(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
02 January 2025 (02.01.2025)

WIPO | PCT

(10) International Publication Number
WO 2025/007120 A2

(51) International Patent Classification:
G06N 7/01 (2023.01)

(21) International Application Number:
PCT/US2024/036382

(22) International Filing Date:
01 July 2024 (01.07.2024)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
63/511,343 30 June 2023 (30.06.2023) US

(71) Applicant: **OHIO STATE INNOVATION FOUNDATION** [US/US]; Mount Hall, 1st Floor, 1050 Carmack Rd, Columbus, Ohio 43210 (US).

(72) Inventors: **SMIDTS, Carol**; c/o Ohio State Innovation Foundation, Mount Hall, 1st Floor, 1050 Carmack Rd, Columbus, Ohio 43210 (US). **DIAO, Xiaoxu**; c/o Ohio State Innovation Foundation, Mount Hall, 1st Floor, 1050 Carmack Rd, Columbus, Ohio 43210 (US). **VADDI, Pavan Kumar**; c/o Ohio State Innovation Foundation, Mount Hall, 1st Floor, 1050 Carmack Rd, Columbus, Ohio 43210 (US). **ZHAO, Yunfei**; c/o Ohio State Innovation Foundation, Mount Hall, 1st Floor, 1050 Carmack Rd, Columbus, Ohio 43210 (US).

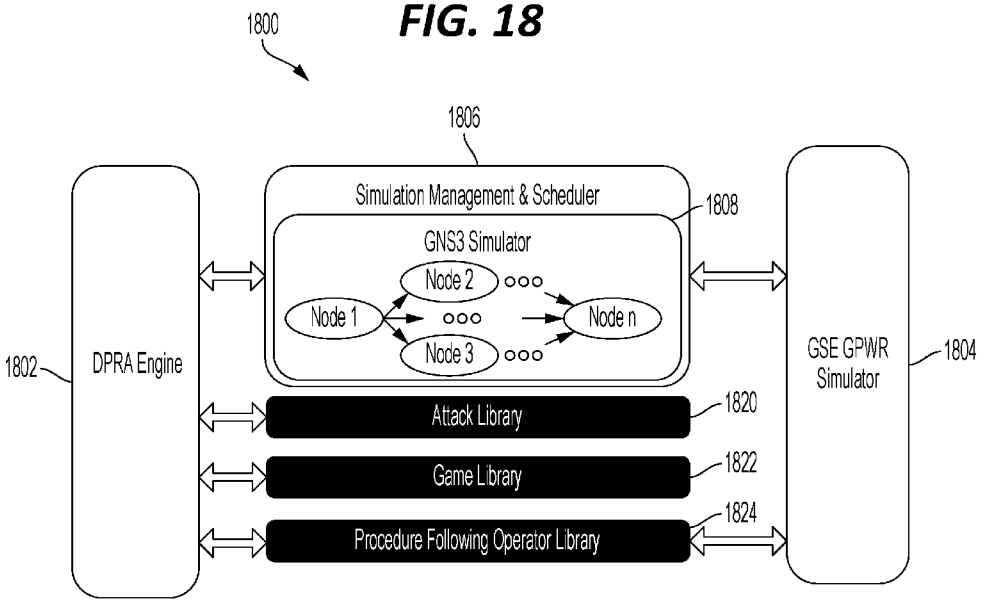
(74) Agent: **HAMILTON, Lee G.** et al.; MEUNIER CARLIN & CURFMAN LLC, 999 Peachtree St. NE, Suite 1300, Atlanta, Georgia 30309 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, CV,

(54) Title: SYSTEMS AND METHODS FOR DYNAMIC PROBABILISTIC RISK ASSESSMENT SIMULATION ENVIRONMENTS

FIG. 18



(57) Abstract: An example method for performing probabilistic risk assessment includes generating a dynamic probabilistic risk assessment corresponding to a system model; and modelling actions in the system model using an action model. An example system for performing probabilistic risk assessment includes a system model comprising a model of a physical system; a dynamic probabilistic risk assessment (DPRA) engine; and an action model.

[Continued on next page]

WO 2025/007120 A2

GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

Published:

— *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

SYSTEMS AND METHODS FOR DYNAMIC PROBABILISTIC RISK ASSESSMENT SIMULATION ENVIRONMENTS

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of U.S. provisional patent application No. 63/511,343 filed on June 30, 2023, and titled "SYSTEMS AND METHODS FOR DYNAMIC PROBABILISTIC RISK ASSESSMENT SIMULATION ENVIRONMENTS," the disclosure of which is expressly incorporated herein by reference in its entirety.

STATEMENT REGARDING FEDERALLY FUNDED RESEARCH

[0002] This invention was made with government support under DE-NE0008986 awarded by the Department of Energy. The government has certain rights in the invention.

BACKGROUND

[0003] Risk assessment involves determining the likelihood of events and the relationships of those events to dangerous scenarios. A common definition of risk is the likelihood of an event multiplied by the expected harm caused by the event. Events can include human errors, malicious actions (e.g., cyberattacks) and mechanical failures. Dangerous scenarios can be the result of evolution of an individual event or the interaction of one or more events that can result in scenarios that would not be caused by individual events alone. Moreover, dangerous scenarios can be caused by the interaction of multiple seemingly benign events interacting in a complex system. Improved systems and methods for predicting risk in complex systems can be used to improve the design, control, and operation of those systems.

SUMMARY

[0004] In some aspects, implementations of the present disclosure include a method for performing dynamic probabilistic risk assessment, the method including: providing a system model of a physical system; providing an action model configured to generate a set of operator and attacker actions based on a game theory analysis; and providing a dynamic probabilistic risk assessment (DPRA) engine, wherein the DPRA engine is configured to: receive the set of operator and attacker actions from the action model; sample the set of operator and attacker actions; and provide the sampled set of operator and attacker actions to the system model for implementation by the system model.

[0005] In some aspects, implementations of the present disclosure include a method, wherein the set of operator and attacker actions is randomly sampled.

[0006] In some aspects, implementations of the present disclosure include a method, wherein the DPRA engine is further configured to receive a state of the physical system from the system model, and provide the state of the physical system to the action model.

[0007] In some aspects, implementations of the present disclosure include a method, wherein the system model includes a network model.

[0008] In some aspects, implementations of the present disclosure include a method, wherein the DPRA engine is further configured to receive a state of the network model from the system model, and provide the state of the network model to the action model.

[0009] In some aspects, implementations of the present disclosure include a method, wherein the action model includes a plurality of sub-models.

[0010] In some aspects, implementations of the present disclosure include a method, wherein the plurality of sub-models include a procedure model, an undetected attacker model, and a game model.

[0011] In some aspects, implementations of the present disclosure include a method, wherein the DPRA engine is configured to implement an attacker action on the system model.

[0012] In some aspects, implementations of the present disclosure include a method, wherein the DPRA engine is configured to implement an operator action on the system model.

[0013] In some aspects, implementations of the present disclosure include a method, wherein the action model is configured to generate a plurality of actions for an operator to implement on the physical system.

[0014] In some aspects, implementations of the present disclosure include a method, wherein the action model is configured to generate a plurality of actions for an attacker to implement on the physical system.

[0015] In some aspects, implementations of the present disclosure include a method, wherein the action model is configured to generate both a plurality of actions for an attacker and a plurality of actions for an operator to implement on the physical system.

[0016] In some aspects, implementations of the present disclosure include a method, further including determining a risk metric based on the actions in the system, wherein the risk metric is based on the action model and the system model.

[0017] In some aspects, implementations of the present disclosure include a method, further including modeling a plurality of actions for an attacker and a plurality of actions for a defender based on a game theory analysis of the attacker and defender's actions.

[0018] In some aspects, implementations of the present disclosure include a method for performing dynamic probabilistic risk assessment, the method including: modelling actions in a system model using an action model and sampling the actions to be implemented on the system model by a DPRA engine from the actions modeled by the action model; and implementing the sampled actions on the system model.

[0019] In some aspects, implementations of the present disclosure include a method, further including determining a risk metric based on the actions in the system.

[0020] In some aspects, implementations of the present disclosure include a system including: a network device; a controller operably coupled to the network device, the controller including a processor and a memory, the memory having computer-executable instructions stored thereon that, when executed by the processor, cause the controller to: model action in a system model using an action model and sample the actions to be implemented on the system model by a DPRA engine from the actions modeled by the action model; and implement the sampled actions on the system model.

[0021] In some aspects, implementations of the present disclosure include a system, wherein the network device is coupled to a network including a functional node.

[0022] In some aspects, implementations of the present disclosure include a system, wherein the functional node includes at least one of a controller or sensor.

[0023] In some aspects, implementations of the present disclosure include a system, wherein the controller is further configured to control the network device based on the sampled actions implemented on the system model.

[0024] Other systems, methods, features and/or advantages will be or may become apparent to one with skill in the art upon examination of the following drawings and detailed description. It is intended that all such additional systems, methods, features and/or advantages be included within this description and be protected by the accompanying claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0025] The components in the drawings are not necessarily to scale relative to each other. Like reference numerals designate corresponding parts throughout the several views.

[0026] FIG. 1A illustrates a simulation environment for performing dynamic probabilistic risk assessment, according to implementations of the present disclosure.

[0027] FIG. 1B illustrates a system for improving the cybersecurity of a network, according to implementations of the present disclosure.

[0028] FIG. 2 is a flow chart illustrating a method for performing dynamic probabilistic risk assessment, according to implementations of the present disclosure.

[0029] FIG. 3 illustrates an example system model, according to implementations of the present disclosure.

[0030] FIG. 4 illustrates an example dynamic probabilistic risk assessment engine according to implementations of the present disclosure.

[0031] FIG. 5 illustrates an example action model, according to implementations of the present disclosure.

[0032] FIG. 6 illustrates an example game model, according to implementations of the present disclosure.

[0033] FIG. 7 illustrates an example computing device.

[0034] FIG. 8 illustrates an example implementation of the present disclosure including a model the strategic interactions between attackers/defenders in a system including physical and digital components.

[0035] FIG. 9 illustrates a model of attacker and defender actions in a system including digital components, mechanical components, and physical modeling, according to implementations of the present disclosure.

[0036] FIG. 10 illustrates an example diagram of a controller, digital flow sensor, and mechanical valve with example discrete states, according to implementations of the present disclosure.

[0037] FIG. 11 illustrates an example trajectory in physics space, according to implementations of the present disclosure.

[0038] FIG. 12 illustrates an example model of a trajectory, according to implementations of the present disclosure.

[0039] FIG. 13 illustrates an example game theory model, according to implementations of the present disclosure.

[0040] FIG. 14 illustrates an example reactor used in a case study according to implementations of the present disclosure.

[0041] FIG. 15 illustrates an example simulation of the reactor shown in FIG. 14, according to implementations of the present disclosure.

[0042] FIG. 16 illustrates example states of components used in the case study shown in FIGS. 14 and 15.

[0043] FIG. 17A illustrates a simulation result of narrow water range levels for steam generator 1, according to implementations of the present disclosure.

[0044] FIG. 17B illustrates a simulation result of narrow water range levels for steam generator 2, according to implementations of the present disclosure.

[0045] FIG. 17C illustrates a simulation result of narrow water range levels for steam generator 3, according to implementations of the present disclosure.

[0046] FIG. 18 illustrates a simulator that can be used to evaluate network architectures and cybersecurity risks, according to implementations of the present disclosure.

[0047] FIG. 19 illustrates co-simulation of network models and physics models, according to implementations of the present disclosure.

[0048] FIG. 20 illustrates node-management and scheduling model that can be used in the simulator shown in FIG. 18, according to implementations of the present disclosure.

[0049] FIG. 21 illustrates a plot of game-theory trajectories for attacker and defender action pairs, according to implementations of the present disclosure.

[0050] FIG. 22 illustrates a timing diagram illustrating parallel operation of a graphical network simulator, physical simulator, procedure following operator library and game library, according to implementations of the present disclosure.

DETAILED DESCRIPTION

[0051] Unless defined otherwise, all technical and scientific terms used herein have the same meaning as commonly understood by one of ordinary skill in the art. Methods and materials similar or equivalent to those described herein can be used in the practice or testing

of the present disclosure. As used in the specification, and in the appended claims, the singular forms “a,” “an,” “the” include plural referents unless the context clearly dictates otherwise. The term “comprising” and variations thereof as used herein is used synonymously with the term “including” and variations thereof and are open, non-limiting terms. The terms “optional” or “optionally” used herein mean that the subsequently described feature, event or circumstance may or may not occur, and that the description includes instances where said feature, event or circumstance occurs and instances where it does not. Ranges may be expressed herein as from "about" one particular value, and/or to "about" another particular value. When such a range is expressed, an aspect includes from the one particular value and/or to the other particular value. Similarly, when values are expressed as approximations, by use of the antecedent "about," it will be understood that the particular value forms another aspect. It will be further understood that the endpoints of each of the ranges are significant both in relation to the other endpoint, and independently of the other endpoint. While implementations will be described for resolving ambiguities in text, it will become evident to those skilled in the art that the implementations are not limited thereto, but are applicable for generative artificial intelligence systems and methods.

[0052] As used herein, the terms "about" or "approximately" when referring to a measurable value such as an amount, a percentage, and the like, is meant to encompass variations of ±20%, ±10%, ±5%, or ±1% from the measurable value.

[0053] The term “artificial intelligence” is defined herein to include any technique that enables one or more computing devices or comping systems (i.e., a machine) to mimic human intelligence. Artificial intelligence (AI) includes, but is not limited to, knowledge bases,

machine learning, representation learning, and deep learning. The term "machine learning" is defined herein to be a subset of AI that enables a machine to acquire knowledge by extracting patterns from raw data. Machine learning techniques include, but are not limited to, logistic regression, support vector machines (SVMs), decision trees, Naïve Bayes classifiers, and artificial neural networks. The term "representation learning" is defined herein to be a subset of machine learning that enables a machine to automatically discover representations needed for feature detection, prediction, or classification from raw data. Representation learning techniques include, but are not limited to, autoencoders. The term "deep learning" is defined herein to be a subset of machine learning that that enables a machine to automatically discover representations needed for feature detection, prediction, classification, etc. using layers of processing. Deep learning techniques include, but are not limited to, artificial neural network or multilayer perceptron (MLP).

[0054] Machine learning models include supervised, semi-supervised, and unsupervised learning models. In a supervised learning model, the model learns a function that maps an input (also known as feature or features) to an output (also known as target or targets) during training with a labeled data set (or dataset). In an unsupervised learning model, the model learns patterns (e.g., structure, distribution, etc.) within an unlabeled data set. In a semi-supervised model, the model learns a function that maps an input (also known as feature or features) to an output (also known as target or target) during training with both labeled and unlabeled data.

[0055] Deep learning models, including LLMs, may include artificial neural networks. An artificial neural network (ANN) is a computing system including a plurality of interconnected

neurons (e.g., also referred to as "nodes"). This disclosure contemplates that the nodes can be implemented using a computing device (e.g., a processing unit and memory as described herein). The nodes can be arranged in a plurality of layers such as input layer, output layer, and optionally one or more hidden layers. An ANN having hidden layers can be referred to as deep neural network or multilayer perceptron (MLP). Each node is connected to one or more other nodes in the ANN. For example, each layer is made of a plurality of nodes, where each node is connected to all nodes in the previous layer. The nodes in a given layer are not interconnected with one another, i.e., the nodes in a given layer function independently of one another. As used herein, nodes in the input layer receive data from outside of the ANN, nodes in the hidden layer(s) modify the data between the input and output layers, and nodes in the output layer provide the results. Each node is configured to receive an input, implement an activation function (e.g., binary step, linear, sigmoid, tanH, or rectified linear unit (ReLU) function), and provide an output in accordance with the activation function. Additionally, each node is associated with a respective weight. ANNs are trained with a dataset to maximize or minimize an objective function. In some implementations, the objective function is a cost function, which is a measure of the ANN's performance (e.g., error such as L1 or L2 loss) during training, and the training algorithm tunes the node weights and/or bias to minimize the cost function. This disclosure contemplates that any algorithm that finds the maximum or minimum of the objective function can be used for training the ANN. Training algorithms for ANNs include, but are not limited to, backpropagation.

[0056] Industrial facilities can include complex networked controllers, computing devices, sensors, and actuators, which interact with physical systems to operate the facility.

These complexed networked systems create cyber risks due to the interaction of the digital systems with the physical systems of the facility, and allow for attackers to cause physical damage to facilities using cyber attacks.

[0057] For example, the increasing adaptation of nuclear power plants to incorporate software-based components along with digital communication networks in their operation has resulted in improved control, automation, monitoring and diagnostics, while simultaneously opening those power plants to a new dimension of risk, cyber-attacks. Additionally, the attackers have become more knowledgeable about the vulnerabilities associated with such software systems and network architectures. The complicated physical and digital systems of industrial facilities pose challenges to existing methods and systems for evaluating cyber attacks and cyber defenses.

[0058] Implementations of the present disclosure improve systems and methods for evaluating cyber attacks and defenses, for example by using a dynamic probabilistic risk assessment (DPRA) framework for nuclear power plants in the context of cyber security. In addition to stochastic events such as component failures, the framework considers cyber-attacks along with defenders' (e.g., the plant operators), and the attackers' behaviors and their interactions in a game theory-based framework.

[0059] The example implementation described herein includes improvements to the simulation of complex systems including nuclear powerplants. The implementations described herein allow a dynamic probabilistic risk assessment simulation environment that can model the physical operation of a facility (e.g., the system model) to operate with game-theory based frameworks for decision making, to improve computation of risk metrics associated with cyber-

attacks. Existing frameworks can fail to account for both physical conditions/events and the strategic interactions between the actions of individuals (e.g., the attackers and the operators in relation to those systems).

[0060] For example, conventional probabilistic risk assessment (PRA) methods include fault trees and event trees, which can identify potential failures in a system. But conventional PRA methods are limited by modeling and incorporating the changes of system properties as functions of physics and time, for example failure rates of components are dependent on physical conditions, modeling the changes in the behavior of human operators with respect to system states and time, for example the operator can be under a significant amount of stress depending on the state of the system, and capturing the evolution of the system over time due to events such as random component failures or operator errors.

[0061] The systems and methods described herein (e.g., the dynamic probabilistic risk assessment methods referred to herein as DPRA) include a set of probabilistic risk assessment techniques that uses deterministic physics based dynamic models of the system to study its evolution in the context of random events [8], [9] and can overcome the limitations of classical PRA and provide more accurate estimates of risk. DPRA systems can include evaluating initiating events such as failure of hardware components either due to aging or random failures [8], [10], failure of digital systems [11] and human operator errors [12]-[14] into consideration, for example using continuous event tree approaches, continuous time DPRA methods, and modeling random events including equipment failures, operator actions, and operator errors. [8], [12].

[0062] The improvements to cybersecurity systems and methods described herein can thereby improve physical systems by improving the security of the networks that sense and control those systems (e.g., the networks of a nuclear power plant or other industrial facility). For example, the cybersecurity systems and methods described herein include systems and methods for analyzing the physical system (e.g., a nuclear power plant) that is sensed and/or controlled by the network, which improves on cybersecurity systems and methods that do not incorporate the physical system into the analysis and therefore do not consider the risks of different types of cyberattacks on the physical system. In other words, the present disclosure includes cybersecurity analysis including both physical modeling and behavioral modeling (e.g., using game theory, that can improve on previous systems which do not combine behavioral modeling with physical modeling. [7], [15].

[0063] With reference to FIG. 1A, a system 100 for performing dynamic probabilistic risk assessment (DPRA) is shown according to an implementation of the present disclosure. As shown in FIG. 1A, the system 100 includes a system model 102. The system model 102 can be a model of a physical system. As a non-limiting example, the physical system can be an industrial facility, for example a nuclear power plant or reactor. The system model 102 can optionally include mathematical models of how physical parts of the system perform under different conditions and/or how those physical parts of the system interact with each other. A non-limiting example of a nuclear power plant system model 300 is shown in FIG. 3. Alternatively or additionally, the system model 102 can include a network model that models a network within the system.

[0064] Still with reference to FIG. 1A, the system 100 can include a DPRA engine 110. The DPRA engine 110 can be configured to implement an attacker action on the system model 102. As an example, an attacker action can include a simulation of a cyber attack by an attacker on the system model 102. A non-limiting example of an attacker action includes a cyber attack on a nuclear facility. FIG. 4 illustrates an example DPRA engine 400 configured for use with the nuclear power plant system model 300 shown in FIG. 3.

[0065] The DPRA engine can be optionally configured to receive a state of the network model from the system model, and provide the state of the network model to the action model. As described in Example 1, herein, the network model can include functional nodes (e.g., any number or combination of sensors, actuators, computers, and controllers) and data links that can be used for sensing and/or control (e.g., the monitoring and control of a nuclear power plant).

[0066] Alternatively or additionally, the DPRA engine 110 can be configured to implement a defender action on the system model 102. A defender action can represent actions taken to prevent or mitigate attacker actions. As a non-limiting example, a defender action can represent actions taken by an operator of a nuclear power plant to prevent a cyber attack or respond to a cyber attack. As used herein, the term "operator" can refer to a "defender," of a facility, an operator of a facility, or both.

[0067] Alternatively or additionally, the DPRA engine 110 can be configured to generate a randomly-generated component failure scenario on the system model 102. The randomly generated component failure scenario can include simulating the failure of part or all of the physical system modelled by the system model.

[0068] Still with reference to FIG. 1A, the system 100 can further include an action model 120. The action model 120 can include one or more sub-models (not shown). Non-limiting examples of sub-models include a procedure model, an undetected attacker model, and a game model. The action model 120 can be configured to generate a plurality of actions for an operator to implement on the system model 102. Alternatively or additionally, the action model can be configured to generate a plurality of actions for an attacker to implement on the system model 102. As yet another example, the action model 120 can optionally include a lookahead game model to simulate the attacker anticipating future moves of the defender and/or the defender anticipating future moves of the attacker.

[0069] A non-limiting example of an action model 500 is illustrated in FIG. 5. The action model 500 can be used in conjunction with the nuclear power plant system model 300 shown in FIG. 3 and the example DPRA engine 400 shown in FIG. 4. As shown in FIG. 5, the example action model 500 can include a game model 502, an undetected attacker model 504, and a procedure model 506. An example game model 600 is illustrated in FIG. 6 and can be used as the game model 502 of FIG. 5., according to implementations of the present disclosure.

[0070] It should be understood that the system model 300, action model 500, and DPRA engine 400 in FIGS. 3-5 are intended only as non-limiting examples of systems and methods that can be used to implement the action model 120, system model 102, and DPRA engine 110 shown in FIG. 1A.

[0071] In some implementations, the action model 500 can be configured to generate both a plurality of actions for an attacker and a plurality of actions for a defender.

Optionally, the actions generated for the attacker and/or defender are based on a game theory analysis of the attacker and defender's actions.

[0072] It should be understood that in the "attacker," "defender," and "operator" actors described herein can be simulations of people acting in those roles, and/or game-theory models of actors (as opposed to human users). Thus, for example, the action model 120 shown in FIG. 1A, or the action model 500 shown in FIG. 5 can be configured to simulate the behaviors of human attackers, defenders, and/or operators in different scenarios. As described herein, the action model 120 can generate sets of actions that an be taken by the human attackers, defenders, and/or operators that are being simulated. The action model 120 can select among the sets of action, using game theory methods to determine which actions the attacker, defender, and/or operator will take and how the actions of the attacker, defender and/or operators can be related.

[0073] In some implementations, the system 100 can further include a post processing module 130. The post processing module 130 can receive information from DPRA engine 110. The post processing module 130 can be configured to determine a risk metric based on the information received from the DPRA engine 110. Alternatively or additionally, the risk metric determined by the post processing module 130 can be based the action model 120 and system model 102.

[0074] With reference to FIG. 1B, implementations of the present disclosure can use the system 100 described with reference to FIG. 1A to improve the cybersecurity of a network 150 (e.g., a nuclear power plant network). The network 150 can include any number of network devices 152, functional nodes 154, and data links 156 operatively connecting the network

devices 152 and/or functional nodes 154 together in any combination. The system 100 can perform the methods described herein (e.g., with reference to FIGS. 1A and 2, and example 1) to control one or more network devices 152 in the network 150 to improve cyber security. For example the network devices 152 can be controlled to suspend data links 156 between network devices 152 and/or functional nodes 154. Suspending data links 156 can improve cyber security by preventing an attacker from propagating an attack to new functional nodes. Thus, implementations of the present disclosure provide systems and methods for integrating both physical modeling of nuclear power plants and game theory to predict attacker actions, model the consequences of those actions, and control a network of the nuclear power plant to mitigate the effect of the attack.

[0075] With reference to FIG. 2, implementations of the present disclosure further include methods of performing probabilistic risk assessment, for example using the systems shown in FIG. 1A and 1B. At step 201 the method includes generating the system state using the system model. At step 202 the method includes providing the system state to the action model using the dynamic probabilistic risk assessment engine. At step 203, the method includes sampling the actions to be implemented on the system model from the actions generated by the action model and sampling component failures to be implemented on the system model using the dynamic probabilistic risk assessment engine. At step 204 the method includes implementing the sampled actions and component failures on the system model. At step 205, the method includes the collection of data generated during the process using the dynamic probabilistic risk assessment engine.

[0076] At step 210, the method 200 includes generating risk metrics using the post processing module in performing the dynamic probabilistic risk assessment corresponding to a system model.

[0077] At step 220, the method 200 further includes modelling actions in the system model using an action model.

[0078] In some implementations, the method 200 further includes modeling a plurality of actions for an attacker and a plurality of actions for a defender based on a game theory analysis of the attacker and defender's actions.

[0079] In some implementations, the method 200 further includes determining a risk metric based on the actions in the system.

[0080] It should be appreciated that the logical operations described herein with respect to the various figures may be implemented (1) as a sequence of computer implemented acts or program modules (i.e., software) running on a computing device (e.g., the computing device described in FIG. 7), (2) as interconnected machine logic circuits or circuit modules (i.e., hardware) within the computing device and/or (3) a combination of software and hardware of the computing device. Thus, the logical operations discussed herein are not limited to any specific combination of hardware and software. The implementation is a matter of choice dependent on the performance and other requirements of the computing device. Accordingly, the logical operations described herein are referred to variously as operations, structural devices, acts, or modules. These operations, structural devices, acts and modules may be implemented in software, in firmware, in special purpose digital logic, and any combination thereof. It should also be appreciated that more or fewer operations may be performed than

shown in the figures and described herein. These operations may also be performed in a different order than those described herein.

[0081] Referring to Fig. 7, an example computing device 1100 upon which the methods described herein may be implemented is illustrated. It should be understood that the example computing device 1100 is only one example of a suitable computing environment upon which the methods described herein may be implemented. Optionally, the computing device 1100 can be a well-known computing system including, but not limited to, personal computers, servers, handheld or laptop devices, multiprocessor systems, microprocessor-based systems, network personal computers (PCs), minicomputers, mainframe computers, embedded systems, and/or distributed computing environments including a plurality of any of the above systems or devices. Distributed computing environments enable remote computing devices, which are connected to a communication network or other data transmission medium, to perform various tasks. In the distributed computing environment, the program modules, applications, and other data may be stored on local and/or remote computer storage media.

[0082] In its most basic configuration, computing device 1100 typically includes at least one processing unit 1106 and system memory 1104. Depending on the exact configuration and type of computing device, system memory 1104 may be volatile (such as random access memory (RAM)), non-volatile (such as read-only memory (ROM), flash memory, etc.), or some combination of the two. This most basic configuration is illustrated in FIG. 7 by dashed line 1102. The processing unit 1106 may be a standard programmable processor that performs arithmetic and logic operations necessary for operation of the computing device 1100. The

computing device 1100 may also include a bus or other communication mechanism for communicating information among various components of the computing device 1100.

[0083] Computing device 1100 may have additional features/functionality. For example, computing device 1100 may include additional storage such as removable storage 1108 and non-removable storage 1110 including, but not limited to, magnetic or optical disks or tapes. Computing device 1100 may also contain network connection(s) 1116 that allow the device to communicate with other devices. Computing device 1100 may also have input device(s) 1114 such as a keyboard, mouse, touch screen, etc. Output device(s) 1112 such as a display, speakers, printer, etc. may also be included. The additional devices may be connected to the bus in order to facilitate communication of data among the components of the computing device 1100. All these devices are well known in the art and need not be discussed at length here.

[0084] The processing unit 1106 may be configured to execute program code encoded in tangible, computer-readable media. Tangible, computer-readable media refers to any media that is capable of providing data that causes the computing device 1100 (i.e., a machine) to operate in a particular fashion. Various computer-readable media may be utilized to provide instructions to the processing unit 1106 for execution. Example tangible, computer-readable media may include, but is not limited to, volatile media, non-volatile media, removable media and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules or other data. System memory 1104, removable storage 1108, and non-removable storage 1110 are all examples of tangible, computer storage media. Example tangible,

computer-readable recording media include, but are not limited to, an integrated circuit (e.g., field-programmable gate array or application-specific IC), a hard disk, an optical disk, a magneto-optical disk, a floppy disk, a magnetic tape, a holographic storage medium, a solid-state device, RAM, ROM, electrically erasable program read-only memory (EEPROM), flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices.

[0085] In an example implementation, the processing unit 1106 may execute program code stored in the system memory 1104. For example, the bus may carry data to the system memory 1104, from which the processing unit 1106 receives and executes instructions. The data received by the system memory 1104 may optionally be stored on the removable storage 1108 or the non-removable storage 1110 before or after execution by the processing unit 1106.

[0086] It should be understood that the various techniques described herein may be implemented in connection with hardware or software or, where appropriate, with a combination thereof. Thus, the methods and apparatuses of the presently disclosed subject matter, or certain aspects or portions thereof, may take the form of program code (i.e., instructions) embodied in tangible media, such as floppy diskettes, CD-ROMs, hard drives, or any other machine-readable storage medium wherein, when the program code is loaded into and executed by a machine, such as a computing device, the machine becomes an apparatus for practicing the presently disclosed subject matter. In the case of program code execution on programmable computers, the computing device generally includes a processor, a storage

medium readable by the processor (including volatile and non-volatile memory and/or storage elements), at least one input device, and at least one output device. One or more programs may implement or utilize the processes described in connection with the presently disclosed subject matter, e.g., through the use of an application programming interface (API), reusable controls, or the like. Such programs may be implemented in a high level procedural or object-oriented programming language to communicate with a computer system. However, the program(s) can be implemented in assembly or machine language, if desired. In any case, the language may be a compiled or interpreted language and it may be combined with hardware implementations.

[0087] **Examples**

[0088] The following examples are put forth so as to provide those of ordinary skill in the art with a complete disclosure and description of how the compounds, compositions, articles, devices and/or methods claimed herein are made and evaluated, and are intended to be purely exemplary and are not intended to limit the disclosure. Efforts have been made to ensure accuracy with respect to numbers (e.g., amounts, temperature, etc.), but some errors and deviations should be accounted for. Unless indicated otherwise, parts are parts by weight, temperature is in °C or is at ambient temperature, and pressure is at or near atmospheric.

[0089] **Example 1**

[0090] An example implementation of the present disclosure was designed and tested to evaluate nuclear power plant cyber security risk. The example implementation of the present disclosure includes a system where “attackers” and “defenders” are modeled in an evolving system including a physics-based model of the system, stochastic events, and a game

theory analysis. The example implementation was implemented according to the system 100 shown in FIG. 1A. In the example implementation, the system 100 was configured to use a full-scale nuclear power plant simulator (GSE) physics model, and a GNS3 network simulator co-simulated with the physics model, and a reduced-order model of a PWR (pressurized water reactor).

[0091] FIG. 8 illustrates an example implementation of the present disclosure including a model 800 of the strategic interactions between attackers/defenders in a system including physical components. The example implementation models an attacker 802, a defender 804 (e.g., the operator of a network) and an industrial control system 810 ("ICS"). The industrial control system 810 includes digital components 812 (e.g., software running on networked devices) and physical components 814 (e.g., controllers, computing devices, sensors, actuators, etc.). The industrial control system 810 is configured to sense and control physical components of a system 820 (e.g., any or all of the components of the nuclear reactor shown in FIG. 14, and/or any other industrial system).

[0092] With reference to FIG. 9, an example method of analyzing the system of FIG. 8 is shown according to implementations of the present disclosure. The method includes analyzing the actions taken by the attacker 802 and defender 804 that affect the digital components 812 of the ICS. The actions of the attacker 802 and defender 804 on the digital components 812 can affect the physical (e.g., mechanical) components 814 of the ICS, and in turn impact the system 820. The example method can therefore include analyzing both the interactions between the attacker 802 and defender 804, as well as the subsequent effects that the actions taken by the attacker 802 and defender 804 have on the entire system by modeling

the effect of attacks on the digital components, physical components, and the system 820. It should be understood that the steps shown in FIG. 9 can be performed iteratively. For example, the actions taken by the attacker 802 and defender 804 can be based on the state of the system 820, so that moves made in one iteration of the method can subsequently effect further iterations of the method. Implementations of the present disclosure can therefore improve analysis of cyber attacks and defenses on complicated physical systems 820 by modeling the ongoing evolution of the attack over time.

[0093] At any time t, the overall system state of the NPP is represented using the tuple ($\bar{x}, i, d, a, t$).

[0094] $\bar{x}$ - A vector representing the physical state of a nuclear power plant.

[0095] Vector of physical variables such as pressure, flowrate, temperature etc.

[0096] $\bar{x} \in \mathbb{R}^n$ is a point in continuous space with its boundaries determined by system physics.

[0097] $\mathbb{X} \subset \mathbb{R}^n$ represents the space of all possible physical state vectors $\bar{x}$.

[0098] i - The states of the components in the NPP are represented using a discrete vector. $\mathbb{I}$ represents the set of all possible component states.

[0099] The couple (d, a) represents the latest pair of actions taken by the defender and the attacker respectively. $D = \{d_1, d_2, d_3 \ldots\}$ is the defender's action space. $A = \{a_1, a_2, a_3 \ldots\}$ is the attacker's action space. The example implementation assumes that the defender and the attacker action spaces are discrete.

[00100] $\pi(\bar{x}, i, d, a, t) =$ the probability density of the overall system state $(\bar{x}, i, d, a, t)$ at time t.

[00101] The example implementation further includes a description of continuous event trees.

[00102] Probability that the system is in unsafe operating region at time $t = \sum_{d,a}\sum_{i}\int_{\mathbb{X}_U}\pi(\bar{x}, i, d, a, t)d\bar{x}$, where $\mathbb{X}_U$ is the Unsafe region in state space

[00103] i - The states of the components are represented using a discrete vector.

[00104] For example, the study can use integers to represent discrete states of components. 0 - Normal, 1 - Failed, 2 - Compromised.

[00105] For the example system, $i = [i_1, i_2, i_3]$, where $i_1, i_2 \in \{0,1,2\}, i_3 \in \{0,1\}$

[00106] Assume that the attacker actions are: $a_1, a_2 \in A$, where a_1 is compromise the controller, and a_2 is compromise the sensor.

[00107] If the components are initially in the state $i^1 = [0,0,0]$ and the attacker takes action a_1, then the system state changes to $i^2 = [2,0,0]$.

[00108] FIG. 10 illustrates an example diagram of a controller, digital flow sensor, and mechanical valve that can each be assigned discrete states, according to implementations of the present disclosure.

[00109] The example implementation of the present disclosure further includes a description of trajectories describing the system's evolution in physics state space represented by a set of differential equations.

[00110] $\frac{\partial}{\partial t}\bar{x} = f_i(\bar{x}, t)$

[00111] Assume that the solution to the above equation is $\bar{x}(t) = g_i(t, \bar{x}_0)$.

[00112] Trajectory is dependent on the component state vector i, which in turn depends on physics (component failures), and the actions of the defender and the attacker (d, a):

$$\pi(\bar{x}, i, d, a, t) = \left[\int_{\mathbb{X}} \pi(\bar{u}, i, d, a, 0) \times \delta[\bar{x} - g_i(t, \bar{u})] \times \left(e^{-\int_0^t \lambda_{i,d,a}[g_i(s,\bar{u})]ds}\right) d\bar{u}\right]$$
$$+ \left[\sum_{j \neq i} \int_{\mathbb{X}} \left\{\int_0^t p(j \rightarrow i \mid \bar{u}) \times \pi(\bar{u}, j, d, a, \tau) \times \delta[\bar{x} - g_i(t - \tau, \bar{u})] \times \left(e^{-\int_\tau^t \lambda_{i,d,a}[g_i(s,\bar{u})]ds}\right) d\tau\right\} d\bar{u}\right]$$
$$+ \left[\sum_{dd', a' \neq d,a} \sum_j \int_{\mathbb{X}} \left\{\int_0^t \pi(\bar{u}, j, d', a', \tau) \times p(d', a' \rightarrow d, a \mid \bar{u}, j, \tau) \times p(j \rightarrow i \mid d, a, \bar{u}) \times \delta[\bar{x} - g_i(t - \tau, \bar{u})] \times \left(e^{-\int_\tau^t \lambda_{i,d,a}[g_i(s,\bar{u})]ds}\right) d\tau\right\} d\bar{u}\right]$$

[00113] FIG. 11 illustrates an example trajectory in physics space, as described herein, and FIG. 12 illustrates a model of an example trajectory.

[00114] The example implementation further includes game theory modeling of actions in the system:

$$\sum_{d', a' \neq d,a} \sum_j \int_{\mathbb{X}} \left\{\int_0^t \pi(\bar{u}, j, d', a', \tau) \times p(d', a' \rightarrow d, a \mid \bar{u}, j, \tau) \times p(j \rightarrow i \mid d, a, \bar{u}) \times \delta[\bar{x} - g_i(t - \tau, \bar{u})] \times \left(e^{-\int_\tau^t \lambda_{i,d,a}[g_i(s,\bar{u})]ds}\right) d\tau\right\} d\bar{u}$$

[00115] As used herein, "game theory" refers to how the attacker and defender behave in their respective objectives, where $p(d', a' \rightarrow d, a \mid \bar{u}, j, \tau)$ represents the probability that the defender and attacker take new actions d, a at time τ,given the physical system state $\bar{u}$ and component state j. As described herein, game theory can quantify this value based on the strategic interactions between one or more "players" (e.g., one or more attackers and one one or more defenders).

[00116] An example game theory model is shown in FIG. 13, where $S, \{A, D\}, \{\pi^a, \pi^d\}, P, \{R^a, R^d\}, \gamma$. As shown in FIG. 13, the model includes two players: attacker and defender, where:

[00117] $S = \mathbb{X} \times \mathbb{I}$ is the set of system states.

[00118] $D = \{d_1, d_2, d_3 \ldots\}$ is the defender's action space.

[00119] $A = \{a_1, a_2, a_3 \ldots\}$ is the attacker's action space.

[00120] π^d and π^a are the action policies of the defender and the attacker.

[00121] $P: S \times D \times A \times S \rightarrow [0,1]$ is the state transition probability mapping.

[00122] $R^d: S \times D \times A \times S \rightarrow \mathbb{R}$ is the reward function of the defender.

[00123] $R^a: S \times D \times A \times S \rightarrow \mathbb{R}$ is the attacker's reward function.

[00124] V^d, V^a - Value functions representing the goodness of the state. These are the expected cumulative rewards by following a given policy from the current state.

[00125] In the example implementation, the model of continuous events in the game theory simulation includes:

$$V^{\alpha}(\bar{x}, i, d, a, t) = r^{\alpha}_{action}(\bar{x}, i, \text{action}^{\alpha}, t) + \sum_{j} p(i \rightarrow j \mid d, a, \bar{x}) \times [r^{\alpha}_{\text{transition}}(i, j)] + \sum_{j} p(i \rightarrow j \mid d, a, \bar{x}) \times \sum_{k} \sum_{d', a'} \int \left\{ \int_{t}^{t_{\text{mission}}} \pi(\bar{u}, \boldsymbol{k}, \boldsymbol{d}, \boldsymbol{a}, \tau \mid \bar{x}, j, \boldsymbol{d}, \boldsymbol{a}, \boldsymbol{t}) \times p(d, a \rightarrow d', a' \mid \bar{u}, k, \tau) \times V^{\alpha}(\bar{u}, k, d', a', \tau) d\tau \right\} du$$

[00126] $r^{\alpha}_{\text{action}}(\bar{x}, i, \text{action}^{\alpha}, t)$ is the cost incurred by the player α (defender or attacker) for taking the action $^{\alpha}(d, a)$ at the physical system state $\bar{x}$, the component state i and time t.

[00127] $\sum_j p(i \rightarrow j \mid d, a, \bar{x}) \times [r^{\alpha}_{\text{transition}}(i, j)]$ - This is the expected immediate reward received by the player α when there is a transition out of the component state i due to the pair of actions (d, a). The term $r^{\alpha}_{\text{transition}}(i, j)$ is the immediate reward received by the

player α, when the component state transitions from i to j due to the pair of actions (d, a), where $p(i \rightarrow j \mid d, a, \bar{x})$ is the probability of that transition.

[00128] The third term represents the recursive equation to compute rewards from future states.

[00129] Equilibrium concepts, for example the concept of mixed strategy Nash Equilibrium can be used to compute the policies of the attacker and the defender given the value functions. In a Nash Equilibrium, no player would unilaterally deviate from his/her strategy at the equilibrium.

[00130] The example implementation can further include simulating consequences (e.g., reactor trips) when cyberattacks are successfully conducted.

[00131] As described with reference to FIG. 1A, the example implementation can be used to model a system including a network with sensors, computers, and controllers that are linked together by a network with one or more data links. Attacker actions can include disrupting one or more of the data links of the network. Additional non-limiting examples of attacker actions include tampering and/or blocking data from different functional nodes. Defender actions can include suspending data links (e.g. from functional nodes that are being tampered with).

[00132] FIG. 14 illustrates an example case study performed of an implementation of the present disclosure. FIG. 14 includes a PWR model with three steam generators implemented on the GPWR by GSE systems is used as the system model, where a Loss of Feedwater event is considered. The PWR model includes three steam generators and feedwater loops, where each steam generator is supplied feedwater through a main feedwater

valve, the feedwater valves are controlled by local valve controllers (VC1, VC2, VC3) which receive control signals from the main computers (MC) and backup computers (BC). The MC and BC receive sensor signals from three flowmeters, one from each loop. (SENS1, SENS2, SENS3)

[00133] To compute the control signals for the valves, the MC and BC use signals from respective sensors. The sensors are connected to the MC and BC through two switches (SW1, SW2), and similarly the valve controllers are connected to the MC and BC through two switches (SW3, SW4). A total of 16 links are considered between the sensors, computers and valve controllers. It should be understood that the case study shown in FIG. 14 is intended only as a non-limiting example, and that different types of reactor and network can be modeled according to implementations of the present disclosure.

[00134] FIG. 15 illustrates an example simulation of the case study shown in FIG. 14.

[00135] FIG. 16 illustrates example states of components used in the case study.

[00136] The action space for the defender in the case study, includes:

[00137] 1-16: Deactivate links 1-16

[00138] 17: Change the valve controllers' state from 1 to 2 (e.g., start using input from the backup computer)

[00139] 18: No action.

[00140] The action space for the attacker in the case study includes:

[00141] 1: Compromise flow sensor 1

[00142] 2: Compromise flow sensor 2

[00143] 3: Compromise flow sensor 3

[00144] 4: Compromise Main Computer

[00145] 5: Compromise Backup Computer

[00146] 6: Compromise Valve controller 1

[00147] 7: Compromise Valve Controller 2

[00148] 8: Compromise Valve Controller 3

[00149] 9: No action.

[00150] The case study assumed:

[00151] 1. The attacker is already in the system and is trying to initiate a Loss of Feedwater event.

[00152] 2. The attacker can successfully compromise any component initially with probability 0.125 .

[00153] 3. If the attacker tries to compromise a component the probability of success is 0.3 if a neighboring component is already compromised, else it is 0.125 .

[00154] 4. Once a component is compromised, it remains in that state until the end of mission time.

[00155] 5. Once a link is deactivated, it remains in that state until the end of mission time.

[00156] *Results*

[00157] The simulation case study was run with a time of 240 seconds in 10-second time steps.

[00158] The trajectories of narrow water range levels in each of the three steam generators are illustrated in FIGS. 17A-17C, where FIG. 17A illustrates the water level in steam generator 1, FIG. 17B illustrates the water level in steam generator 2, and FIG. 17C illustrates the water level in steam generator 3.

[00159] The simulation case study used a Mixed-Strategy Nash Equilibrium to compute the attacker and defender strategies.

[00160] In the case study, the reactor was tripped once in thirty scenarios.

[00161] While the case study was employed for modeling a nuclear reactor, it should be understood that any kind of industrial network can be modeled and evaluated for cyber security. The present disclosure contemplates that, for example, firewalls, network diodes, and other network components can be modeled and evaluated according to implementations of the present disclosure. Additionally, it should be understood that the attacker and defender actions described herein are non-limiting examples.

[00162] Finally, it should be understood that implementations of the present disclosure can be used to control the performance of industrial networks (e.g., nuclear power plant controls). For example, implementations of the present disclosure can be used to evaluate an ongoing attack, and/or model the performance of potential attacker and defender actions to improve the cyber security of a power plant in real time.

[00163] **Example 2:**

[00164] A study was performed of an example implementation of the present disclosure shown in FIG. 1A.

[00165] The study included configuring the example implementation for dynamic probabilistic risk assessment (DPRA) configured to incorporate the stochastic nature of the operator's as well as the attacker's actions using a game theory based framework in addition to component failures, and study their effects on the evolution of the system. Hence, the example DPRA simulation architecture is configured to include the three below elements:

[00166] (1) The system model to emulate the evolution of the system under consideration, in this case a nuclear power plant.

[00167] (2) The DPRA engine, that monitors the state of the system, and implements the sampled attacker and defender actions as well as randomly generated component failure scenarios [15].

[00168] (3) The action model which generates a set of possible operator and attacker actions to be implemented on the system. In addition to the system model and the DPRA engine, the example implementation explicitly considers an action model that generates possible defender and attacker actions to be implemented on the system model using a game theory based analysis. The action model and its components are explained in this section.

[00169] Additionally, a post processing module can be used to analyze the large amounts of data generated during the simulations and compute meaningful risk metrics. FIG. 1A depicts a high-level schematic of the DPRA simulation architecture used to perform cybersecurity risk analysis.

[00170] The study included configuring the implementation of the DPRA engine 400 shown in FIG. 4. The DPRA engine was configured so that the data collection module 402 of the DPRA engine 400 receives system physical state information (e.g., a vector of relevant

physical variables, and the current component state information) from the system model. This information can be disseminated to the component failure models element 404 of the DPRA engine and the action model 500. Using this information as input, the component failure models element 404 updates the failure rates and the failure modes of the NPP components and transmits the updated values back to the data collection module 402. Similarly, the action model 500 returns a set of possible attacker and defender actions to the data collection module 402. The output of the component failure models element 404, and the action model 500 received by the data collection module 402 is transmitted to the sampler. The next state transition time, the next state transition to be implemented on the system model from the set of failure modes generated by the component failure models and the next operator and attacker actions to be enforced on the system model form the inputs received from the action model are sampled by the sampler element 406 of the DPRA engine 400 and provided to the data collection module 402. The data collection module 402 then shares this information with the scheduler 408, which then implements the sampled transitions and the sample actions on the system model 102. The data collection module assimilates all the data generated by all the elements of the DPRA architecture. This data can be sent to the post processing module for analysis and computation of relevant risk metrics.

[00171] The example study further included configuring the system model 300 shown in FIG. 3. The system model 300 includes mathematical models of the components in the physical systems layer under different states, mathematical models of the controllers, and the models of other network elements. The system model 300 can be configured to simulate the evolution of the physical system when subjected to stochastic events including component

failures and cyber-attacks. The example implementation included an abnormal event detection and classification system 302 [16] that can detect an abnormal event and differentiate it as either a safety event caused by component failure or a cyber-attack. This can be used to explore scenarios of undetected attackers and evaluate the associated risks, according to implementations of the present disclosure. Implementations of the present disclosure can further be configured to evaluate scenarios in which a cyber-attack is undetected, resulting in the operator taking incorrect actions, which can therefore generate risks.

[00172] The study further included configuring the example action model 500 shown in FIG. 5. The action model 500 can receive system monitoring information (e.g., information about the system state as defined by the vector of a list of physical variables and vector of component states) and generate a set of possible attacker and operator actions. As shown in FIG. 5, the example action model 500 can include three individual models, the procedure model 502, the undetected attacker model 504, and the game model 506. The study assumed that the operator is procedure following and is not prone to errors. The procedure model 502 is used to generate possible operator actions when no cyberattack is detected by the abnormal event detection module in the system model. The undetected attacker model 504 is used to represent and model initial launching of cyber-attacks and the scenarios in which a cyber-attack remains undiagnosed.

[00173] The study further included configuring the game model 600 shown in FIG. 6. The action model 500 can switch to the game model 600 when a cyberattack is detected in the system. In the presence of a cyber-attack, the operator takes the role of a defender. The components of the game model 600 are the game itself of which the attacker and defender(s)

are the players, and a spinoff DPRA process. The optimal attacker and defender actions can depend on the physics of the system. The game model 600 can consider the computation of optimal actions based on physics and implement those on the system to observe it's evolution. So the example implementation can simulate (e.g., by running a parallel model to predict the system behavior under different pairs of attacker and defender actions to a certain time point in the future and use that knowledge to inform the attacker and defender policies in the original DPRA process.

[00174] The study further included an analysis of continuous event trees for cybersecurity risk analysis, which can be implemented using the system 100 shown in FIG. 1A.

[00175] The study included a mathematical formulation of continuous event trees [8] for cybersecurity risk analysis that can be applied using the system 100 of FIG. 1A.

[00176] An example nuclear power plant system was modeled as follows in the example implementation studied:

[00177] The physical state of a nuclear power plant can be represented using a state vector, $\bar{x}$, a vector of physical variables such as pressure, flowrate, temperature etc. It is implicit that $\bar{x} \in \mathbb{R}^n$ is a point in continuous space with its boundaries determined by the physics. Let $\mathbb{X} \subset \mathbb{R}^n$ represent the space of all possible physical state vectors $\bar{x}$.

[00178] The states of the components in the NPP can be represented using a vector i, a vector on a discrete space. The example implementation uses discrete component states (e.g., normal or failed states). Integers can be used to represent such discrete states of components. For example, 0, 1 and 2 can be used to denote that a component is in normal state or failed state or compromised state respectively. Consider an example system with three

components, a digital controller, a digital sensor and a mechanical valve. Then $i = [i_1, i_2, i_3]$ where i_1, i_2, and i_3 represent the states of the controller, the sensor and the valve respectively. Here, $i_1, i_2 \in \{0,1,2\}$ and $i_3 \in \{0,1\}$ because the digital controller and the sensor can be compromised whereas the mechanical valve cannot be subjected to cyber-attacks. Let $\mathbb{C}$ represent the set of all possible component states.

[00179] $D = \{d_1, d_2, d_3 \ldots\}$ is the defender's action space and $A = \{a_1, a_2, a_3 \ldots\}$ is the attacker's action space. The study assumes that the defender and the attacker action spaces are discrete, with actions such as switch from main controller to backup controller or compromise the controller and shut down the pump etc. Continuous action spaces can also be used in implementations of the present disclosure. Feasible defender and attacker actions can depend on the physical state as well as the state of components. For example, the example implementation can assume that a component that has been compromised by the attacker cannot be compromised again and remains in that state, until it is "repaired." Additionally, certain actions may not be physically possible depending on the physical state of system, and implementations of the present disclosure can model the physically possible actions of the system. For example, the speed of a pump cannot be reduced to zero instantaneously.

[00180] At any time t, the overall system state of the NPP can be represented using the tuple $(\bar{x}, i, d, a, t)$, where the physical state vector $\bar{x}$ represents the physical system state, the component state is represented by the vector i, and the couple (d, a) represents the latest pair of actions taken by the defender and the attacker respectively. For the remainder of this example, the expressions physical state vector and physical state will be used

interchangeably. The same is true for the expressions component state vector and component state.

[00181] A system of differential equations as presented by equation (1) can be used to represent the trajectory of the NPP in the physical state space [8].

[00182] $$\frac{\partial}{\partial t}\bar{x} = f_i(\bar{x}, t)$$

[00183] It is implicit that these differential equations are dependent on the states of the components i.e., when the states of components change, the differential equations representing the dynamics of the system will change and the system follows a different trajectory in the state space, as shown in FIG. 11.

[00184] The component state i can change either due to random component failures or due to the actions of the defender and the attacker. Additionally, the defender and attacker may not be able to directly interfere with the physical process and may only change the component state vector, which in turn will change the trajectory of system evolution i.e., the trajectory of the reactor in the physical state space is conditionally independent of the actions of the attacker and the defender given the component state vector. In the example depicted in FIG. 11, initially the system is in the state ($\bar{x}_0, i, d_0, a_0, t_0$), and evolves along the trajectory defined by f_i until time τ, where the physical state vector is $\bar{x}_1$. The defender and the attacker then take actions (d_1, a_1) at $(\bar{x}_1, \tau)$ and the component state vector transitions from i to j as a result, following which the system evolves along the trajectory defined by f_j. The study assumes that the component state transitions due to attacker and defender actions if any, are instantaneous, but the present disclosure contemplates that non-instantaneous transitions can be modeled in implementations of the present disclosure.

[00185] Let equation (2) represent the solution to the system of differential equations presented in equation (1), where $\bar{x}_0$ is the initial condition [8]. It is implicit that for time $t = 0, \bar{x}_0 = g_i(0, \bar{x}_0)$. It is also implicit that f_i and g_i represent the same trajectories in the physical state space given the component state vector i.

$$\bar{x}(t) = g_i(t, \bar{x}_0) \tag{2}$$

[00186] Let the probability density of the overall system state ($\bar{x}, i, d, a, t$) at time t be $\pi(\bar{x}, i, d, a, t)$. Let the conditional probability density that the system is in state $(\bar{x}, j, d, a, t)$ at time t, given the initial state ($\bar{x}_0, i, d_0, a_0, t_0$) be denoted by $\pi(\bar{x}, j, d, a, t \mid \bar{x}_0, i, d_0, a_0, t_0)$ [8]. It is implicit that:

$$\pi(\bar{x}, j, d, a, t_0 \mid \bar{x}_0, i, d_0, a_0, t_0) = \delta(\bar{x} - \bar{x}_0) \times \delta_{ij} \times \delta_{(d_0,a_0),(d,a)} \tag{3}$$

[00187] where δ is the Dirac delta function, and δ_{ij} and $\delta_{(d_0,a_0),(d,a)}$ are Kronecker delta functions. An objective of the study is to compute the value $\pi(\bar{x}, i, d, a, t)$ and to consequently estimate the probability that the physical state of the reactor is in a certain region of the state space at any given instant of time.

[00188] As discussed above, the trajectories in physical state space are determined by the component state vector. The component state vector can change either due to random failures of components or due to the actions of the attacker and the defender. In summary, the trajectories in the physical state space are dependent on the substate (i, d, a).

[00189] Let $\lambda_{i,d,a}(\bar{x})\Delta t$ be the conditional probability that there is a transition out of the substate (i, d, a) in the interval Δt, when the system is in state ($\bar{x}, i, d, a, t$). The failure rates of the components are dependent on the physical state $\bar{x}$. Let $p_s(i \to j \mid \bar{x})\Delta t$ be the conditional probability that the component state transitions from i to j in the interval Δt

explicitly due to random component failures when the physical state vector is $\bar{x}$ and the defender and attacker take no new actions. Additionally, the physical state $\bar{x}$ influences the actions of the defender as well as the attacker. Let $p_c(i \to j \mid d', a', \bar{x})\Delta t$ be the conditional probability that the component state transitions from i to j in the interval Δt when the defender and attacker take new actions d' and a' respectively at the physical state $\bar{x}$. The relation between $\lambda_{i,d,a}(\bar{x}), p_s(i \to j \mid \bar{x})$ and $p_c(i \to j \mid d', a', \bar{x})$ is presented in equation (4) where $p(d, a \to d', a')$ is the probability that the defender and the attacker take new actions d', a' respectively.

$$\lambda_{i,d,a}(\bar{x}) = \sum_{j \neq i} p_s(i \to j \mid \bar{x}) + \sum_{d', a' \neq d, a} p(d, a \to d', a' \mid \bar{x}) \times \sum_{j} p_c(i \to j \mid d', a', \bar{x}) \quad (4)$$

[00190] If the overall system state is $(\bar{x}_0, i, d, a, t_0)$ initially, then the probability density that the system reaches the physical state $\bar{x}$ at time t while remaining in the substate (i, d, a) until time t is given by the product $\delta[\bar{x} - g_i(t - t_0, \bar{x}_0)] \times e^{-\int_{t_0}^{t} \lambda_{i,d,a}[g_i(s,\bar{x}_0)]ds}$ where the term $e^{-\int_{t_0}^{t} \lambda_{i,d,a}[g_i(s,\bar{x}_0)]ds}$ represents the probability that the system remains in the state (i, d, a) and consequently evolves along the trajectory defined by g_i during the interval $[t_0, t]$ and the term $\delta[\bar{x} - g_i(t - t_0, \bar{x}_0)]$ represents the probability density that $(\bar{x}, t)$ is the only point reachable from $(\bar{x}_0, t_0)$ along the trajectory defined by g_i[8].

[00191] The study further included determining the equation of an extended continuous event tree.

[00192] Equation (5) presents the integral form of the continuous event tree equation based on the Chapman Kolmogorov equation [8] extended to the cyber-attack case to compute the probability density that the system is in state $(\bar{x}, i, d, a, t)$ at time t.

$$\pi(\bar{x}, i, d, a, t) \tag{5}$$
$$= \left[\int_{\mathbb{X}} \pi(\bar{u}, i, d, a, 0) \times \delta[\bar{x} - g_i(t, \bar{u})] \times \left(e^{-\int_0^t \lambda_{i,d,a}[g_i(s,\bar{u})]ds}\right) d\bar{u}\right]$$
$$+ \left[\sum_{j \neq i} \int_{\mathbb{X}} \left\{\int_0^t p(j \right.\right.$$
$$\left.\left. \rightarrow i \mid \bar{u}) \times \pi(\bar{u}, j, d, a, \tau) \times \delta[\bar{x} - g_i(t-\tau, \bar{u})] \times \left(e^{-\int_\tau^t \lambda_{i,d,a}[g_i(s,\bar{u})]ds}\right) d\tau\right\} d\bar{u}\right]$$
$$+ \left[\sum_{d',a' \neq d,a} \sum_j \int_{\mathbb{X}} \left\{\int_0^t \pi(\bar{u}, j, d', a', \tau) \times p(d', a' \rightarrow d, a \mid \bar{u}, j, \tau) \times p(j \right.\right.$$
$$\left.\left. \rightarrow i \mid d, a, \bar{u}) \times \delta[\bar{x} - g_i(t-\tau, \bar{u})] \times \left(e^{-\int_\tau^t \lambda_{i,d,a}[g_i(s,\bar{u})]ds}\right) d\tau\right\} d\bar{u}\right]$$

[00193] The equation has the following 3 major parts (as separated by the square parentheses):

[00194] (1) $\int_{\mathbb{X}} \pi(\bar{u}, i, d, a, 0) \times \delta[\bar{x} - g_i(t, \bar{u})] \times \left(e^{-\int_0^t \lambda_{i,d,a}[g_i(s,\bar{u})]ds}\right) d\bar{u}$ - This integrand represents the probability density that the system is initially in the state ($\bar{u}, i, d, a, 0$), and reaches the physical state $\bar{x}$ at time t while remaining in the substate (i, d, a) until time t, along the trajectory defined by g_i. While the integral is computed over the entire physical state space $\mathbb{X}$, the Dirac delta function $\delta[\bar{x} - g_i(t, \bar{u})]$ ensures that only a specific subset of appropriate $\bar{u}$ values are valid.

[00195] (2) $\sum_{j \neq i} \int_{\mathbb{X}} \left\{\int_0^t p(j \rightarrow i \mid \bar{u}) \times \pi(\bar{u}, j, d, a, \tau) \times \delta[\bar{x} - g_i(t-\tau, \bar{u})] \times \left(e^{-\int_\tau^t \lambda_{i,d,a}[g_i(s,\bar{u})]ds}\right) d\tau\right\} d\bar{u}$ – The inner most integrand in the second part represents the probability density that the system is in state ($\bar{u}, j, d, a, \tau$) at some intermediate time τ between 0 and t, when the component state transitions from j to i due to a random event and not due to attacker and defender actions, and after that the system evolves along the trajectory defined by g_i while remaining in the state (i, d, a) from time τ to t, and eventually reaches the physical system state $\bar{x}$ at time t. The sum of this probability density over all possible ($\bar{u}, j, \tau$) is

computed. The second part of equation (5) is recursive in nature, similar to the Chapman-Kolmogorov equation [8].

[00196] (3) $\sum_{d',a' \neq d,a} \sum_j \int_{\mathbb{X}} \left\{ \int_0^t \pi(\bar{u}, j, d', a', \tau) \times p(d', a' \rightarrow d, a \mid \bar{u}, j, \tau) \times p(j \rightarrow i \mid d, a, \bar{u}) \times \delta[\bar{x} - g_i(t-\tau, \bar{u})] \times \left(e^{-\int_\tau^t \lambda_{i,d,a}[g_i(s,\bar{u})]ds} \right) d\tau \right\} d\bar{u}$ – The inner most integrand of the third term represents the probability density that the system is in state ($\bar{u}, j, d', a', \tau$) at some time τ between 0 and t, at which point the defender and the attacker take new actions (d, a) as a result of which the component state transitions from j to i and subsequently the system evolves along the trajectory defined by g_i, while remaining in the substate (i, d, a) from time τ to t and eventually arrives at the physical system state $\bar{x}$ at time t. The probability that the defender and the attacker take a new pair of actions $p(d', a' \rightarrow d, a \mid \bar{u}, j, \tau)$ i.e., the mixed equilibrium strategies of the defender and the attacker at physical state $\bar{u}$, component state j and time τ can be computed using a game theory based approach. It can also be observed that the third term of equation (5) is recursive in nature as well.

[00197] The conditional probability density $\pi(\bar{x}, i, d, a, t \mid \bar{x}_0, i_0, d_0, a_0, t_0)$ that the system is in the state $(\bar{x}, i, d, a, t)$ at time t given that it was initially in the state ($\bar{x}_0, i_0, d_0, a_0, t_0$) can be computed using equation (6) (see equation (3)).

$$\pi(\bar{x}, i, d, a, t \mid \bar{x}_0, i_0, d_0, a_0, t_0) \tag{5}$$

$$= \left[\int_{\mathbb{X}} \delta[\bar{x}_0 - \bar{u}] \times \delta_{ii_0} \times \delta_{(d_0,a_0)(d,a)} \times \delta[\bar{x} - g_i(t - t_0, \bar{u})] \times \left(e^{-\int_0^t \lambda_{i,d,a}[g_i(s,\bar{u})]ds}\right) d\bar{u}\right]$$

$$+ \left[\sum_{j \neq i} \int \left\{\int_{t_0}^{t} p(j \rightarrow i \mid \bar{u}) \times \pi(\bar{u}, j, d, a, \tau \mid \bar{x}_0, i_0, d_0, a_0, t_0) \times \delta[\bar{x} - g_i(t - \tau, \bar{u})] \times \left(e^{-\int_\tau^t \lambda_{i,d,a}[g_i(s,\bar{u})]ds}\right) d\tau\right\} d\bar{u}\right]$$

$$+ \left[\sum_{d,a \neq d,a} \sum_{j} \int_{\mathbb{X}} \left\{\int_{t_0}^{t} \pi(\bar{u}, j, d', a', \tau \mid \bar{x}_0, i_0, d_0, a_0, t_0) \times p(d', a' \rightarrow d, a \mid \bar{u}, j, \tau) \times p(j \rightarrow i \mid d, a, \bar{u}) \times \delta[\bar{x} - g_i(t - \tau, \bar{u})] \times \left(e^{-\int_\tau^t \lambda_{i,d,a} | g_i(s,\bar{u})| ds}\right) d\tau\right\} d\bar{u}\right]$$

[00198] Equation (7) presents the expected cumulative rewards received by the players i.e., the defender and the attacker. Here α is used as an index and is not an exponent. $R^1(\bar{x}, i, d, a, t)$ and $R^2(\bar{x}, i, d, a, t)$ are the expected cumulative rewards received by the defender and the attacker respectively when the defender takes an action d and the attacker takes an action a at the physical system state $\bar{x}$, the component state i and time t. These rewards are used to compute the mixed strategies of the defender and the attacker using a game theory based approach [7]. The computation of mixed strategies is not explicitly discussed herein.

$$R^{\alpha}(\bar{x}, i, d, a, t) \tag{7}$$

$$= r^{\alpha}_{\text{action}}(\bar{x}, i, \text{ action }^{\alpha}, t) + \sum_j p(i \rightarrow j \mid d, a, \bar{x}) \times [r^{\alpha}_{\text{transition}}(i, j)]$$

$$+ \sum_j p(i \rightarrow j \mid d, a, \bar{x}) \times \sum_k \sum_{d,a,} \int_{\mathbb{X}} \left\{\int_t^{t_{mission}} \pi(\bar{u}, k, d', a', \tau \mid \bar{x}, j, d, a, t) \times R^{\alpha}(\bar{u}, k, d', a', \tau) d\tau\right\} d\bar{u}$$

[00199] Equation (5) can be a recursive equation as well and the three terms in equation (5) are explained below:

1. $r^{\alpha}_{\text{action}}(\bar{x}, i, \text{action }^{\alpha}, t)$: $r^{1}_{\text{action}}(\bar{x}, i, d, t)$ is the cost incurred by the defender for taking the action d at the physical system state $\bar{x}$, the component state i and time t,

while $r^2_{action}(\bar{x}, i, a, t)$ is the cost incurred by the attacker for taking the action a at the physical system state $\bar{x}$, the component state i and time t.

2. $\sum_j p(i \to j \mid d, a, \bar{x}) \times [r^{\alpha}_{\text{transition}}(i,j)]$ is the expected immediate reward received by the player α when there is a transition out of the component state i due to the pair of actions (d, a). The term $r^{\alpha}_{\text{transition}}(i,j)$ is the immediate reward received by the player α, when the component state transitions from i to j due to the pair of actions (d, a), where $p(i \to j \mid d, a, \bar{x})$ is the probability of that transition.

3. $\sum_j p(i \to j \mid d, a, \bar{x}) \times \sum_k \sum_{d',a'} \int \left\{ \int_t^{t_{mission}} \pi(\bar{u}, k, d', a', \tau \mid \bar{x}, j, d, a, t) \times R^{\alpha}(\bar{u}, k, d', a', \tau) d\tau \right\} du$ – represents the expected future rewards received by the player α. At the physical system state $\bar{x}$, the component state i and time t, when the defender takes the action d and the attacker takes the action a, there is an immediate transition in the component state from i to j, and the new system state is $(\bar{x}, j, d, a, t)$. The term $R^{\alpha}(\bar{u}, k, d', a', \tau)$ represents the reward received by the player α at some future state ($\bar{u}, k, d', a', \tau$) at time $t < \tau < t_{\text{mission}}$, physical system state $\bar{u}$ and component state k, when the defender takes action d' and the attacker takes an action a'. The term $\pi(\bar{u}, k, d', a', \tau \mid \bar{x}, j, d, a, t)$ represents the conditional probability density of arriving at the system state ($\bar{u}, k, d', a', \tau$) given that the initial state is ($\bar{x}, j, d, a, t$). The probability that the players take the pair of actions (d', a') at $(\bar{u}, k, \tau)$ is encoded in this conditional probability density.

[00200] **Example 3**

[00201] An example implementation of the present disclosure was designed and tested. The example implementation includes a simulator 1800. The simulator 1800 can be used to evaluate the protection offered by various network architectures and quantify cyber security risks. For example, the simulator 1800 can be used to meet the demands of cybersecurity research with increasing digitalization of nuclear power plants. FIG. 18 illustrates the example implementation of the simulator 1800. The simulator 1800 includes a DPRA engine 1802, a physical simulator 1804, and simulation management and scheduling module 1806, a graphical network simulator 1808, an attack library 1820, a game library 1822, and a procedure following operator library 1824.

[00202] As described with reference to examples 1 and 2, for example, the DPRA engine 1802 can be based on Dynamic Probabilistic Risk Assessment (DPRA), a set of probabilistic risk assessment (PRA) techniques in which a deterministic physics-based model is used to study the evolution of a system in the context of stochastic events. The DPRA engine 1802 monitors the state of the system (e.g., a system modeled by a physical simulator 1804), and implements the sampled attacker and defender actions as well as randomly generated component failure scenarios.

[00203] The physical simulator 1804 can model the state of a physical system. As described herein, the physical simulator 1804 can optionally model any kind of industrial facility. In the example shown in FIG. 18, the physical simulator 1804 is a generic pressurized water reactor simulator sold under the tradename GSE. In the example shown in FIG. 18, the physical simulator implements a physics model of a pressurized water reactor.

[00204] The graphical setwork simulator 1808 (GNS3) includes a simulated network model. The network model is co-simulated with the physics model of the physical simulator 1804 with the graphical network simulator 1808. Fig. 19 illustrates the co-simulation of the network and physics models implemented by the graphical network simulator 1808 and the physical simulator 1804, respectively. The memory read/write process for data transmission in the physics model of the physical simulator 1804 is replaced with data transmission across a simulated network implemented on the graphical network simulator 1808 simulator with components such as network switches. As a non-limiting example, a platform sold under the tradename Emulab can be used to implement the network model as an alternative or addition to the graphical network simulator 1808.

[00205] An example implementation of a node-management and scheduling module 1830 that can be used with the simulation management and scheduler module 1806 is shown in FIG. 20. The node management and scheduler module 1830 can include the communication and digital component node in the example GNS3 environment. The node management and schedule module 1830 module schedules communications (e.g., dictates the frequency with which information is received and sent, and identifies the nodes from which the input data should be received and to which the output data is sent) through the node communication function 1832 using the input from the simulation management and scheduler module 1806. Additionally, the node management and scheduler module 1830 provides the node state information to the node component function 1834. It is implicit that each node can represent a digital component in the system such as controller, sensor etc., as described with reference to the example implementations herein, and the function of such a component is dependent on its state.

[00206] Again with reference to FIG. 18, the simulator 1800 can further include an attack library 1820. The attack library can include a set of cyber-attack actions and scenarios that can be implemented using the graphical network simulator and/or the physical simulator 1804. As non-limiting examples, the scenarios of the attack library 1820 can include abnormal event detection and/or classification scenarios.

[00207] The simulator 1800 can further include a game library 1822. The game library can apply a game-theory model to model behaviors of rational attackers/defenders in the simulator 1800. The game library 1822 can optionally generate attacker and/or defender strategies. Optionally attacker and defender strategies can be implemented using game theory

techniques including mixed-strategy Nash equilibriums and/or Stackelberg equilibriums, for example. When the attacker and/or defender actions are implemented by the simulator 1800 (e.g., by modeling the attacker/defender actions on the physical model of the physical simulator1804), trajectories can be generated for the mixed strategies of the attacker and defender. Pairs of defender-attacker actions can be sampled from the mixed strategies by the DPRA engine 1802, and implemented using the physical simulator and/or the graphical network simulator 1808. An example plot 2100 of game-theory trajectories 2104 for attacker and defender action pairs 2102 in a lookahead game is illustrated in FIG. 21.

[00208] Again with reference to FIG. 18, the simulator 1800 can further include a procedure following operator library 1824. The procedure following operator library 1824 can be configured to emulate a procedure that is performed during an operation of the physical simulator 1804. As a non-limiting example, the procedure following operator library 1824 can include procedures performed by an operator of a physical system modeled by the physical simulator 1804 under emergency conditions (e.g., the operation of a nuclear plant under a cyber attack). The procedure following operator library can optionally run in parallel to the graphical network simulator 1808 and/or the physical simulator 1804 to simulate procedures that are performed based on the states of the physical system simulated by the physical simulator 1804 and/or based on the state of a network modeled by the graphical network simulator.

[00209] Still with reference to FIG. 18, the simulation management and scheduling module 1806 is configured to manage/control the operation of the other modules (e.g., any or all of the DPRA engine 1802, physical simulator 1804, graphical network simulator

1808, attack library 1820, game library 1822, and/or procedure following operator library 1824).

[00210] FIG. 22 illustrates a timing diagram illustrating the parallel operation of the graphical network simulator 1808, physical simulator 1804, procedure following operator library 1824, and game library 1824. The simulation management and scheduling module 1806 can initialize the communication and digital component nodes of the graphical network simulator 1808 and initialize states of any or all of the DPRA engine 1802, physical simulator 1804, graphical network simulator 1808, attack library 1820, game library 1822, and/or procedure following operator library 1824. The simulation is considered to start at time T_0, and the graphical network simulator 1808, physical simulator 1804 and procedure following operator library 1824 run in parallel as shown in FIG 22. At time steps T step T_i, $i > 0$, the game library 1822 can use a lookahead game simulation to implement all possible pairs of attacker and defender actions on the low order model and runs a predictive simulation for predefined number of time steps into the future as presented in FIG. 21. Again, as shown in FIG. 21, the mixed strategies of attacker and defender can be computed from those trajectories 2104. The DPRA engine 1802 samples one or more attacker and defender action pairs 2102 from the mixed strategies, and implements those actions on the physical simulator 1804 and/or graphical network simulator 1808 to evaluate the effects of the attacker and defender action pairs 2102 on the physical system and network modeled by the physical simulator 1804 and/or graphical network simulator 1808.

[00211] Although the subject matter has been described in language specific to structural features and/or methodological acts, it is to be understood that the subject matter

defined in the appended claims is not necessarily limited to the specific features or acts described above. Rather, the specific features and acts described above are disclosed as example forms of implementing the claims.

[00212] The following patents, applications, and publications, as listed below and throughout this document, describes various application and systems that could be used in combination the exemplary system and are hereby incorporated by reference in their entirety herein

[00213] [1] A. Mosleh, "PRA: A PERSPECTIVE ON STRENGTHS, CURRENT LIMITATIONS, AND POSSIBLE IMPROVEMENTS," Nucl. Eng. Technol., vol. 46, no. 1, pp. 1-10, Feb. 2014, doi: 10.5516/NET.03.2014.700.

[00214] [2] R. J. Breeding, T. J. Leahy, and J. Young, "Probabilistic risk assessment course documentation. Volume 1: PRA fundamentals," Energy, Inc., Seattle, WA (USA), NUREG/CR-4350/1; SAND85-1495/1, Aug. 1985. doi: 10.2172/6277413.

[00215] [3] International Atomic Energy Agency, "Defining initiating events for purposes of probabilistic safety assessment," IAEA-TECDOC-719, International Atomic Energy Agency. 1993.

[00216] [4] N. J. McCormick, Reliability and risk analysis: methods and nuclear power applications. Academic Press New York, 1981.

[00217] [5] J. W. Park and S. J. Lee, "Probabilistic safety assessment-based importance analysis of cyberattacks on nuclear power plants," Nucl. Eng. Technol., vol. 51, no. 1, pp. 138-145, Feb. 2019, doi: 10.1016/j.net.2018.09.009.

[00218] [6] Y. Zhao, L. Huang, C. Smidts, and Q. Zhu, "A game theoretic approach for responding to cyberattacks on nuclear power plants," Nucl. Sci. Eng., 2021.

[00219] [7] Y. Zhao, L. Huang, C. Smidts, and Q. Zhu, "Finite-horizon semi-Markov game for timesensitive attack response and probabilistic risk assessment in nuclear power plants," Reliab. Eng. Syst. Saf., vol. 201, p. 106878, Sep. 2020, doi: 10.1016/j.ress.2020.106878.

[00220] [8] J. Devooght and C. Smidts, "Probabilistic Reactor Dynamics-I: The Theory of Continuous Event Trees," Nucl. Sci. Eng., vol. 111, no. 3, pp. 229-240, Jul. 1992, doi: 10.13182/NSE92A23937.

[00221] [9] T. Aldemir, "A survey of dynamic methodologies for probabilistic safety assessment of nuclear power plants," Ann. Nucl. Energy, vol. 52, pp. 113-124, Feb. 2013, doi: 10.1016/j.anucene.2012.08.001.

[00222] [10] C. Smidts and J. Devooght, "Probabilistic Reactor Dynamics-II: A Monte Carlo Study of a Fast Reactor Transient," Nucl. Sci. Eng., vol. 111, no. 3, pp. 241-256, Jul. 1992, doi: 10.13182/NSE92-A23938.

[00223] [11] T. Aldemir et al., "NUREG/CR-6942: Dynamic Reliability Modeling of Digital Instrumentation and Control Systems for Nuclear Reactor Probabilistic Risk Assessments," 2007.

[00224] [12] J. Devooght and C. Smidts, "Probabilistic Reactor Dynamics - III. A Framework for TimeDependent Interaction between Operator and Reactor during a Transient Involving Human Error," Nucl. Sci. Eng., vol. 112, no. 2, pp. 101-113, Oct. 1992, doi: 10.13182/NSE92-A28407.

[00225] [13] C. Smidts, "Probabilistic Reactor Dynamics -IV. An Example of Man/Machine Interaction," Nucl. Sci. Eng., vol. 112, no. 2, pp. 114-126, Oct. 1992, doi: 10.13182/NSE92-A28408.

[00226] [14] K.-S. Hsueh and A. Mosleh, "The development and application of the accident dynamic simulator for dynamic probabilistic risk assessment of nuclear power plants," Reliab. Eng. Syst. Saf., vol. 52, no. 3, pp. 297-314, Jun. 1996, doi: 10.1016/0951-8320(95)00140-9.

[00227] [15] Y. Zhao et al., "Dynamic Probabilistic Risk Assessment for Cyber Security Risk Analysis of the Electric Grid," in Proceedings of the 30th European Safety and Reliability Conference and 15th Probabilistic Safety Assessment and Management Conference, 2020, pp. 2020-2027. doi: 10.3850/978-981-14-8593-0_5058-cd.

[00228] [16] P. K. Vaddi et al., "Dynamic bayesian networks based abnormal event classifier for nuclear power plants in case of cyber security threats," Prog. Nucl. Energy, vol. 128, p. 103479, Oct. 2020, doi: 10.1016/j.pnucene.2020.103479.

WHAT IS CLAIMED:

1. A method for performing dynamic probabilistic risk assessment, the method comprising:

providing a system model of a physical system;

providing an action model configured to generate a set of operator and attacker actions based on a game theory analysis; and

providing a dynamic probabilistic risk assessment (DPRA) engine, wherein the DPRA engine is configured to:

receive the set of operator and attacker actions from the action model;

sample the set of operator and attacker actions; and

provide the sampled set of operator and attacker actions to the system model for implementation by the system model.

2. The method of claim 1, wherein the set of operator and attacker actions is randomly sampled.

3. The method of claim 1 or claim 2, wherein the DPRA engine is further configured to receive a state of the physical system from the system model, and provide the state of the physical system to the action model.

4. The method of any one of claims 1-3, wherein the system model comprises a network model.

5. The method of claim 4, wherein the DPRA engine is further configured to receive a state of the network model from the system model, and provide the state of the network model to the action model.

6. The method of any one of claims 1-5, wherein the action model comprises a plurality of sub-models.

7. The method of claim 6, wherein the plurality of sub-models comprise a procedure model, an undetected attacker model, and a game model.

8. The method of any one of claims 1-7, wherein the DPRA engine is configured to implement an attacker action on the system model.

9. The method of any one of claims 1-8, wherein the DPRA engine is configured to implement an operator action on the system model.

10. The method of any one of claims 1-9, wherein the action model is configured to generate a plurality of actions for an operator to implement on the physical system.

11. The method of any one of claims 1-10, wherein the action model is configured to generate a plurality of actions for an attacker to implement on the physical system.

12. The method of any one of claims 1-11, wherein the action model is configured to generate both a plurality of actions for an attacker and a plurality of actions for an operator to implement on the physical system.

13. The method of any one of claims 1-12, further comprising determining a risk metric based on the actions in the system, wherein the risk metric is based on the action model and the system model.

14. The method of claim 13, further comprising modeling a plurality of actions for an attacker and a plurality of actions for a defender based on a game theory analysis of the attacker and defender's actions.

15. A method for performing dynamic probabilistic risk assessment, the method comprising:

modelling actions in a system model using an action model and

sampling the actions to be implemented on the system model by a DPRA engine from the actions modeled by the action model; and

implementing the sampled actions on the system model.

16. The method of claim 15, further comprising determining a risk metric based on the actions in the system.

17. A system comprising:

a network device;

a controller operably coupled to the network device, the controller comprising a processor and a memory, the memory having computer-executable instructions stored thereon that, when executed by the processor, cause the controller to:

model action in a system model using an action model;

sample the actions to be implemented on the system model by a DPRA engine from the actions modeled by the action model; and

implement the sampled actions on the system model.

18. The system of claim 17, wherein the network device is coupled to a network comprising a functional node.

19. The system of claim 18, wherein the functional node comprises at least one of a controller or sensor.

20. The system of any one of claims 17-19, wherein the controller is further configured to control the network device based on the sampled actions implemented on the system model.

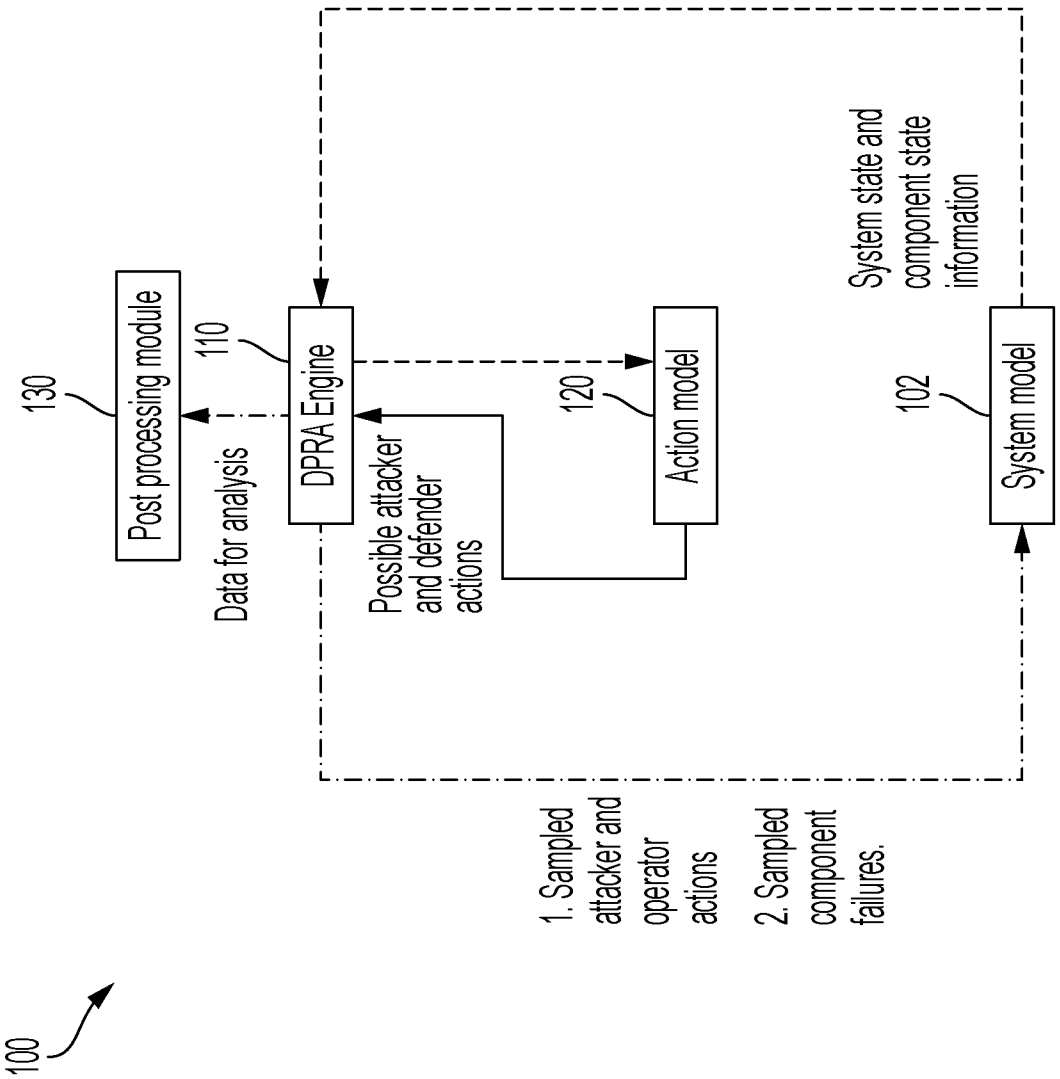

100
130
Post processing module
110
DPRA Engine
120
Action model
102
System model
Data for analysis
Possible attacker and defender actions
System state and component state information
1. Sampled attacker and operator actions
2. Sampled component failures.

FIG. 1A

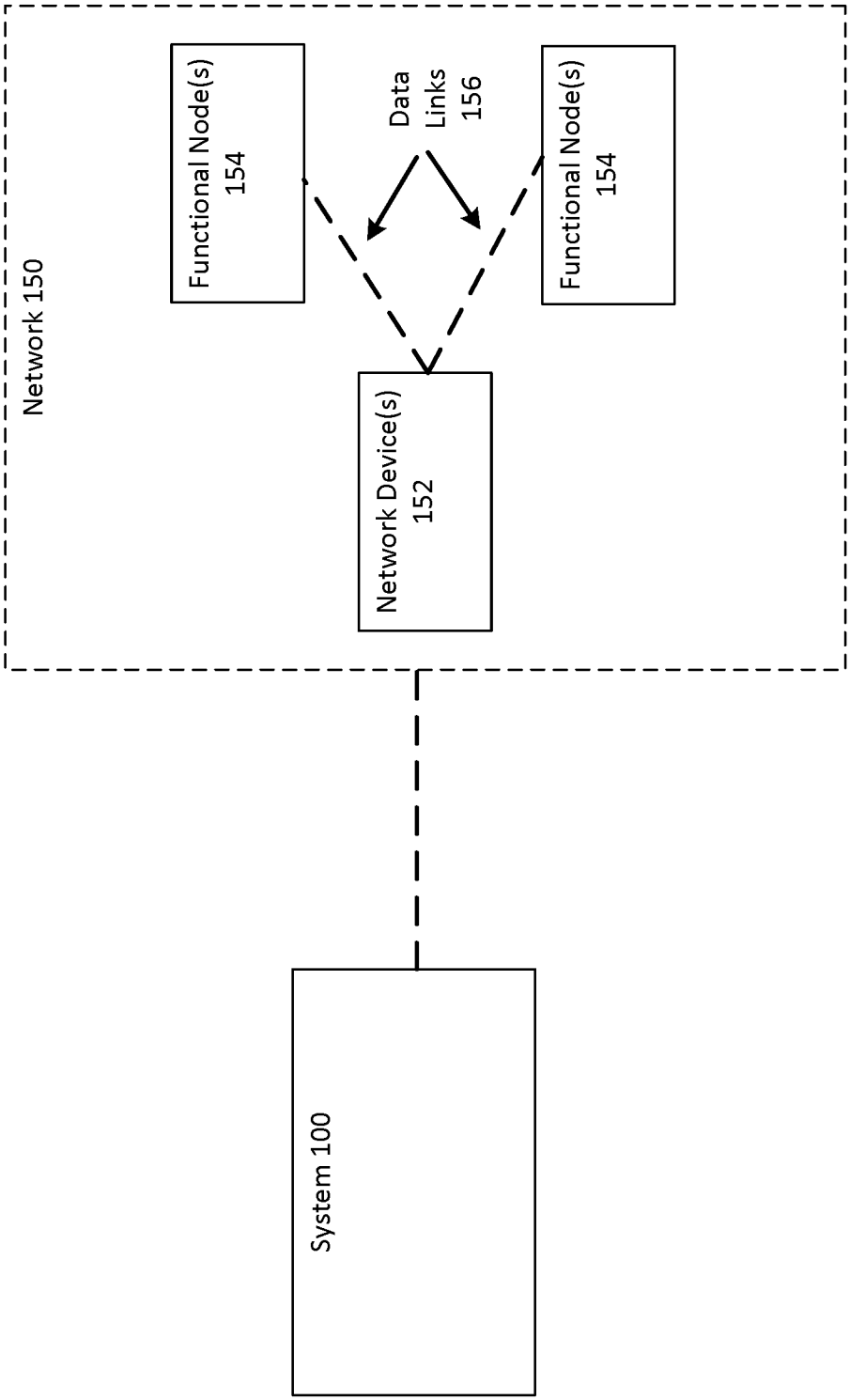

Network 150
Functional Node(s) 154
Data Links 156
Functional Node(s) 154
Network Device(s) 152
System 100

FIG. 1B

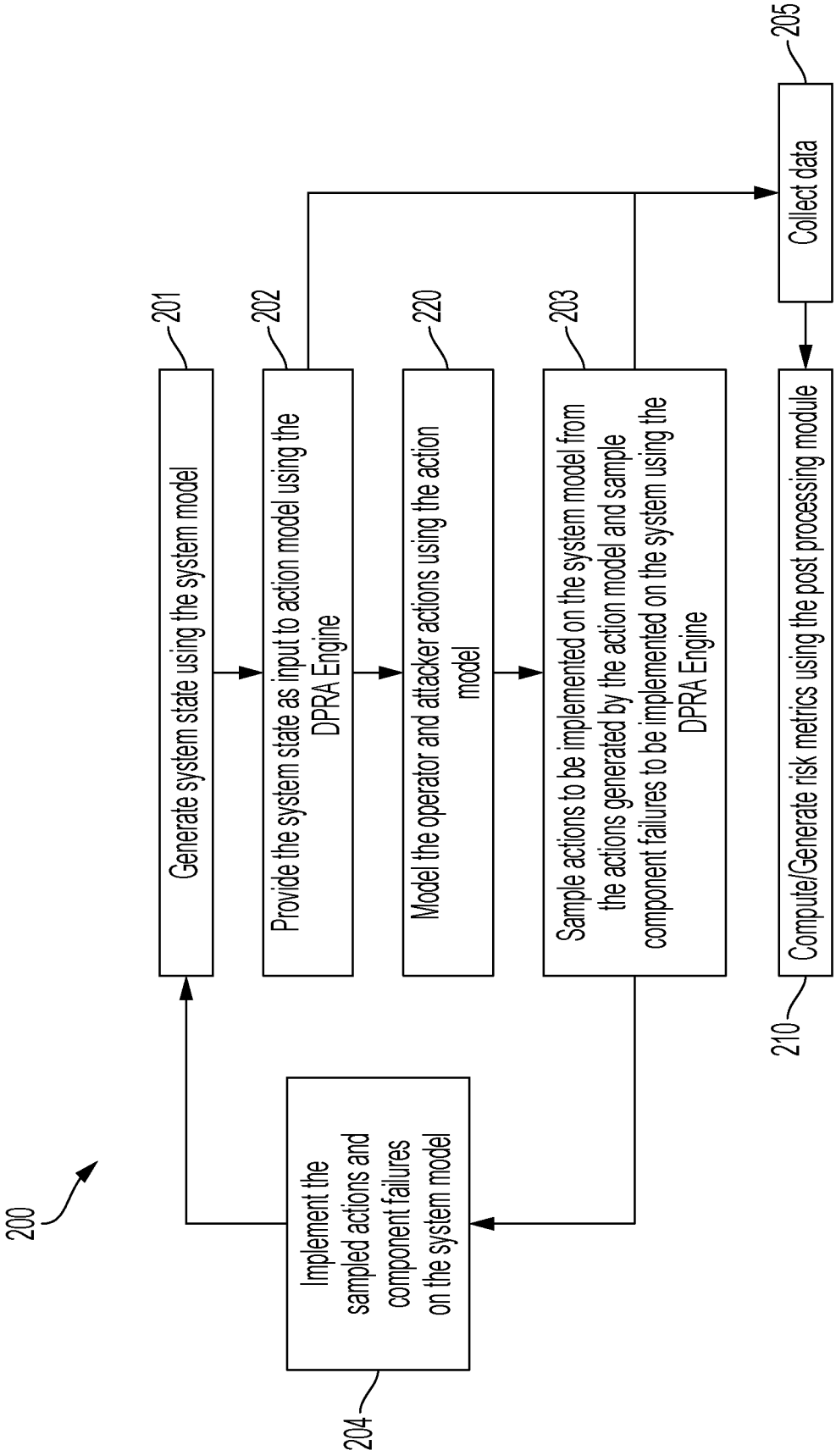

200
201
Generate system state using the system model
202
Provide the system state as input to action model using the DPRA Engine
220
Model the operator and attacker actions using the action model
203
Sample actions to be implemented on the system model from the actions generated by the action model and sample component failures to be implemented on the system using the DPRA Engine
204
Implement the sampled actions and component failures on the system model
205
Collect data
210
Compute/Generate risk metrics using the post processing module

FIG. 2

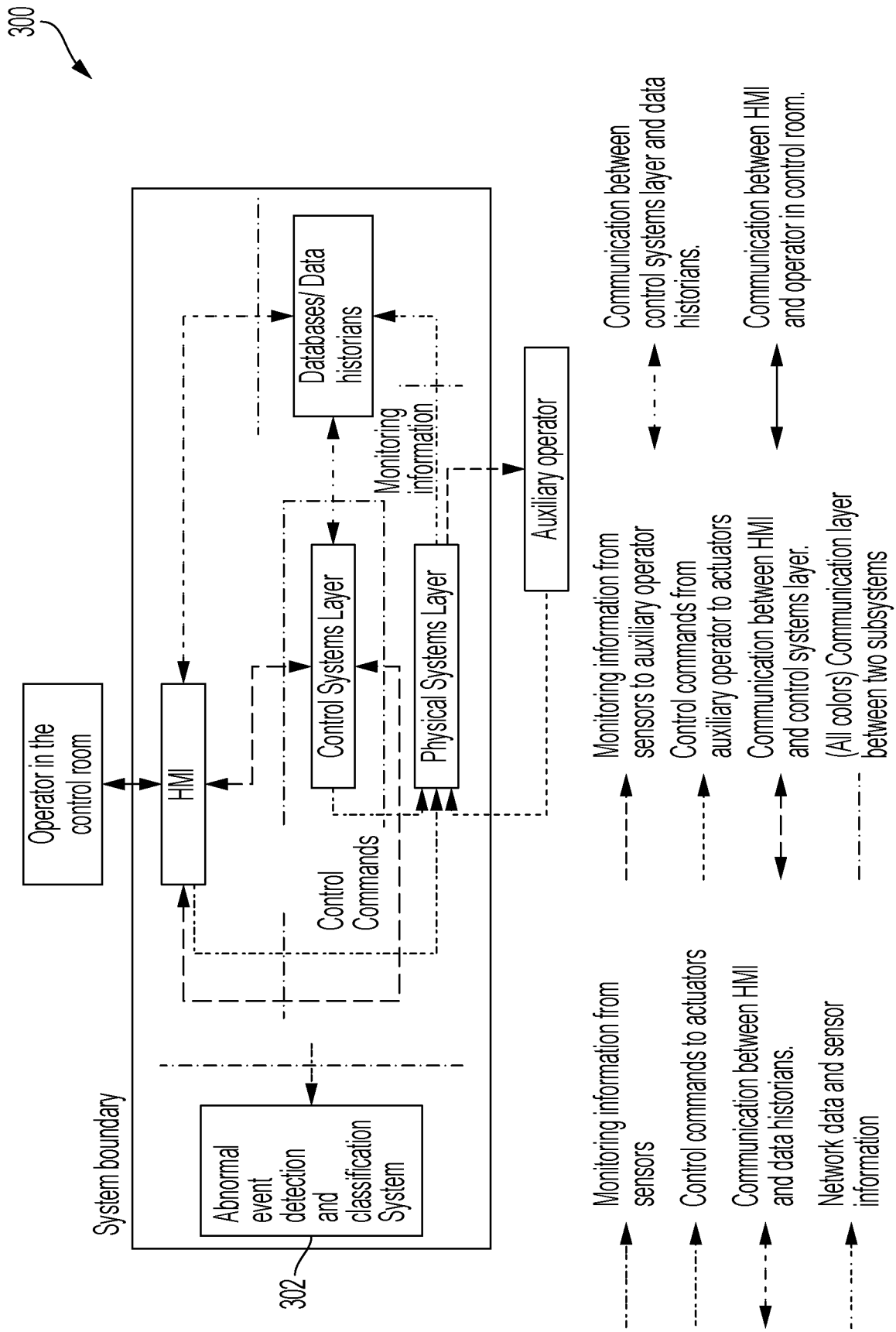

300
Operator in the control room
HMI
Control Systems Layer
Physical Systems Layer
Databases/ Data historians
Auxiliary operator
Monitoring information
Control Commands
System boundary
Abnormal event detection and classification System
302
Monitoring information from sensors
Control commands to actuators
Communication between HMI and data historians.
Network data and sensor information
Monitoring information from sensors to auxiliary operator
Control commands from auxiliary operator to actuators
Communication between HMI and control systems layer.
(All colors) Communication layer between two subsystems
Communication between control systems layer and data historians.
Communication between HMI and operator in control room.

FIG. 3

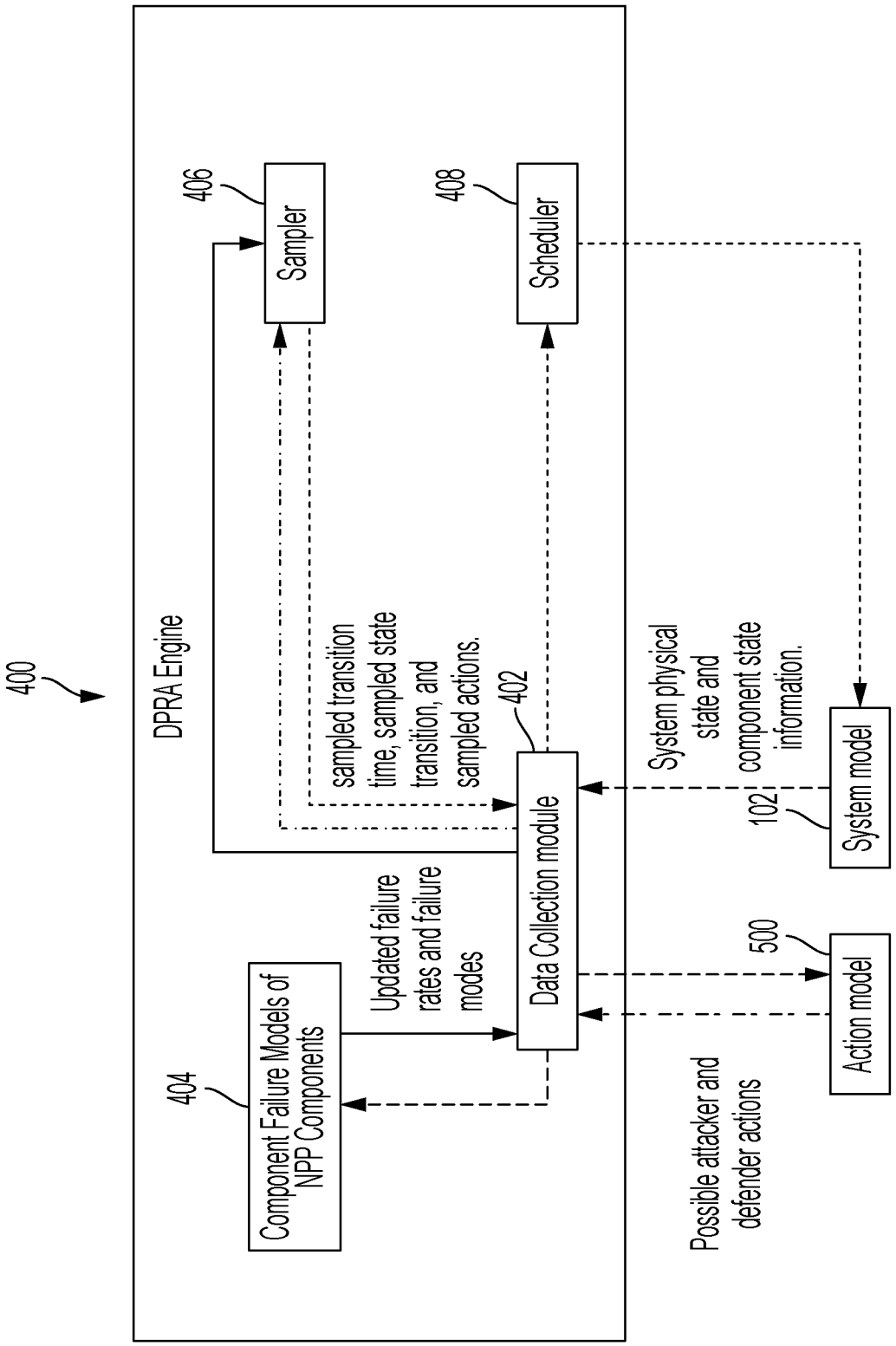

400
DPRA Engine
406
Sampler
408
Scheduler
sampled transition time, sampled state transition, and sampled actions.
402
Data Collection module
System physical state and component state information.
102
System model
Updated failure rates and failure modes
404
Component Failure Models of NPP Components
500
Action model
Possible attacker and defender actions

FIG. 4

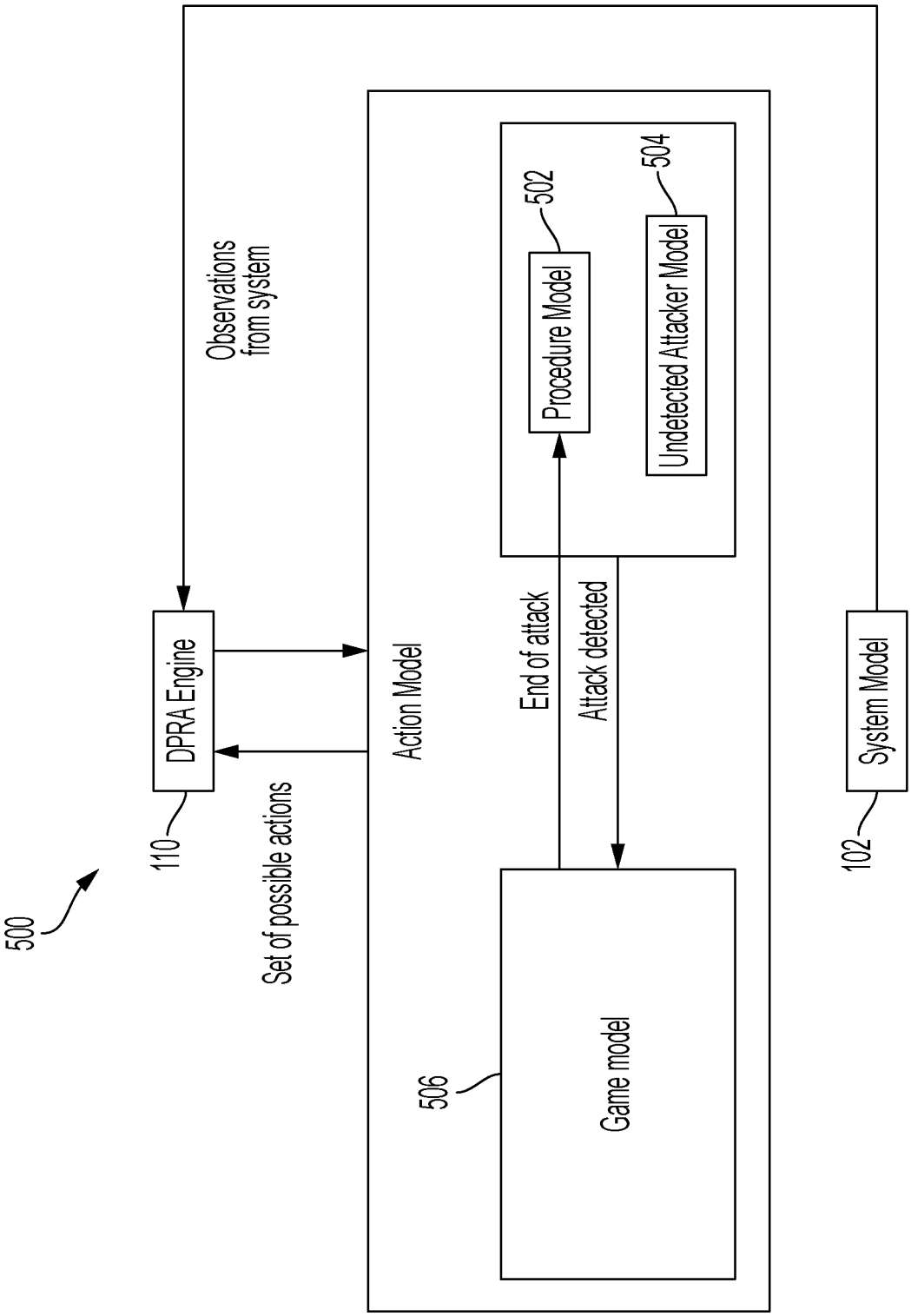

500
110
DPRA Engine
Set of possible actions
Observations from system
Action Model
Procedure Model
502
Undetected Attacker Model
504
End of attack
Attack detected
Game model
506
System Model
102

FIG. 5

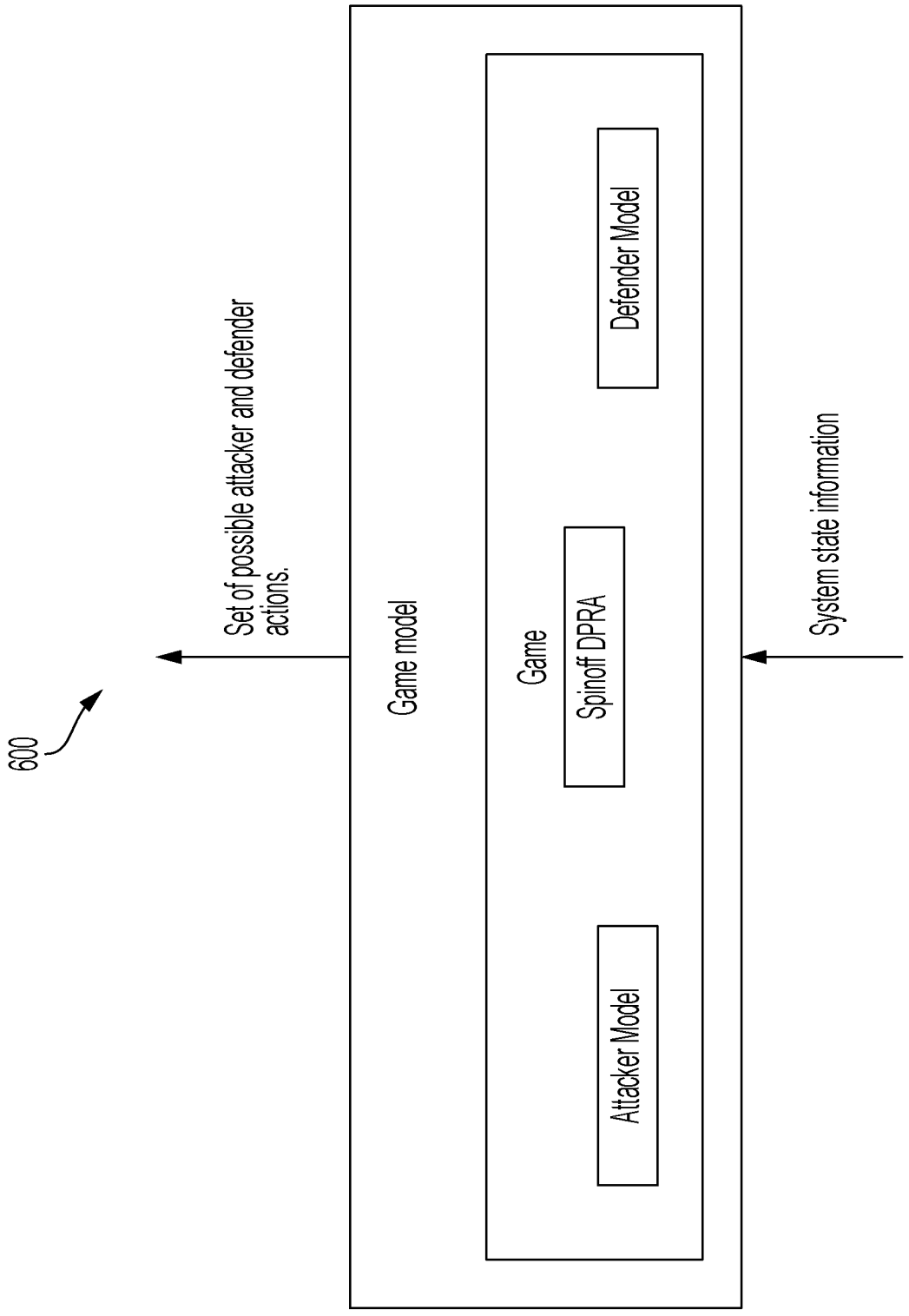

600
Set of possible attacker and defender actions.
Game model
Game
Spinoff DPRA
Attacker Model
Defender Model
System state information

FIG. 6

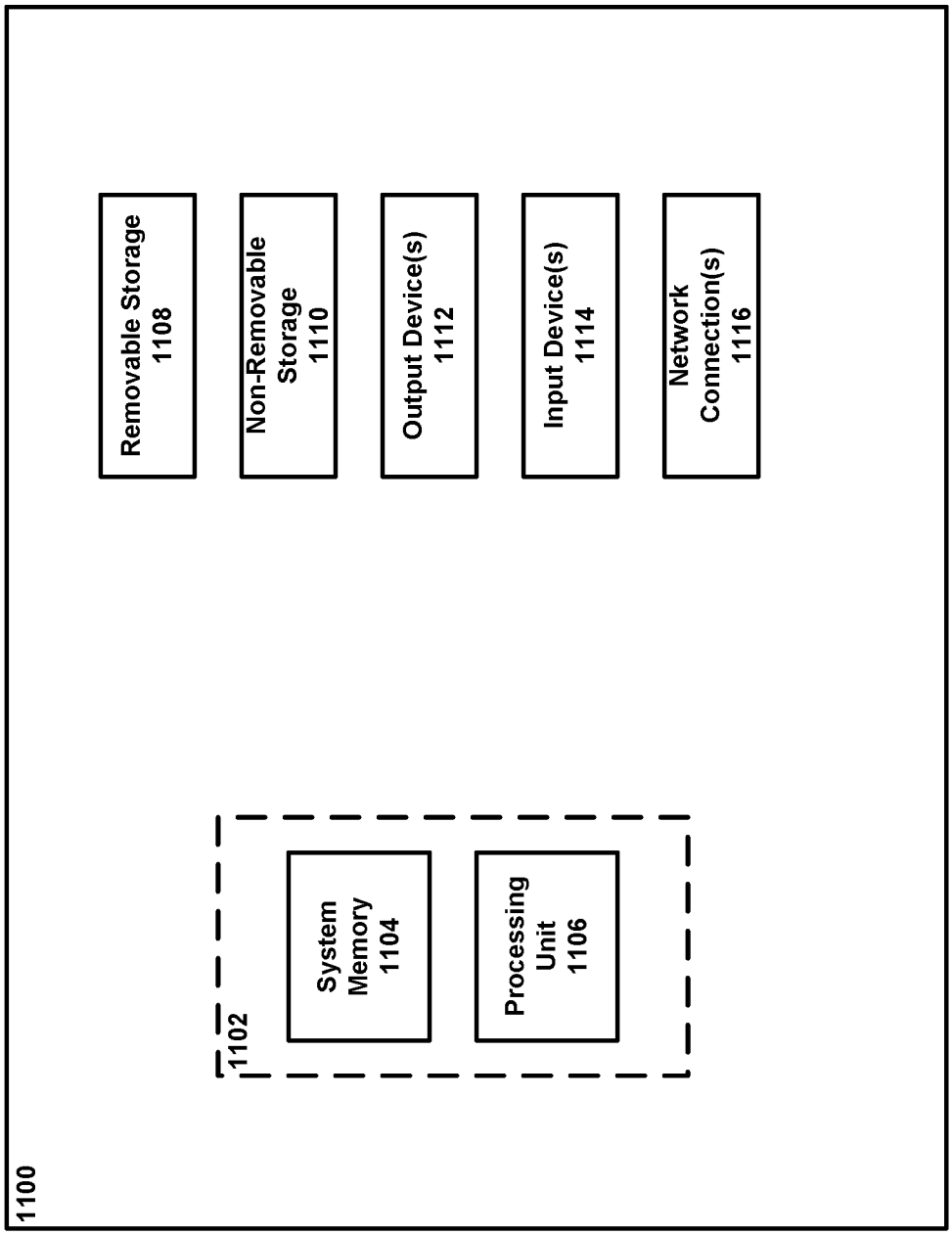

1100
1102
System Memory 1104
Processing Unit 1106
Removable Storage 1108
Non-Removable Storage 1110
Output Device(s) 1112
Input Device(s) 1114
Network Connection(s) 1116

FIG. 7

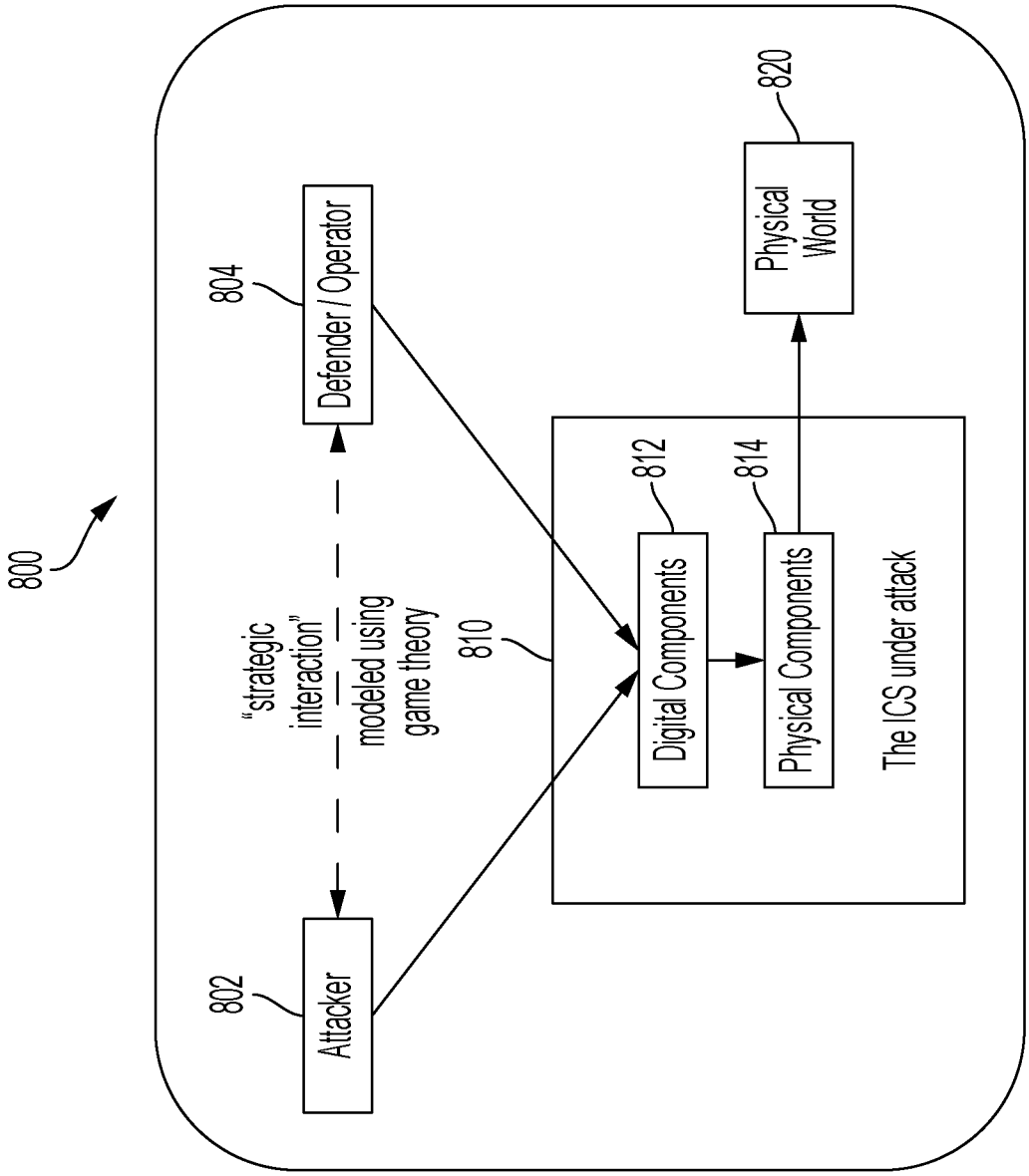

800
804
Defender / Operator
820
Physical World
"strategic interaction" modeled using game theory
810
812
814
Digital Components
Physical Components
The ICS under attack
802
Attacker

FIG. 8

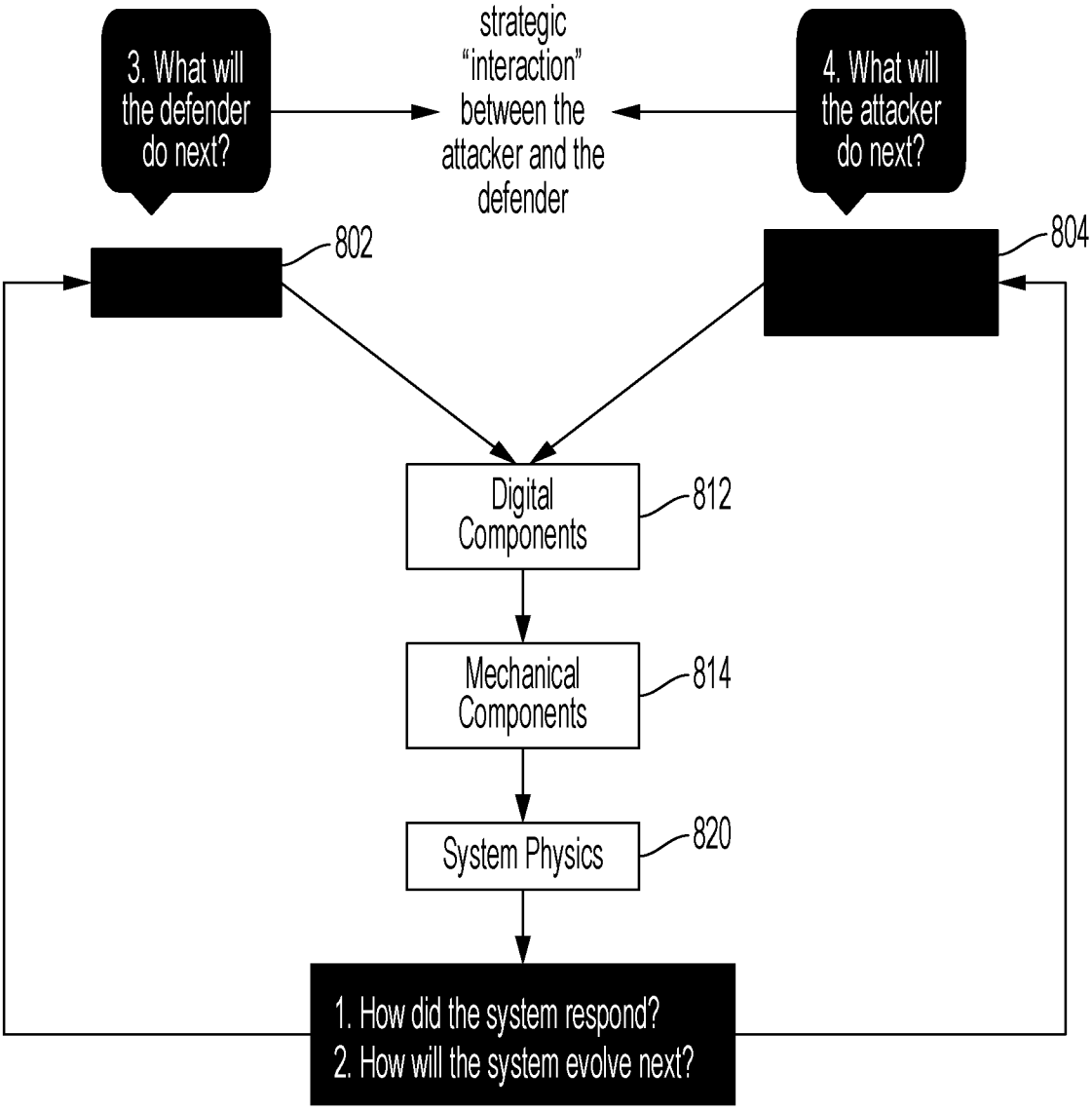

3. What will the defender do next?
strategic "interaction" between the attacker and the defender
4. What will the attacker do next?
802
804
Digital Components
812
Mechanical Components
814
System Physics
820
1. How did the system respond?
2. How will the system evolve next?

FIG. 9

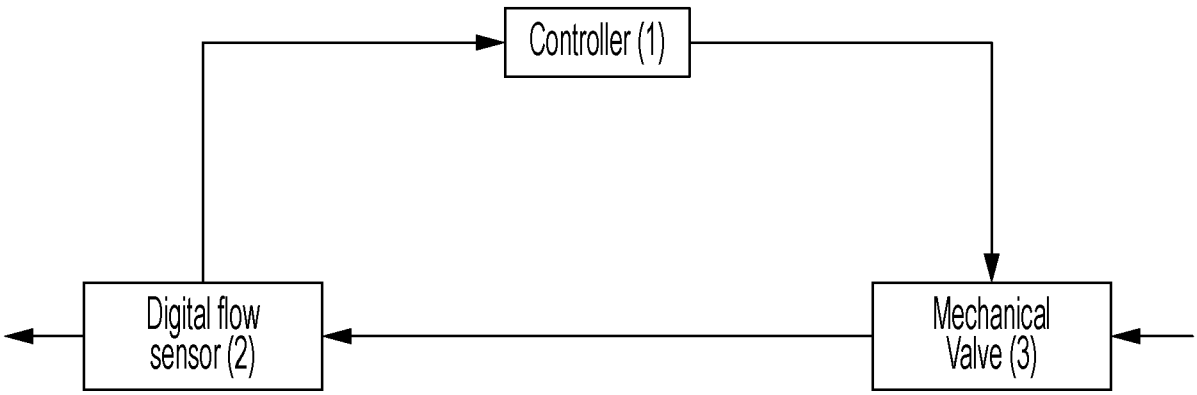


FIG. 10

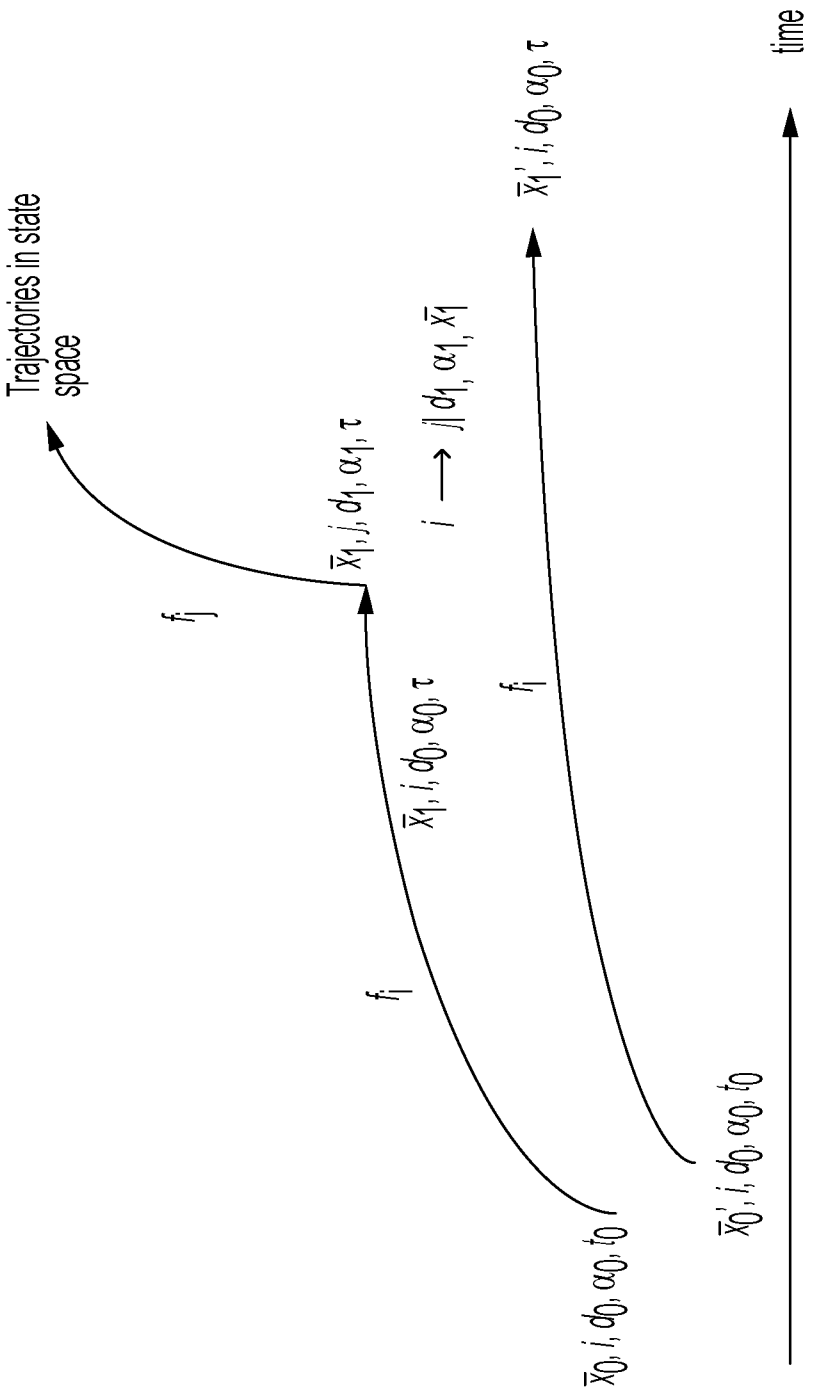

Trajectories in state space
time
f_j
$\bar{x}_1, j, d_1, \alpha_1, \tau$
$i \longrightarrow j \mid d_1, \alpha_1, \bar{x}_1$
$\bar{x}_1, i, d_0, \alpha_0, \tau$
f_i
$\bar{x}_1', i, d_0, \alpha_0, \tau$
f_i
$\bar{x}_0, i, d_0, \alpha_0, t_0$
$\bar{x}_0', i, d_0, \alpha_0, t_0$

FIG. 11

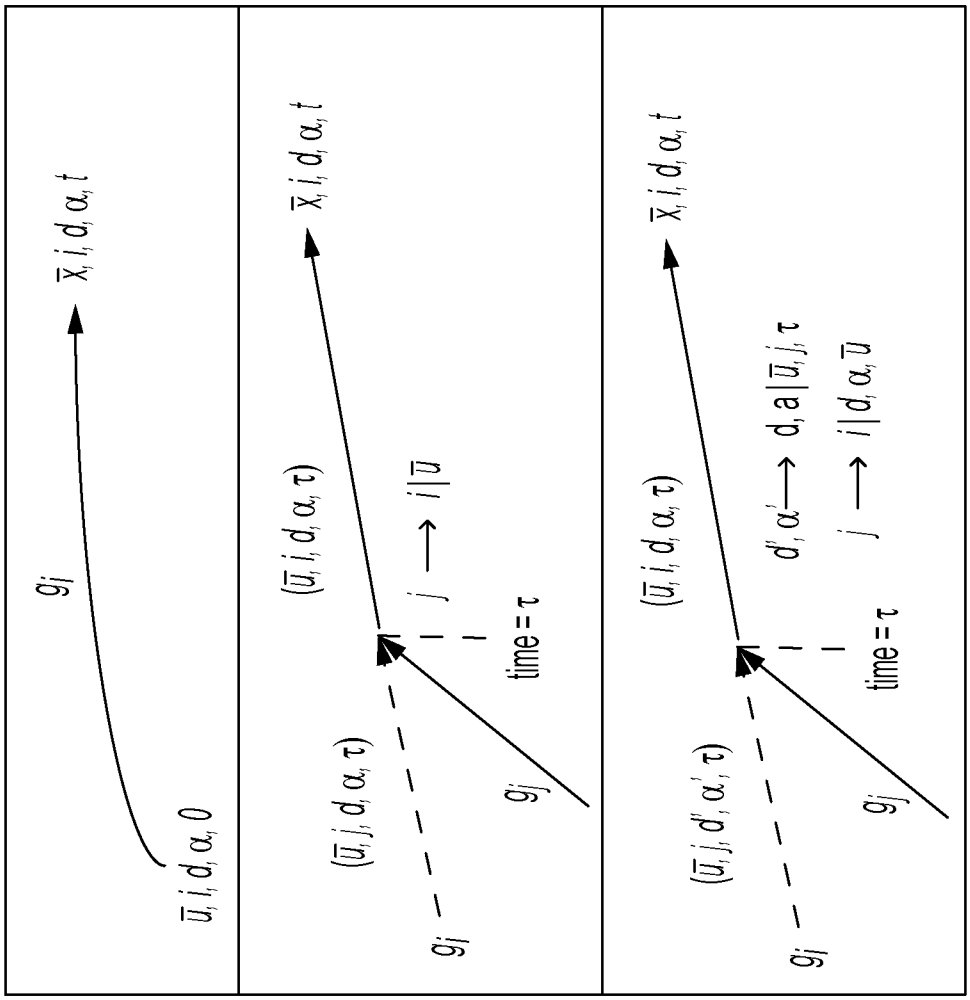

$\bar{u}, i, d, \alpha, 0$
g_j
$\bar{\chi}, i, d, \alpha, t$
$(\bar{u}, j, d, \alpha, \tau)$
g_j
$j \longrightarrow i \,|\, \bar{u}$
time = τ
g_j
$(\bar{u}, i, d, \alpha, \tau)$
$\bar{\chi}, i, d, \alpha, t$
$(\bar{u}, j, d', \alpha', \tau)$
g_j
$d', \alpha' \longrightarrow d, a \,|\, \bar{u}, j, \tau$
$j \longrightarrow i \,|\, d, \alpha, \bar{u}$
time = τ
g_j
$(\bar{u}, i, d, \alpha, \tau)$
$\bar{\chi}, i, d, \alpha, t$

FIG. 12

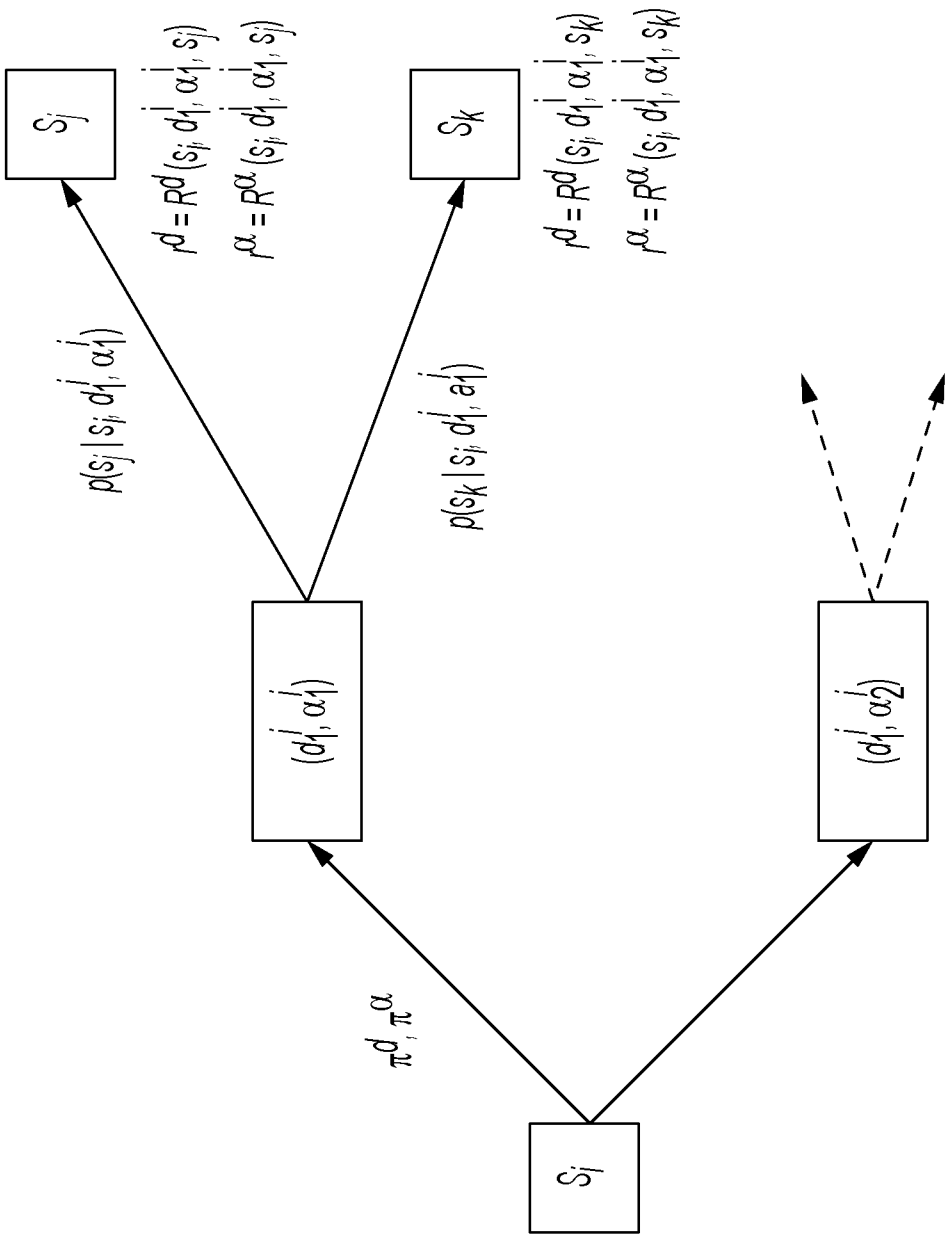

s_j
s_k
s_i
(d_1^i, α_1^j)
(d_1^i, α_2^j)
$p(s_j \mid s_i; d_1^i, \alpha_1^j)$
$p(s_k \mid s_i; d_1^i, a_1^j)$
$r^d = R^d(s_i; d_1^i, \alpha_1^j, s_j)$
$r^\alpha = R^\alpha(s_i; d_1^i, \alpha_1^j, s_j)$
$r^d = R^d(s_i; d_1^i, \alpha_1^j, s_k)$
$r^\alpha = R^\alpha(s_i; d_1^i, \alpha_1^j, s_k)$
π^d, π^α

FIG. 13

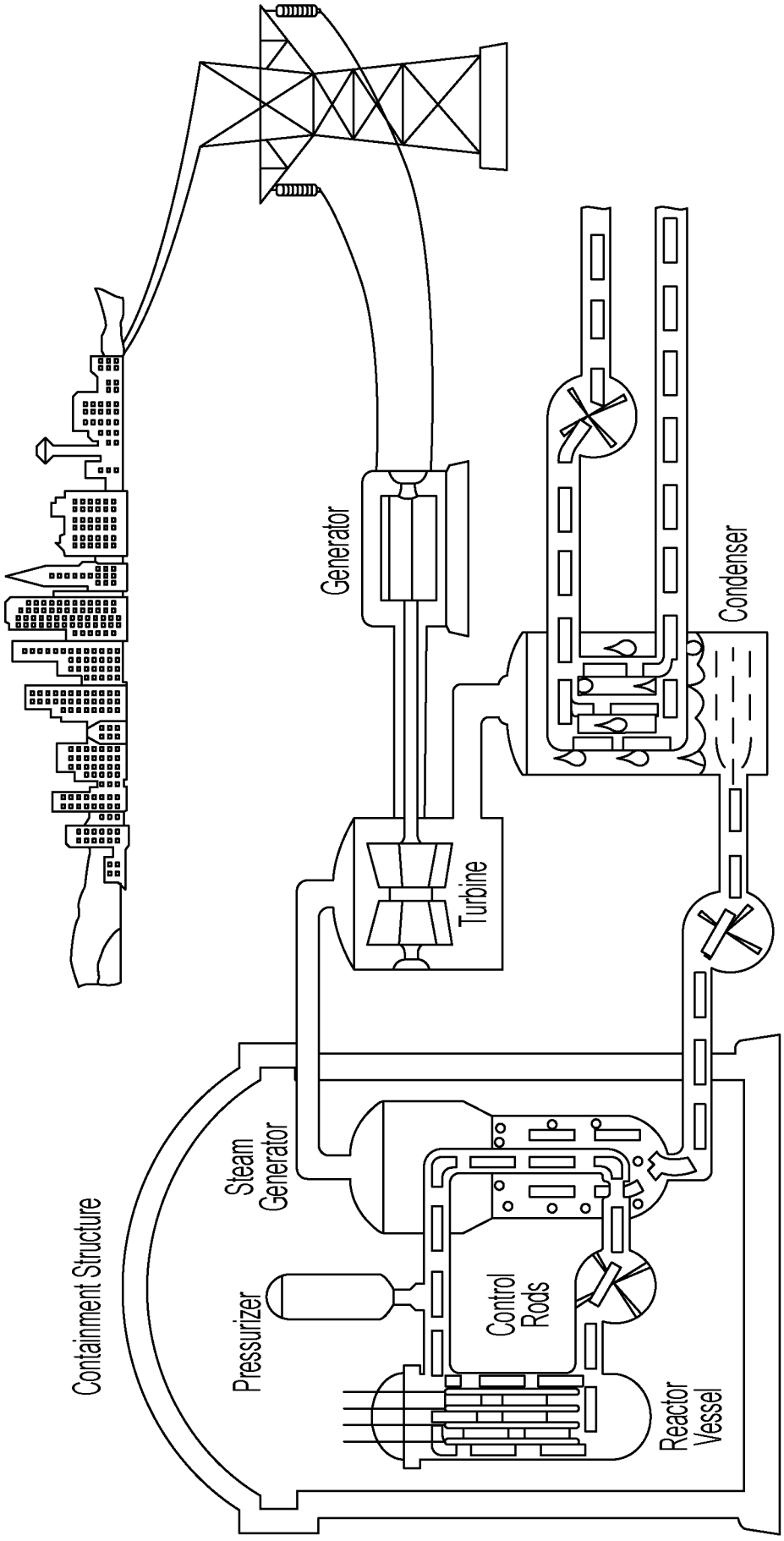

Containment Structure
Steam Generator
Pressurizer
Control Rods
Reactor Vessel
Turbine
Generator
Condenser

FIG. 14

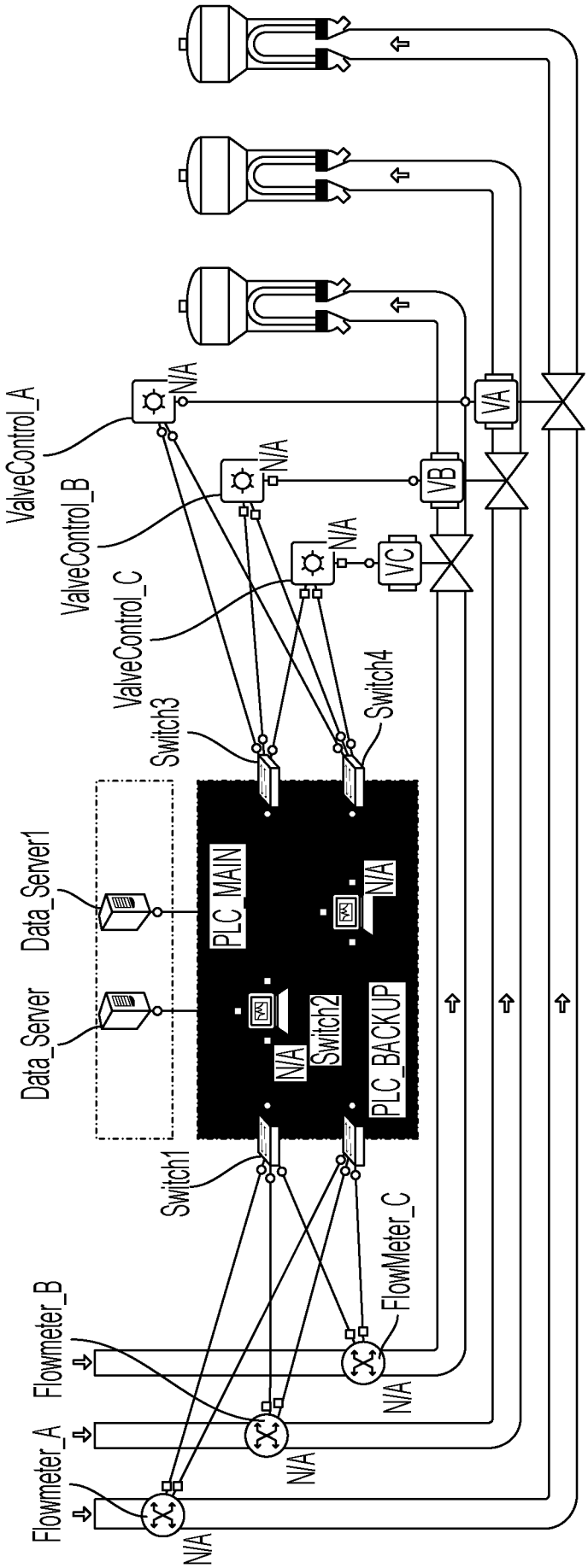

ValveControl_A
ValveControl_B
ValveControl_C
N/A
N/A
N/A
VA
VB
VC
Switch3
Switch4
Data_Server1
Data_Server
PLC_MAIN
N/A
N/A
Switch2
PLC_BACKUP
Switch1
FlowMeter_C
Flowmeter_B
Flowmeter_A
N/A
N/A
N/A

FIG. 15

Components	States
Link	1- Active: Information is transmitted across the link successfully. 0- Inactive: No information is transmitted across the link.
Sensor	1- Normal: The sensor measures and transmits the correct flowrate value. 2- Compromised: The sensor transmits a flowrate value that is higher than normal, to make the controller reduce the feedwater flow into the SG.
Computer (Main, Backup)	1- Normal: The correct control value is computed. If there is no input from the sensor, default value is used for computation. 2- Compromised: The computer tries to reduce the feedwater flow to the steam generator.
Valve Controller	1- Normal 1: The control value received from the MC is implemented. If there is no input, a default value is used. 2- Normal 2: The control value received from the BC is used. If there is no input, a default value is used. 3- Compromised: The valve controller tries to reduce the feedwater flow to the steam generator.

FIG. 16

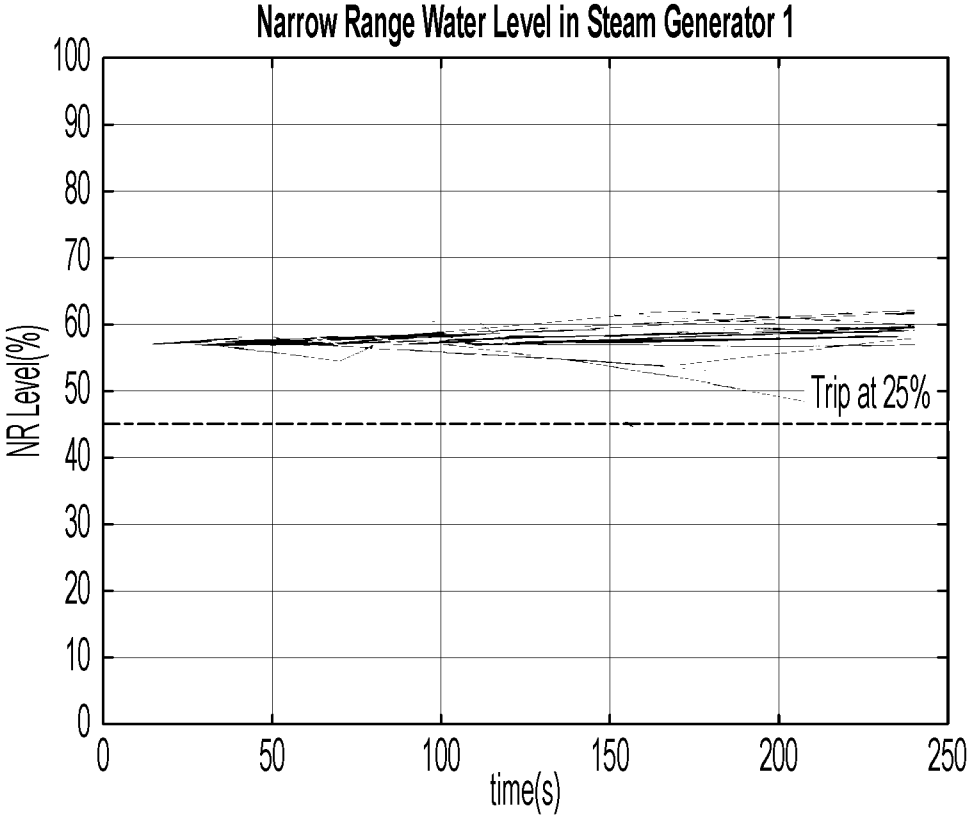


FIG. 17A

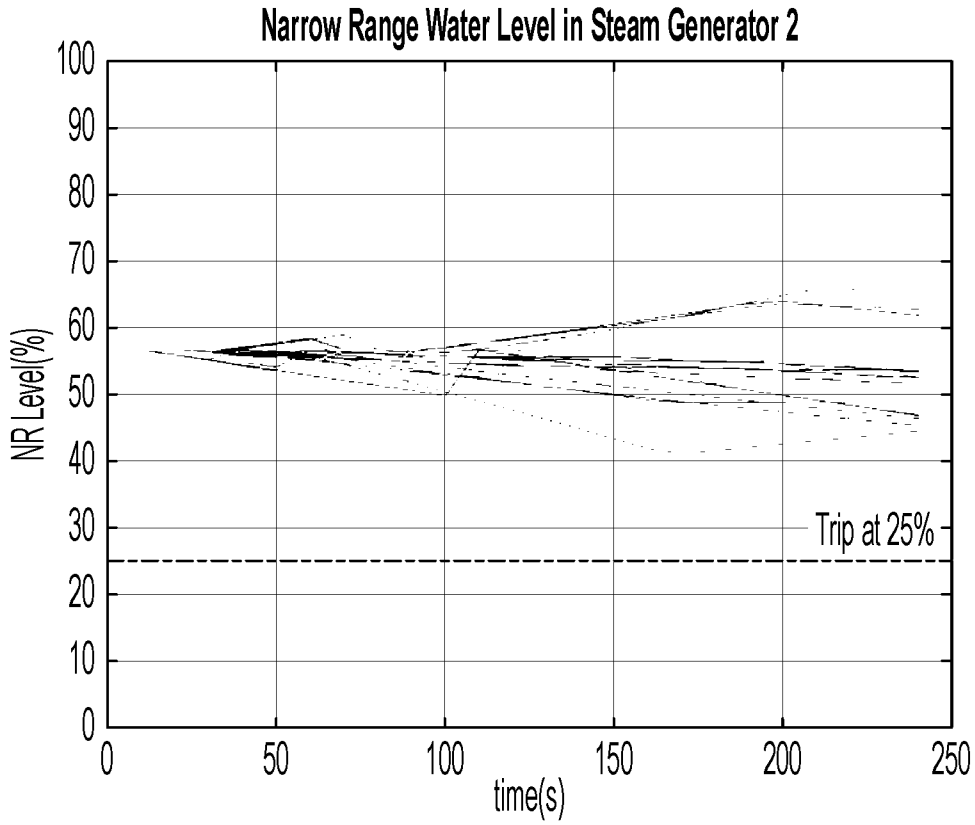


FIG. 17B

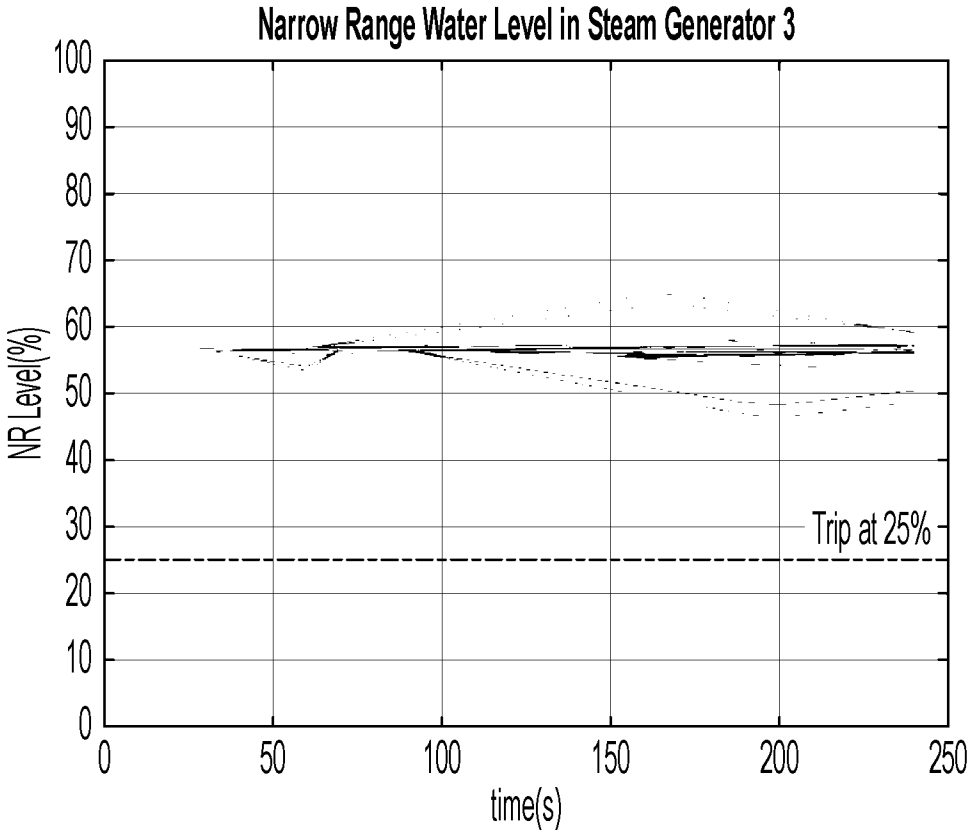

Narrow Range Water Level in Steam Generator 3
NR Level(%)
100
90
80
70
60
50
40
30
20
10
0
0
50
100
150
200
250
time(s)
Trip at 25%

FIG. 17C

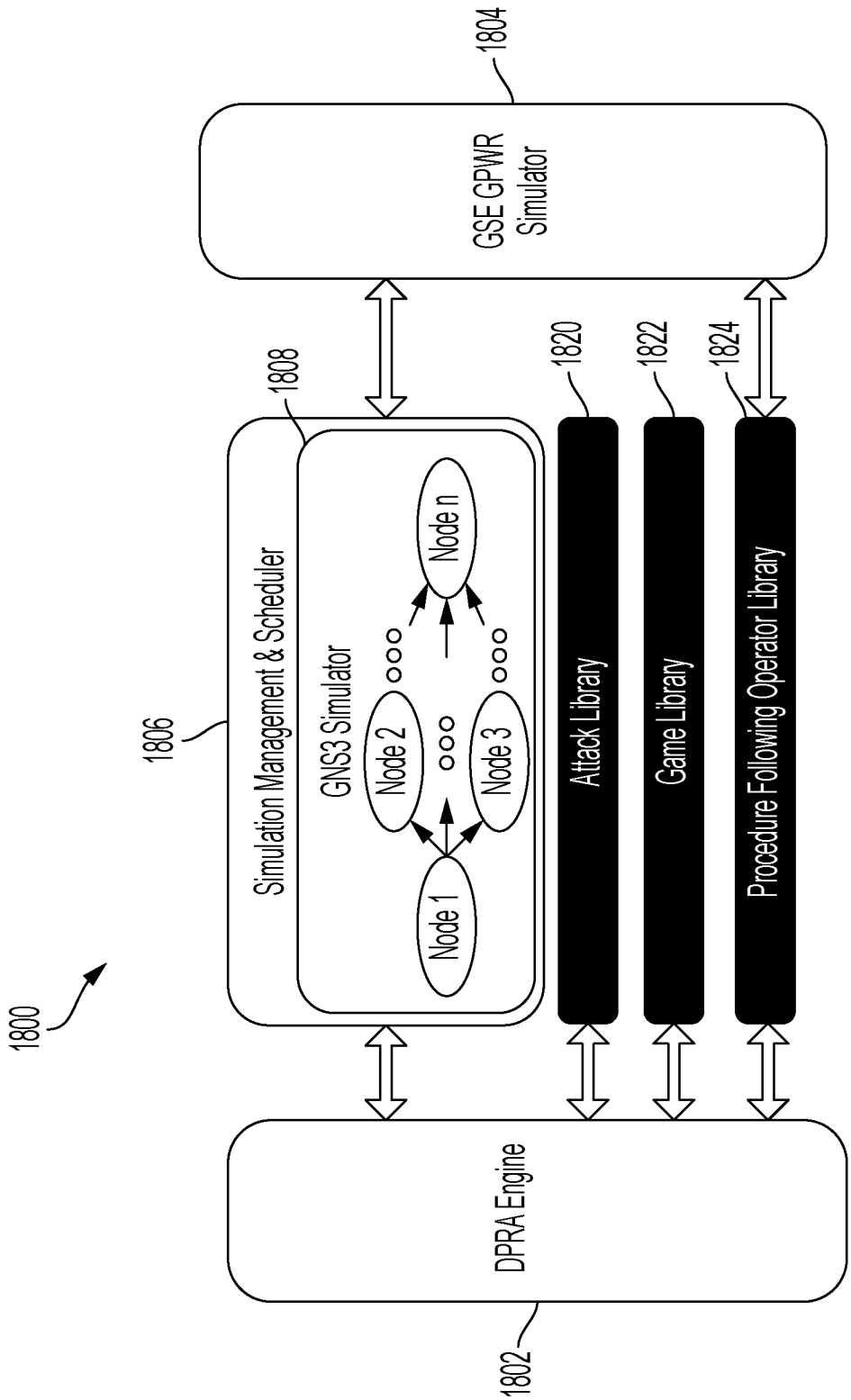

1800
1802
DPRA Engine
1804
GSE GPWR Simulator
1806
Simulation Management & Scheduler
1808
GNS3 Simulator
Node 1
Node 2
Node 3
Node n
1820
Attack Library
1822
Game Library
1824
Procedure Following Operator Library

FIG. 18

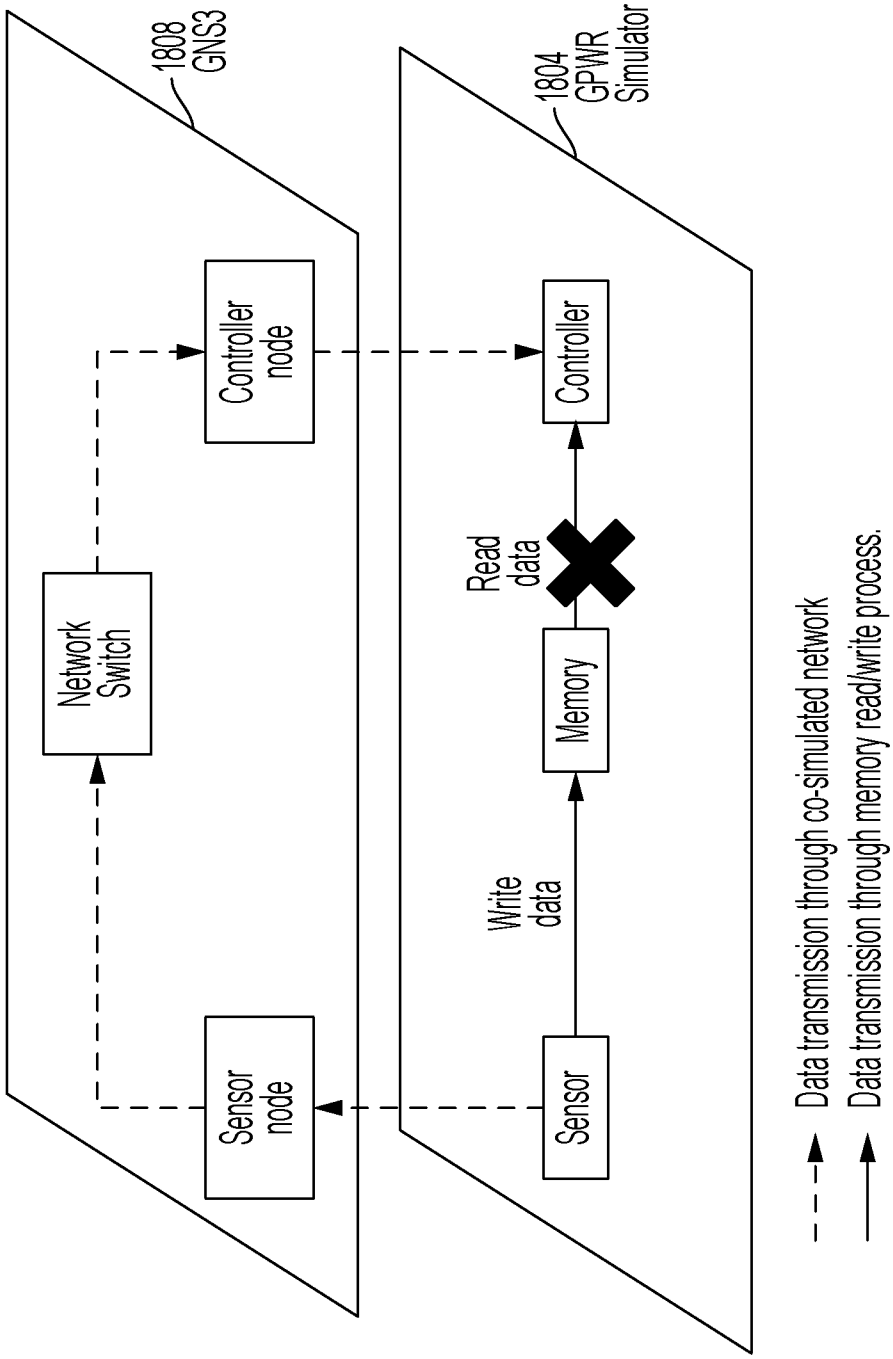

1808
GNS3
1804
GPWR
Simulator
Controller
node
Network
Switch
Sensor
node
Controller
Read
data
Memory
Write
data
Sensor
Data transmission through co-simulated network
Data transmission through memory read/write process.

FIG. 19

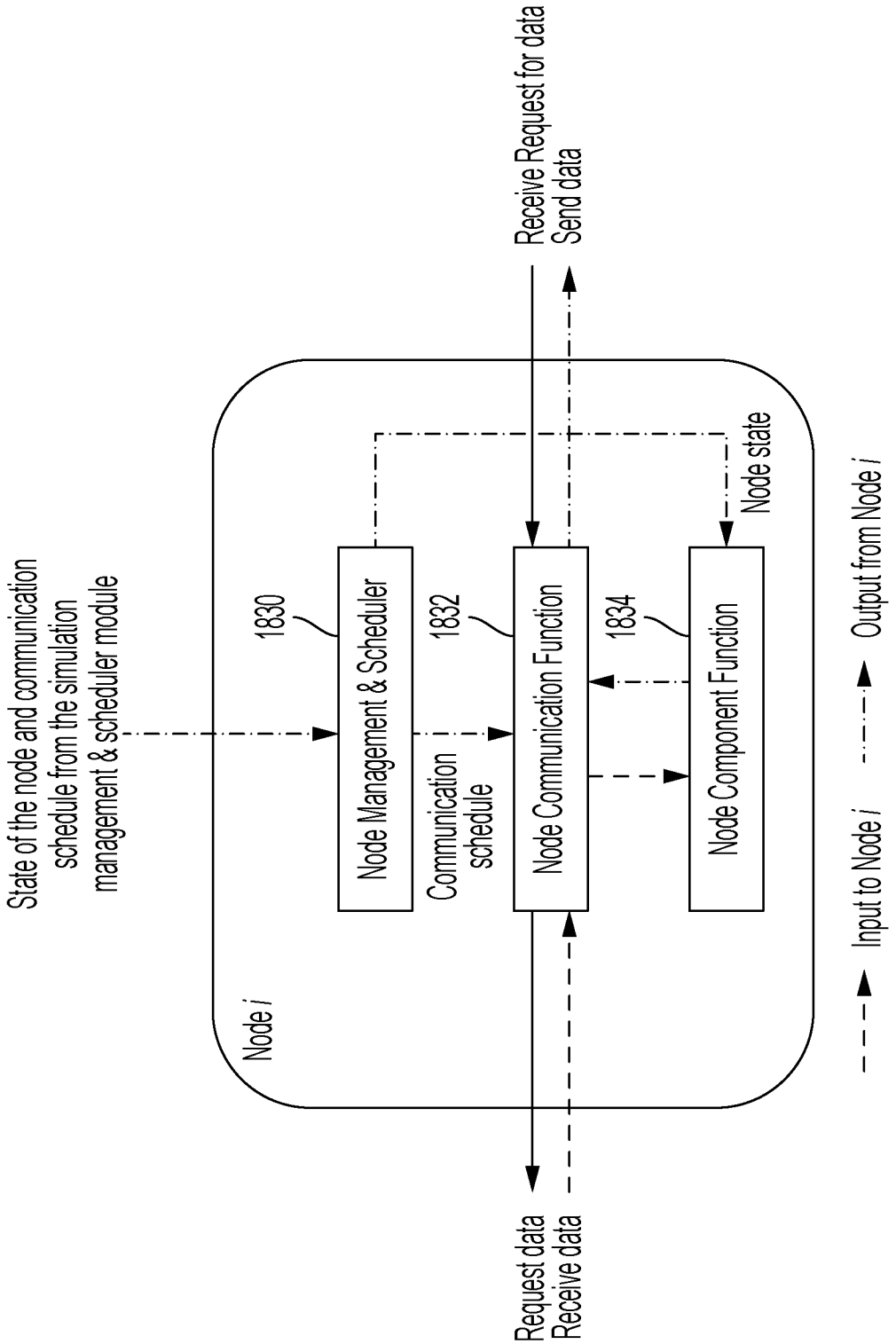

State of the node and communication schedule from the simulation management & scheduler module
Node i
1830
Node Management & Scheduler
1832
Communication schedule
Node Communication Function
1834
Node Component Function
Node state
Receive Request for data
Send data
Request data
Receive data
Input to Node i
Output from Node i

FIG. 20

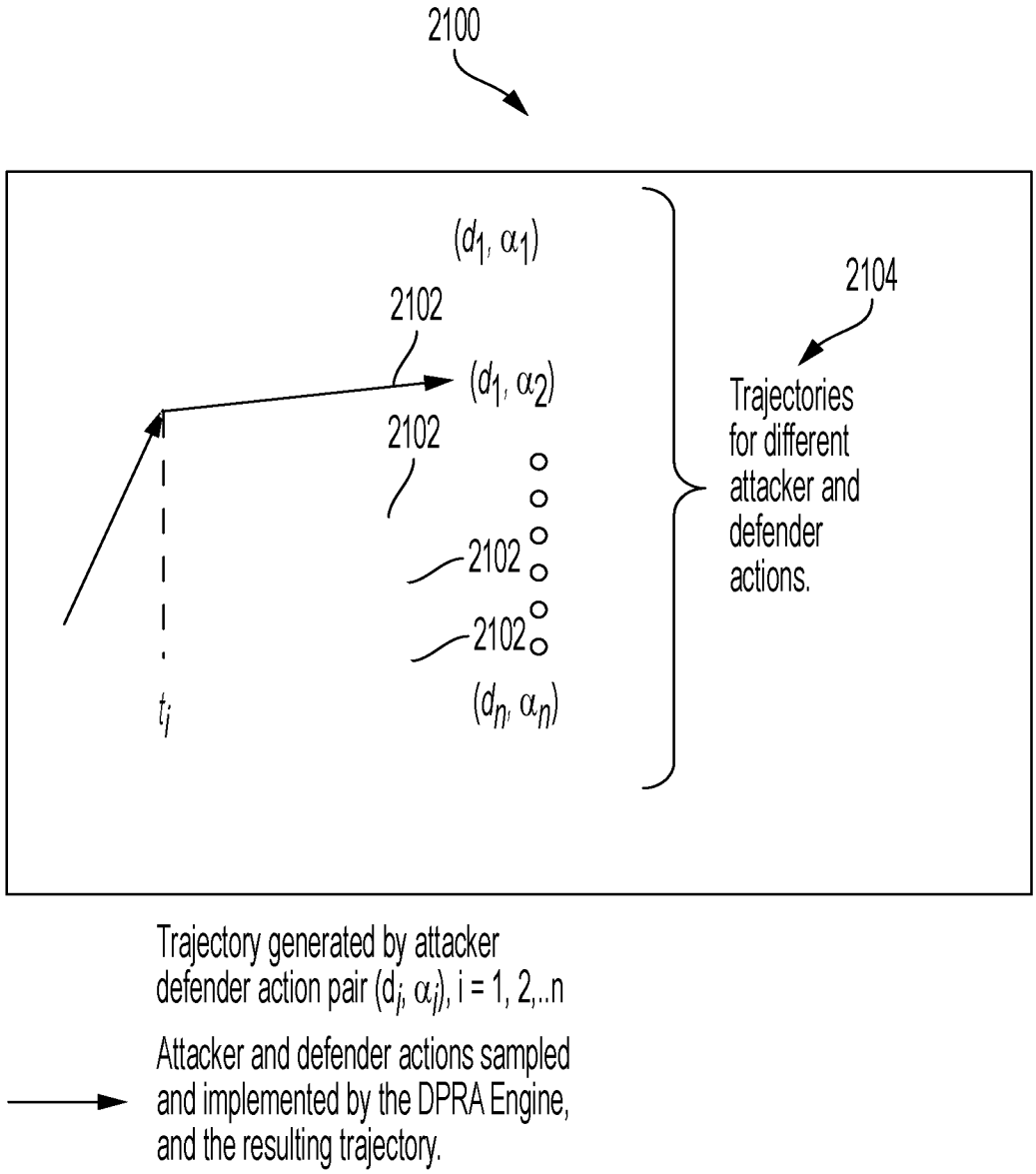

2100
(d_1, α_1)
2102
2104
(d_1, α_2)
2102
Trajectories for different attacker and defender actions.
2102
2102
t_j
(d_n, α_n)
Trajectory generated by attacker defender action pair (d_i, α_i), i = 1, 2,..n
Attacker and defender actions sampled and implemented by the DPRA Engine, and the resulting trajectory.

FIG. 21

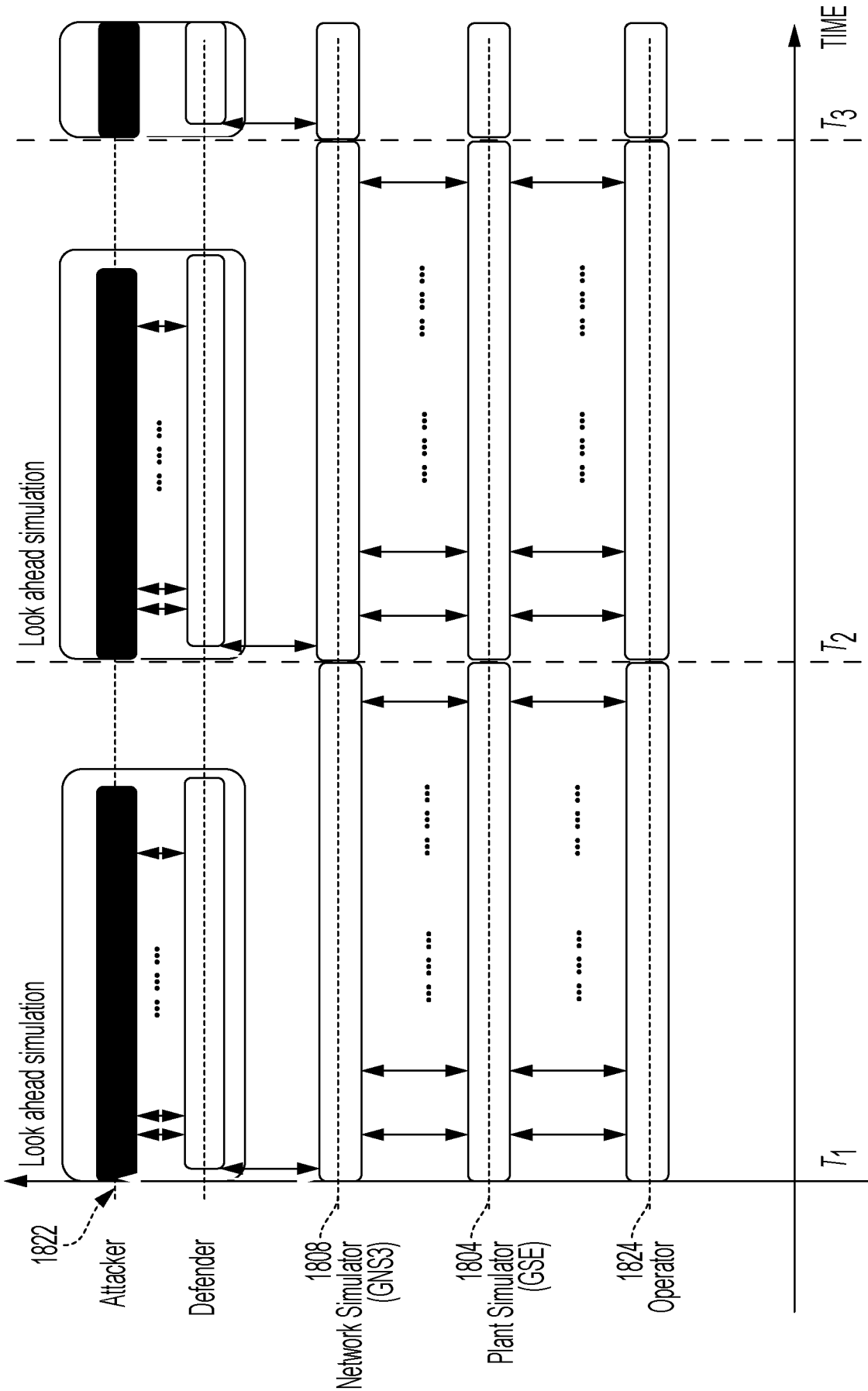

TIME
T_3
T_2
T_1
Look ahead simulation
Look ahead simulation
1822
Attacker
Defender
1808
Network Simulator
(GNS3)
1804
Plant Simulator
(GSE)
1824
Operator

FIG. 22