scheme by the processor **710**, for determining the component spent characteristics associated with the one or more blending products based on the industrial process product characteristics for producing the industrial process product. In one or more embodiments, the method **1200** comprises, using the asynchronous update scheme, by the processor **710**, for determining the component spent characteristics based on monitoring an amount of material (e.g., gases, hydrocarbons) included in the one or more blending components, such as material included in the current component inventory **402**. Additionally, or alternatively, in an embodiment, the method **1200** comprises, using the asynchronous update scheme for determining an amount of material used by the one or more blending products (e.g., the product **406**) of one or more batch operations associated with the production of the industrial process product for the order of the one or more orders. In one or more embodiments, the method **1200** comprises, using the asynchronous update scheme for determining the component spent characteristics based on the industrial process product characteristics for the industrial process product for the order of the one or more orders. In one or more embodiments, the method **1200** comprises, using the asynchronous update scheme for determining the component spent characteristics by predicting one or more component spent characteristics (e.g., using machine learning). For example, in one or more embodiments, the method **1200** comprises, using the asynchronous update scheme for performing a machine learning process with respect to industrial process data related to the industrial process. In certain embodiments, the method **1200** comprises, using the asynchronous update scheme for determining at least a portion of the component spent characteristics based on relationships, correlations and/or predictions between aspects of industrial process data related to the industrial process.

[0164] In one or more embodiments, the method **1200** further comprising, determining, by the processor **710**, the component spent characteristics based on monitoring content of material included in the one or more blending components. In an example, the method **1200** comprises, monitoring, by the processor **710**, content of material (e.g., flammable mixture of gases) included in the current component inventory **402** for producing the regular gas. In another example, the method **1200** comprises, monitoring, by the processor **710**, another content of material (e.g., percentage of hydrocarbons) included in the current component inventory **402** for producing the diesel based on the received request. Optionally, the content in a blending component (e.g., a product tank) can be transferred to another blending component when the asynchronous update scheme runs. For example, during the production of the natural gas, the content from the blending component can be transferred to another blending component to fulfil the order.

[0165] The method **1200** further comprises, determining, by the processor **710**, the completion of the production of the industrial process product for the order of the one or more orders. In addition, the method **1200** further comprises, determining, by the processor **710**, the utilization of the one or more virtual spent components for blending the i.sup.th blending product from the products **406** in a batch process of the production of the industrial process product for the order and within the optimization time period. For example, the method **1200** comprises, selecting, second set of virtual spent components (e.g., v.sub.s,2) from the second column of the matrix V.sub.s (as shown in FIG. **7**A within a box) for blending the 2.sup.nd blending product from the products **406**, which is also selected (as shown in FIG. **7**A within another box). Thereafter, the method **1200** further comprises, optimizing, by the processor **710**, the industrial process **602** for blending the 2.sup.nd blending product from the products **406** and for producing the industrial process product. Furthermore, the method **1200** further comprises, determining, by the processor **710**, whether the 2.sup.nd blending product is blended for producing the industrial process product to fulfill one or more orders within the optimization time period or not. In an example, the 2.sup.nd blending product can be blended for producing the regular gas. In another example, the 2.sup.nd blending product can be blended for producing diesel, and the like. Thereafter, the method **1200** comprises, determining, by the processor **710**, the utilization of the second set of virtual spent components (e.g., v.sub.s,2) from the second column of the matrix V.sub.s, which are used for blending the 2.sup.nd blending product. Optionally, the method **1200** further comprises, utilizing the asynchronous update scheme for determining the utilization of the second set of virtual spent components (e.g., v.sub.s,2) from the second column of the matrix V.sub.s. In an implementation, the asynchronous update scheme can be executed on a schedule or based on demand, such as there is no need to synchronize the asynchronous update scheme with the plantwide control and optimization execution schedule.

[0166] In one or more embodiments, the method **1200** further comprising, utilizing the virtual content tracker for tracking the component spent characteristics associated with the one or more blending products based on the optimization time period. In various embodiments, the virtual content tracker is strategically inserted into a continuous control process or an optimization process (e.g., a plant optimizer) for the industrial process **602** or industrial facility to monitor spent inventory for a batch process. In one or more embodiments, the virtual content tracker is a discontinuous system configured for monitoring spent inventory related to a batch process. In one or more embodiments, the virtual content tracker is employed with the asynchronous update scheme that is used for determining how batch operations or batch productions should be updated when a continuous optimizer (e.g., a plantwide optimizer) is receiving new input at a pre-specified frequency, or at a user request. Optionally, the method **1200** comprises, utilizing, the virtual content tracker for simultaneously tracking the optimization time period as well as the component spent characteristics associated with the current component inventory **402**, such as for the order of the one or more orders. As a result, the virtual content tracker is beneficial to fulfil the demand and/or for completing a task associated with the continuous industrial process based on the received optimization request.

[0167] At block **1216**, the method **1200** comprises, updating, by the processor **710**, demand data for the one or more feed products utilized for blending the one or more blending products based on the component spent characteristics. In an implementation, the method **1200** comprises, determining, by the processor **710**, the utilization of the one or more feed products, for example for determining the utilization of the crude oil **302** into the current component inventory **402** (e.g., blending components). In one or more embodiments, the utilization of the one or more feed products is measured by determining the one or more blending products, the component spent characteristics, and then back-calculating the product feed products. In an implementation, utilization of the one or more feed products is measured by measuring utilization of each column vector “v.sub.s,i” of the matrix V.sub.s. Optionally, utilization of each column vector “v.sub.s, i” can be measured in at least two different ways, either by measuring and accumulating the blending component that are used for blending the i.sup.th product order or by measuring the volume (or level) in the one or more blending product and then back-calculating the virtual spent components. In an example, the blending component that are used for blending the i.sup.th product order can be measured and accumulated by measuring the utilization of crude oil **302** into the current component inventory **402** (e.g., blending components). Similarly. The volume (or level) in the one or more blending product can be measured by determining the virtual spent components spent while blending the one or more blending product. For example, if the second set of virtual spent components (e.g., v.sub.s,2) from the second column of the matrix V.sub.s are utilized for blending the 2.sup.nd blending product, then the method **1200** comprises, determining, by the processor **710**, the volume of the products **406** based on the determination of the utilization of the second set of virtual spent components (e.g., v.sub.s,2). Thereafter, the method **1200** further comprises, updating, by the processor **710**, the demand data for the one or more feed products, such as by updating one or more blending products and by updating the virtual spent component in the matrix V.sub.s. In an implementation, the method **1200** further comprises, updating, by the processor **710**, the industrial process product produced by deleting the blended orders in a confirmed demand vector (P.sub.m), such as by deleting the 2.sup.nd blending product from the products **406**, as shown in FIG. **7**B by a blank tank number ‘2’. An exemplary implementation for the same is further shown and described in FIG. **8**B. Similarly, the method **1200** further comprises, resetting, by the processor **710**, the corresponding spent components, “v.sub.s,i” to zero, such as deleting the second set of virtual spent components (e.g., v.sub.s,2) from the second column of the matrix V.sub.s that are utilized for blending the 2.sup.nd blending product as shown in FIG. **7**B by a blank set of components in second row. By virtue of updating the demand data, the method **1200** providing one or more technical improvements such as, but not limited to, extended functionality for a computing device and/or improving accuracy of data provided to a computing device. In addition, the method **1200** providing an improved product demand horizon without compromising the control and optimization responsiveness of the plantwide control and optimization.

[0168] In one or more embodiments, the method **1200** further comprising, updating, by the processor **710**, the demand data for the one or more feed products utilized for producing the one or more blending components based on the component spent characteristics and in response to updating the demand data, the method **1200** further comprising, utilizing the processor **710** for resetting the first time period. In other words, if a batch for the production of the industrial process product for the order of the one or more orders is completed, then the method **1200** comprises, updating corresponding downstream inventory, by the processor **710** and resetting the first time period. In an implementation, the method **1200** comprises, utilizing the asynchronous update scheme for resetting the first time period. By virtue of resetting the first time period, the virtual content tracker associated with the current component inventory **402** is also reset. In an implementation, by virtue of updating the demand data for the one or more feed products, the method **1200** comprises, using, the processor **710** for revising a future minimum component vector c.sub.m for the plantwide control and optimization, as shown below:

[00014] $c_m = \int_0^{m\text{ - Ets}} G_F * u\,dt + V_0 + V_s = \int_t^{T_s} G_F * u\,dt + V_0 + V_s$ [0169] where E.sub.ts corresponds to elapsed time since the last asynchronous update scheme run, and T.sub.s is the demand end time (i.e., optimal components cm will be produced by Ts).

[0170] In one or more embodiments, the method **1200** further comprising, updating, by the processor **1200**, the demand data by updating one or more portions of the demand data in response to a change in demand for the industrial process product. For example, if an updated optimization request is received with the change in demand for the industrial process product (e.g., the current component inventory **402** and/or the products **406** being blended), then the method **1200** comprises, utilizing, the virtual content component **706** for updating one or more portions of the demand data based on the updated optimization request. In an implementation, if there is change in demand for the industrial process product (i.e., any confirmed demand is changed since the last asynchronous update scheme), then the method **1200** comprises, updating the confirmed demand vector (P.sub.m) to include the new/updated orders. Thereafter, the method **1200** further comprises, resetting the elapsed time (E.sub.ts) to zero for remaining one or more orders and setting the first time period (T.sub.s*) to a new value for remaining one or more orders, which is different from the previous first time period.

[0171] In an implementation, the method **1200** further comprises, updating, by the processor **710**, the demand data by updating the confirmed demand vector (P.sub.m), the component vector c.sub.m, and component mass balance equation in the plantwide control and optimization formulation. In an example, the confirmed demand vector (P.sub.m) is updated and revised to P.sub.m1=((T.sub.s1−T.sub.0)/m.sub.0)*P.sub.m, where m.sub.0=T.sub.s−T.sub.0, where T.sub.s1 is the end of the optimization horizon at time ‘t’. In an implementation, the method **1200** further comprises, tracking, by the processor **710**, the spent components from the matrix V.sub.s for the production of the industrial process product for an order of the one or more orders and updating, by the processor **710**, the component vector c.sub.m to c.sub.m1 and also updating component mass balance equation as shown below:

[00015]
$$R \,.\, p_{m1}^{\text{before}} = R \,.\, p_{m1}^{\text{after}} + v_{s,2} \le \int_0^{\text{OH}} G_F\,u\,dt + V_0 + V_s^{\text{after}} + v_{s,2}$$
$$R \,.\, p_{m1}^{\text{after}} \le c_{m1} = \int_0^{\text{OH}} G_{\text{Fudt}} + V_o + V_s^{\text{after}}$$

[0172] Where, OH=(t, T.sub.s1), u=conjoint MVs, and R denote a recipe matrix:

[00016]
$$R = \begin{bmatrix} r_{1,1} & r_{1,2} & \text{.Math.} & r_{1,\text{np}} \\ r_{2,1} & r_{2,2} & \text{.Math.} & r_{2,\text{np}} \\ \text{.Math.} & & & \text{.Math.} \\ r_{\text{nc},1} & r_{\text{nc},2} & \text{.Math.} & r_{\text{nc},\text{np}} \end{bmatrix}$$
$$\underset{i=1}{\overset{\text{nc}}{\text{.Math.}}}\, r_{i,j} \equiv 1$$

[0173] The i-th column in the R is a fractional recipe for blending the i-th product.

[0174] Moreover, the updated component mass balance equation (i.e., after blending) is same as that of an original equation (i.e., before blending). Furthermore, the matrix V.sub.s is used for tracking each blend that has been used by the blending components, thus the matrix V.sub.s can be used for tracking both the blend amount and corresponding recipe deviations. Furthermore, the remainder of the spent components become “reusable” components for the future demand, which may potentially introduce a small bias to the solution.

[0175] In another implementation, the method **1200** further comprises, tracking, by the processor **710**, the blended products for producing the industrial process product for an order of the one or more orders and updating, by the processor **710**, the component vector c.sub.m to c.sub.m2 and updating the component mass balance equation as shown below:

[00017]
$$R(p_{m1}^{after} - \text{sum}(V_s^{after})^T) = R(p_{m1}^{after} - \text{sum}(V_s^{after})^T) \leq c_{m2} = \int_0^{OH} G_F \text{udt} + V_0$$
$$R.\ .\text{Math.}\ p_{m1} - \text{sum}(V_s)^T\ .\text{Math.}\ \leq c_{m2} = \int_0^{OH} G_F \text{udt} + V_0$$

[0176] Here, to prevent any reverse flows c.sub.m2<0. Furthermore, ColSum (V.sub.s) is used for tracking all blends, both in-progress and completed since the last asynchronous update scheme run, which are then excluded from the plantwide control and optimization. Beneficially, the processor **710** can remove the possibility of a small bias that could potentially be introduced in previous equation related to tracking spent components. Moreover, in such implementation, if any in-progress blend is canceled, then the method **1200** comprises, using, the processor **710** for updating the confirmed demand vector (i.e., P.sub.m1) and reusing the affected one as a starting heel for a future blend of the same/similar grade. In an example, if production disturbance is too noisy, then such implementation can be implemented for buffering the disturbances before rejecting the products.

[0177] In such implementation, the method **1200** further comprises, using, the processor **710**, for tracking the blended products for producing the industrial process product for an order of the one or more orders, updating the confirmed demand vector (P.sub.m) and revising the confirmed demand vector from P.sub.m to P.sub.m2. Moreover, the method **1200** comprises, using, the processor **710** for updating the component vector from c.sub.m to c.sub.m2 and updating the component mass balance equation more precisely, as shown below:

[00018]
$$R.p_{m2} \leq c_{m2} = \int_0^{OH} G_F \text{udt} + V_0$$
$$p_{m2} = \frac{OH}{T_s - t}\ .\text{Math.}\ \ .\text{Math.}\ p_m - \text{sum}(V_s)^T\ .\text{Math.}\ = \frac{T_{S1} - t}{T_s - t}\ .\text{Math.}\ \ .\text{Math.}\ p_m - \text{sum}(V_s)^T\ .\text{Math.}$$

[0178] Here, P.sub.m2=[OH/(T.sub.s−t)]*abs (P.sub.m−sum (V.sub.s)'). Moreover, in such implementation, an if the accumulative number of blended products in elapsed window (T.sub.0, t), is different from an “expected amount”, due to disturbances, then an auto production catch-up in the remaining time window (t, T.sub.s), which is beneficial for rejecting undesired disturbance.

[0179] At block **1218**, the method **1200** comprises, transmitting, by the processor **710**, a control signal to control the flow of the feed component in the continuous process. The transmitted control signal provides one or more technical improvements such as, but not limited to, providing a varied experience for a computing device. In one or more embodiments, the method **1200** further comprising, generating, by the processor, one or more batch blending recipes based on the updated demand data. In various embodiments, the management of the industrial process optimization provides one or more optimal batch blending recipes (e.g., optimal recipes, optimal batch-blending plantwide recipes, etc.) for the industrial process.

[0180] The method **1200** is used for controlling the industrial process with improved product demand horizon without compromising the control and optimization responsiveness of the plantwide control and optimization. The method **1200** is beneficial for controlling the plantwide control and optimization for operating with a time horizon suitable for mitigating different types of disturbances during the production of the industrial process product to fulfill one or more orders. Therefore, the method **1200** liberates the plantwide control and optimization from the constraint of the product demand horizon, transforming the plantwide control and optimization into an improved method, which can be used for batch blending oil refineries.

[0181] FIG. **13** illustrates a flow diagram for managing an industrial process in a process plant, in accordance with one or more embodiments described herein. At step **1302** of exemplary flowchart **1300**, the industrial process computer system **702** comprises means such as, asynchronous update component **704** to determine a status of a first order of one or more orders based at least on component spent characteristics. The first order is associated with production of an industrial process product by the industrial process. At step **1304** of the exemplary flowchart **1300**, the industrial process computer system **702** comprises means such as, processor **710** to set a flag to a first value if the status of the first order corresponds to a first status. In one or more embodiments, the flag indicates the status of an order of the one or more orders. At step **1306** of the exemplary flowchart **1300**, the industrial process computer system **702** comprises means such as, processor **710** to receive an optimization request to optimize the industrial process with respect to a second order of the one or more orders. The second order is associated with production of the industrial process product by the industrial process. At step **1308** of the exemplary flowchart **1300**, the industrial process computer system **702** comprises means such as, processor **710** or control component **708** to render on a display one or more first notifications based on the first value. In this regard, the one or more first notifications can correspond to displaying the first status of the first order based on the first value and providing an alert to for instance, an operator of the process plant to turn controller **502** to warm state. At step **1310** of the exemplary flowchart **1300**, the industrial process computer system **702** comprises means such as, processor **710** to set the flag to a second value if the status of the first order corresponds to a second status. In response to setting the flag to the second value, at step **1312**, the industrial process computer system **702** comprises means such as, processor **710** or control component **708** to render on the display, one or more second notifications based on the second value. In this regard, the one or more second notifications can correspond to displaying the second status of the first order based on the second value and providing an action to an operator to: update details of the first order if the details of the first order are absent in demand data and turn the controller **502** to on state. Further, at step **1314**, the industrial process computer system **702** comprises means such as, processor **710** or control component **708** to update the demand data associated with the one or more orders based on the component spent characteristics and the optimization request. Furthermore, at step **1316**, the industrial process computer system **702** comprises means such as, processor **710** or control component **708** to transmit a control signal based on the demand data to the controller **502** configured to optimize the industrial process with respect to the second order.

[0182] FIG. **14** depicts a system **1400** that may execute techniques presented herein. FIG. **14** is a simplified functional block diagram of a computer that may be configured to execute techniques described herein, according to exemplary embodiments of the present disclosure. Specifically, the computer (or “platform” as it may not be a single physical computer infrastructure) may include a data communication interface **1460** for packet data communication. The platform also may include a central processing unit (“CPU”) **1420**, in the form of one or more processors, for executing program instructions. The platform may include an internal communication bus **1410**, and the platform also may include a program storage and/or a data storage for various data files to be processed and/or communicated by the platform such as ROM **1430** and RAM **1440**, although the system **1400** may receive programming and data via network communications. The system **1400** also may include input and output ports **1450** to connect with input and output devices such as keyboards, mice, touchscreens, monitors, displays, etc. Of course, the various system functions may be implemented in a distributed fashion on a number of similar platforms, to distribute the processing load. Alternatively, the systems may be implemented by appropriate programming of one computer hardware platform.

[0183] The foregoing embodiments are provided merely as illustrative examples and are not intended to require or imply that the steps of the various embodiments must be performed in the order presented. As will be appreciated by one of skill in the art the order of steps in the foregoing embodiments can be performed in any order. Words such as “thereafter,” “then,” “next,” etc. are not intended to limit the order of the steps; these words are simply used to guide the reader through the description of the methods. Further, any reference to claim elements in the singular, for example, using the articles “a,” “an” or “the” is not to be construed as limiting the element to the

singular.
[0184] It is to be appreciated that ‘one or more’ includes a function being performed by one element, a function being performed by more than one element, e.g., in a distributed fashion, several functions being performed by one element, several functions being performed by several elements, or any combination of the above.
[0185] Moreover, it will also be understood that, although the terms first, second, etc. are, in some instances, used herein to describe various elements, these elements should not be limited by these terms. These terms are only used to distinguish one element from another. For example, a first contact could be termed a second contact, and, similarly, a second contact could be termed a first contact, without departing from the scope of the various described embodiments. The first contact and the second contact are both contacts, but they are not the same contact.
[0186] The terminology used in the description of the various described embodiments herein is for the purpose of describing particular embodiments only and is not intended to be limiting. As used in the description of the various described embodiments and the appended claims, the singular forms “a”, “an” and “the” are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will also be understood that the term “and/or” as used herein refers to and encompasses any and all possible combinations of one or more of the associated listed items. It will be further understood that the terms “includes,” “including,” “comprises,” and/or “comprising,” when used in this specification, specify the presence of stated features, integers, steps, operations, elements, and/or components, but do not preclude the presence or addition of one or more other features, integers, steps, operations, elements, components, and/or groups thereof.
[0187] As used herein, the term “if” is, optionally, construed to mean “when” or “upon” or “in response to determining” or “in response to detecting,” depending on the context. Similarly, the phrase “if it is determined” or “if [a stated condition or event] is detected” is, optionally, construed to mean “upon determining” or “in response to determining” or “upon detecting [the stated condition or event]” or “in response to detecting [the stated condition or event],” depending on the context.
[0188] The systems, apparatuses, devices, and methods disclosed herein are described in detail by way of examples and with reference to the figures. The examples discussed herein are examples only and are provided to assist in the explanation of the apparatuses, devices, systems, and methods described herein. None of the features or components shown in the drawings or discussed below should be taken as mandatory for any specific implementation of any of these the apparatuses, devices, systems or methods unless specifically designated as mandatory. For case of reading and clarity, certain components, modules, or methods may be described solely in connection with a specific figure. In this disclosure, any identification of specific techniques, arrangements, etc. are either related to a specific example presented or are merely a general description of such a technique, arrangement, etc. Identifications of specific details or examples are not intended to be, and should not be, construed as mandatory or limiting unless specifically designated as such. Any failure to specifically describe a combination or sub-combination of components should not be understood as an indication that any combination or sub-combination is not possible. It will be appreciated that modifications to disclosed and described examples, arrangements, configurations, components, elements, apparatuses, devices, systems, methods, etc. can be made and may be desired for a specific application. Also, for any methods described, regardless of whether the method is described in conjunction with a flow diagram, it should be understood that unless otherwise specified or required by context, any explicit or implicit ordering of steps performed in the execution of a method does not imply that those steps must be performed in the order presented but instead may be performed in a different order or in parallel.
[0189] Throughout this disclosure, references to components or modules generally refer to items that logically can be grouped together to perform a function or group of related functions. Like reference numerals are generally intended to refer to the same or similar components. Components

and modules can be implemented in software, hardware, or a combination of software and hardware. The term “software” is used expansively to include not only executable code, for example machine-executable or machine-interpretable instructions, but also data structures, data stores and computing instructions stored in any suitable electronic format, including firmware, and embedded software. The terms “information” and “data” are used expansively and includes a wide variety of electronic information, including executable code; content such as text, video data, and audio data, among others; and various codes or flags. The terms “information,” “data,” and “content” are sometimes used interchangeably when permitted by context.

[0190] The hardware used to implement the various illustrative logics, logical blocks, modules, and circuits described in connection with the aspects disclosed herein can include a general purpose processor, a digital signal processor (DSP), a special-purpose processor such as an application specific integrated circuit (ASIC) or a field programmable gate array (FPGA), a programmable logic device, discrete gate or transistor logic, discrete hardware components, or any combination thereof designed to perform the functions described herein. A general-purpose processor can be a microprocessor, but, in the alternative, the processor can be any processor, controller, microcontroller, or state machine. A processor can also be implemented as a combination of computing devices, e.g., a combination of a DSP and a microprocessor, a plurality of microprocessors, one or more microprocessors in conjunction with a DSP core, or any other such configuration. Alternatively, or in addition, some steps or methods can be performed by circuitry that is specific to a given function.

[0191] In one or more example embodiments, the functions described herein can be implemented by special-purpose hardware or a combination of hardware programmed by firmware or other software. In implementations relying on firmware or other software, the functions can be performed as a result of execution of one or more instructions stored on one or more non-transitory computer-readable media and/or one or more non-transitory processor-readable media. These instructions can be embodied by one or more processor-executable software modules that reside on the one or more non-transitory computer-readable or processor-readable storage media. Non-transitory computer-readable or processor-readable storage media can in this regard comprise any storage media that can be accessed by a computer or a processor. By way of example but not limitation, such non-transitory computer-readable or processor-readable media can include random access memory (RAM), read-only memory (ROM), electrically erasable programmable read-only memory (EEPROM), FLASH memory, disk storage, magnetic storage devices, or the like. Disk storage, as used herein, includes compact disc (CD), laser disc, optical disc, digital versatile disc (DVD), floppy disk, and Blu-ray Disc™, or other storage devices that store data magnetically or optically with lasers. Combinations of the above types of media are also included within the scope of the terms non-transitory computer-readable and processor-readable media. Additionally, any combination of instructions stored on the one or more non-transitory processor-readable or computer-readable media can be referred to herein as a computer program product.

[0192] Many modifications and other embodiments of the inventions set forth herein will come to mind to one skilled in the art to which these inventions pertain having the benefit of teachings presented in the foregoing descriptions and the associated drawings. Although the figures only show certain components of the apparatus and systems described herein, it is understood that various other components can be used in conjunction with the supply management system. Therefore, it is to be understood that the inventions are not to be limited to the specific embodiments disclosed and that modifications and other embodiments are intended to be included within the scope of the appended claims. Moreover, the steps in the method described above can not necessarily occur in the order depicted in the accompanying diagrams, and in some cases one or more of the steps depicted can occur substantially simultaneously, or additional steps can be involved. Although specific terms are employed herein, they are used in a generic and descriptive sense only and not for purposes of limitation.

[0193] It is intended that the specification and examples be considered as exemplary only, with a true scope and spirit of the disclosure being indicated by the following claims.

Claims

1. A method for managing an industrial process in a process plant, the method comprising: determining a status of a first order of one or more orders based at least on component spent characteristics, wherein the first order is associated with production of an industrial process product by the industrial process; setting a flag to a first value if the status of the first order corresponds to a first status, wherein the flag indicates the status of an order of the one or more orders; receiving an optimization request to optimize the industrial process with respect to a second order of the one or more orders, wherein the second order is associated with production of the industrial process product by the industrial process; in response to receiving the optimization request, rendering on a display one or more first notifications based on the first value; setting the flag to a second value if the status of the first order corresponds to a second status; and in response to setting the flag to the second value: rendering on the display, one or more second notifications based on the second value; updating demand data associated with the one or more orders based on the component spent characteristics and the optimization request; and transmitting a control signal based on the demand data to a controller configured to optimize the industrial process with respect to the second order.
2. The method of claim 1, wherein rendering on the display the one or more first notifications based on the first value comprising: displaying the first status of the first order based on the first value; and providing an alert to an operator to turn the controller to warm state.
3. The method of claim 1, wherein rendering on the display the one or more second notifications based on the second value comprising: displaying the second status of the first order based on the second value; and providing an action to an operator to: update details of the first order if the details of the first order are absent in the demand data; and turn the controller to on state.
4. The method of claim 1, further comprising, receiving the optimization request in response to a determination that time data for the industrial process satisfies a defined criterion.
5. The method of claim 1, further comprising: determining the component spent characteristics based at least on industrial process product characteristics for the industrial process product and one or more virtual spent components, wherein the one or more virtual spent components are associated with at least one of: an asynchronous update scheme and a virtual content tracker; utilizing the virtual content tracker for tracking the component spent characteristics based on an optimization time period; and utilizing the asynchronous update scheme for determining the component spent characteristics based on the industrial process product characteristics for producing the industrial process product.
6. The method of claim 5, further comprising: determining the component spent characteristics based on monitoring content of material included in one or more blending components to produce the industrial process product.
7. The method of claim 1, further comprising: updating the demand data by updating one or more portions of the demand data in response to a change in demand for the industrial process product.
8. The method of claim 1, further comprising generating one or more batch blending recipes based on the updated demand data.
9. A system for managing an industrial process in a process plant, the system comprising: a processor; a memory communicatively coupled to the processor, wherein the memory comprises one or more instructions which when executed by the processor, cause the processor to: determine a status of a first order of one or more orders based at least on component spent characteristics, wherein the first order is associated with production of an industrial process product by the industrial process; set a flag to a first value if the status of the first order corresponds to a first status, wherein the flag indicates the status of an order of the one or more orders; receive an

optimization request to optimize the industrial process with respect to a second order of the one or more orders, wherein the second order is associated with production of the industrial process product by the industrial process; in response to receiving the optimization request, render on a display one or more first notifications based on the first value; set the flag to a second value if the status of the first order corresponds to a second status; and in response to setting the flag to the second value: render on the display, one or more second notifications based on the second value; update demand data associated with the one or more orders based on the component spent characteristics and the optimization request; and transmit a control signal based on the demand data to a controller configured to optimize the industrial process with respect to the second order.

10. The system of claim 9, wherein the one or more first notifications based on the first value correspond to: displaying the first status of the first order based on the first value; and providing an alert to an operator to turn the controller to warm state.

11. The system of claim 9, wherein the one or more second notifications based on the second value correspond to: displaying the second status of the first order based on the second value; and providing an action to an operator to: update details of the first order if the details of the first order are absent in the demand data; and turn the controller to on state.

12. The system of claim 9, wherein the processor is further configured to receive the optimization request in response to a determination that time data for the industrial process satisfies a defined criterion.

13. The system of claim 9, wherein the processor is further configured to: determine the component spent characteristics based at least on industrial process product characteristics for the industrial process product and one or more virtual spent components, wherein the one or more virtual spent components are associated with at least one of: an asynchronous update scheme and a virtual content tracker; utilize the virtual content tracker for tracking the component spent characteristics based on an optimization time period; and utilize the asynchronous update scheme for determining the component spent characteristics based on the industrial process product characteristics for producing the industrial process product.

14. The system of claim 13, wherein the processor is further configured to update the demand data by updating one or more portions of the demand data in response to a change in demand for the industrial process product.

15. The system of claim 11, wherein the processor is further configured to generate one or more batch blending recipes based on the updated demand data.

16. A non-transitory, computer-readable storage medium having stored thereon executable instructions that, when executed by one or more processors, cause the one or more processors to: determine a status of a first order of one or more orders based at least on component spent characteristics, wherein the first order is associated with production of an industrial process product by the industrial process; set a flag to a first value if the status of the first order corresponds to a first status, wherein the flag indicates the status of an order of the one or more orders; receive an optimization request to optimize the industrial process with respect to a second order of the one or more orders, wherein the second order is associated with production of the industrial process product by the industrial process; in response to receiving the optimization request, render on a display one or more first notifications based on the first value; set the flag to a second value if the status of the first order corresponds to a second status; and in response to setting the flag to the second value: render on the display, one or more second notifications based on the second value; update demand data associated with the one or more orders based on the component spent characteristics and the optimization request; and transmit a control signal based on the demand data to a controller configured to optimize the industrial process with respect to the second order.

17. The non-transitory, computer-readable storage medium of claim 16, wherein the one or more first notifications based on the first value correspond to: displaying the first status of the first order based on the first value; and providing an alert to an operator to turn the controller to warm state.

18. The non-transitory, computer-readable storage medium of claim 16, wherein the one or more second notifications based on the second value correspond to: displaying the second status of the first order based on the second value; and providing an action to an operator to: update details of the first order if the details of the first order are absent in the demand data; and turn the controller to on state.

19. The non-transitory, computer-readable storage medium of claim 16, wherein the one or more processors is further configured to receive the optimization request in response to a determination that time data for the industrial process satisfies a defined criterion.

20. The non-transitory, computer-readable storage medium of claim 16, wherein the one or more processors is further configured to: determine the component spent characteristics based at least on industrial process product characteristics for the industrial process product and one or more virtual spent components, wherein the one or more virtual spent components are associated with at least one of: an asynchronous update scheme and a virtual content tracker; utilize the virtual content tracker for tracking the component spent characteristics based on an optimization time period; and utilize the asynchronous update scheme for determining the component spent characteristics based on the industrial process product characteristics for producing the industrial process product.