Safety-Critical Control Synthesis for network systems with Control Barrier Functions and Assume-Guarantee Contracts

Yuxiao Chen, James Anderson, Karan Kalsi, Aaron D. Ames, and Steven H. Low

***Abstract*—This paper presents a contract based framework for safety-critical control synthesis for network systems. To handle the large state dimension of such systems, an assume-guarantee contract is used to break the large synthesis problem into smaller subproblems. Parameterized signal temporal logic (pSTL) is used to formally describe the behaviors of the subsystems, which we use as the template for the contract. We show that robust control invariant sets (RCIs) for the subsystems can be composed to form a robust control invariant set for the whole network system under a valid assume-guarantee contract. An epigraph algorithm is proposed to solve for a contract that is valid, —an approach that has linear complexity for a sparse network, which leads to a robust control invariant set for the whole network. Implemented with control barrier function (CBF), the state of each subsystem is guaranteed to stay within the safe set. Furthermore, we propose a contingency tube Model Predictive Control (MPC) approach based on the robust control invariant set, which is capable of handling severe contingencies, including topology changes of the network. A power grid example is used to demonstrate the proposed method. The simulation result includes both set point control and contingency recovery, and the safety constraint is always satisfied.**

I. Introduction

Safety-critical systems refer to the systems where a violation of the safety constraint might lead to severe consequences such as loss of life and large economic loss. One such example is the power grid. It is well known that if not controlled properly, large-scale blackouts may occur, causing severe economic losses, sometimes even life losses due to the power loss at important facilities such as the hospitals and factories. Correct-by-construction control synthesis has seen recent success in safety-critical applications such as vehicle control [1], [2] and robot navigation [3]. It refers to a collection of methods based on concepts such as reachable sets and control invariant sets [4] to synthesize a controller that is capable of enforcing safety constraints. Informally, a *robust control invariant set* $\mathcal{S}$ is a subset of the state space, such that a dynamical system initiated from within $\mathcal{S}$ can be controlled to stay in $\mathcal{S}$ for all future time, in the presence of disturbances. Typically, correct-by-construction control synthesis relies on computational tools such as the Hamilton Jacobi PDE [5], Linear Matrix Inequalities (LMIs) [6], and sum-of-squares (SOS) programming [7], [8]. Unfortunately, these methods do not scale well with the state-dimension of the system. This problem, sometimes referred to as "the curse of dimensionality," limited the applications of correct-by-construction control synthesis to systems with low state dimension. There has been effort to break "the curse of dimensionality," which, at the system level, typically utilizes either compositional analysis or system symmetry [9], [10], [11], [12]. Alternatively, advances in numerical methods are also being made [13], [14].

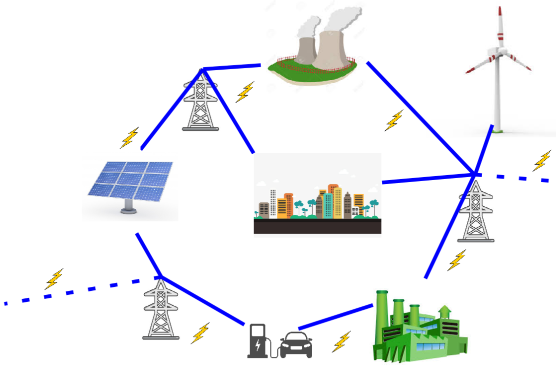

Fig. 1: Power grid with generator buses and load buses

To the best of the authors' knowledge, the synthesis of robust invariant sets for network systems with heterogeneous subsystems and strong coupling between them remains an open problem. Power grids are prominent examples of systems that exhibit the problematic phenomena just described. Typically, they consist of various types of generation buses e.g., hydroelectric, solar, and wind plants, and load buses all coupled via transmission lines, as shown in Fig. 1.

The approach we propose to break the "curse of dimension" is the assume-guarantee contract [15], [16], which decomposes the overall performance guarantee into individual contracts for the different parts of the system. Under a network system setting, every node in the network can take the performance guarantee from other nodes as assumptions and in turn give its own performance guarantee, which then becomes part of the assumptions for other nodes in the network. In this way, the big synthesis problem is decomposed into small subproblems. For discrete transition systems, there exist algorithms that automatically generate assume-guarantee contracts [17]. However, for dynamical systems with a continuous state space, there exists no efficient method that generates a valid assume-guarantee contract automatically.

The contributions of this paper are:

- We propose the formulation of an assume-guarantee contract approach to compute *robust control invariant sets* (RCIs) for networked systems, and prove set invariance

Yuxiao Chen and Aaron D. Ames are with the Department of Mechanical and Civil Engineering, Caltech, Pasadena, CA, 91106, USA. Emails: {chenyx, ames}@caltech.edu

James Anderson and Steven H. Low are with the Computing and Mathematical Sciences Department, Caltech, Pasadena, CA, 91106, USA. Emails: {james, slow}@caltech.edu

Karan Kalsi is with Pacific Northwest National Laboratory, Richland, WA, 99352, USA. Email: Karanjit.Kalsi@pnnl.gov

This work is supported by the Battelle Memorial Institute, Pacific Northwest Division, Grant #424858.

of an RCI for a network system which is composed of subsystem RCIs with a network assume-guarantee contract.

- We propose an epigraph algorithm that searches for valid assume-guarantee contracts. The algorithm has a computational complexity that scales linearly with system size (assuming the system graph is sparse or the coupling signals from multiple neighbors are summable). Moreover, the epigraph algorithm is general-purpose and can be combined with any RCI computation method to compute RCIs for network systems.
- We propose a contingency tube MPC algorithm based on the assume-guarantee contract for set invariance, which is able to handle severe contingencies such as a change in the network topology.

Under nominal working conditions, we show that the computed RCI, together with control barrier functions (CBF) guarantees that the state never leave the RCI under disturbances such as load and generation fluctuation, which in turn guarantees the safety constraint. During severe contingencies such as line loss and shortcut, we extend the RCI framework to propose a contingency tube MPC algorithm that when feasible, guarantees a smooth transition to the new set point without violating the safety constraint. By carefully analyzing the model uncertainty and communication constraint, we show that the contingency tube MPC protocol can be implemented in real time, is applicable to the nonlinear dynamics of the power grid, and respects the communication constraint.

The paper significantly extends the conference version [18] in the following aspects. (1) We include more detail of the network assume-guarantee contract (2) We present the contingency tube MPC algorithm based on the RCI algorithm, which is able to handle contingencies that cause the operating point to change. (3) We show simulation results of the proposed method with the high-fidelity simulation environment PST [19] on a network with considerable size (39 buses).

In the remainder of the paper, Section II presents the dynamic model of the power grid and the problem setup; Section III reviews the major tools necessary for the proposed method, including a robust linear programming algorithm for robust invariant set computation, control barrier functions, and parameterized assume-guarantee contracts; Section IV presents the main result of this paper, proving set invariance with assume-guarantee contract for network systems; Section V presents the epigraph algorithm that searches for a valid assume-guarantee contract with convex optimization; the application of the proposed method on power grid is explained in Section VI; then Section VII presents the contingency tube MPC formulation for smooth operating point transition and finally we conclude in Section VIII.

Nomenclature For the remainder of the paper, $\mathbb{N}$ denotes the set of natural numbers, $\mathbb{R}$ denotes the set of real numbers, $\mathbb{B} = \{0, 1\}$ denotes the set of binary numbers. $\mathbb{R}^n$ denote the Euclidean space and $\mathbb{R}^n_{\geq 0}$ denotes the positive orthant. We use $p \in \mathcal{P}$ to denote a parameter, with $\mathcal{P}$ as its domain. For a variable $x \in \mathcal{X}$, $x(t)$ denotes its value at the t-th time instance, the bold form $\mathbf{x} = x(0)x(1)x(2)... \in \mathcal{X}^\omega$ denotes the infinite evolution trajectory of x for $t = 0, 1, ...$ Correspondingly, $\mathcal{X}^\omega$ denotes the space of all possible evolutions of x. To avoid confusion, in a value iteration process, $p[i]$ denotes the value of a parameter p after the i-th iteration. For a vector $x \in \mathcal{X} = \mathcal{X}_1 \times ... \times \mathcal{X}_n$, $x \downarrow \mathcal{X}_i$ denotes the projection of x onto $\mathcal{X}_i$. $\mathbf{Poly}(P, q) = \{x \mid Px \leq q\}$ denotes a polytope defined with matrix P, q.

II. Problem setup

In this section, we present the problem setup and show how the power grid control synthesis can be handled with the proposed method.

A. Network system dynamics

Although this paper is motivated by a power grid control problem, we will present the proposed methodology under a more general network control context since it can be extended to other applications. We consider a network dynamic system consisting of subsystems with coupling dynamics. Each subsystem treats the coupling between neighboring subsystems as bounded disturbances. Therefore, the following product of subsystems is considered: [1]

$$\Sigma = \Sigma_1 \times \Sigma_2 \times ... \times \Sigma_N. \tag{1}$$

It is assumed that each subsystem can be written in the form

$$\Sigma_i := \begin{cases} x_i^+ = f_i\left(x_i, y_{\mathcal{N}_i}, u_i, d_i\right), \\ y_i = h_i\left(x_i\right), \end{cases} \tag{2}$$

where $x_i \in \mathcal{X}_i \subseteq \mathbb{R}^{n_i}$ is the i^{th} current state and x_i^+ denotes the successor state. The control input is $u_i \in \mathcal{U}_i \subseteq \mathbb{R}^{m_i}$, the exogenous disturbance is $d_i \in \mathcal{D}_i \subseteq \mathbb{R}^{l_i}$, and $y_{\mathcal{N}_i}$ denotes the vector of signals consisting of the outputs of all of the neighboring subsystems connected to subsystem Σ_i. The vector $y_{\mathcal{N}_i}$ can be further decomposed as

$$y_{\mathcal{N}_i} = \begin{bmatrix} y_{j_1} \\ \vdots \\ y_{j_{N_i}} \end{bmatrix} \text{for all } j_1, \ldots, j_{N_i} \in \mathcal{N}_i, \tag{3}$$

where $\mathcal{N}_i$ is the neighbor set of the i^{th} node with cardinality $|\mathcal{N}_i| = N_i$. The full dynamics of the networked system takes the form:

$$f(x, u, d) = \begin{bmatrix} f_1(x_1, y_{\mathcal{N}_1}, u_1, d_1) \\ \vdots \\ f_N(x_N, y_{\mathcal{N}_N}, u_N, d_N) \end{bmatrix}, h(x) = \begin{bmatrix} h_1(x_1) \\ \vdots \\ h_N(x_N) \end{bmatrix}. \tag{4}$$

The overall state space and output space are denoted as $\mathcal{X} = \mathcal{X}_1 \times ... \times \mathcal{X}_N$ and $\mathcal{Y} = \mathcal{Y}_1 \times ... \times \mathcal{Y}_N$, respectively. Since the method was first proposed for fixed point control, without loss of generality, it is assumed w.l.o.g. that the equilibrium point is at the origin, i.e. $f(0, 0, 0) = 0$ and that $h(0) = 0$.

Given the dynamics, the behavior of the i^{th} subsystem is uniquely determined by $x_i(0)$, $\mathbf{y}_{\mathcal{N}_i}$, $\mathbf{u}_i$ and $\mathbf{d}_i$, let $\mathcal{I}_i = \mathcal{X}_i \times$

[1] Note that for a general network dynamical system, the corresponding model would be defined over a graph structure [20]; as noted, in the context of this paper, because we view the coupling between systems as bounded disturbances, we can consider a network of dynamical systems as simply the product system.

$\mathcal{Y}^{\omega}_{\mathcal{N}_i} \times \mathcal{U}^{\omega}_i \times \mathcal{D}^{\omega}_i$ denote the space of input signals and initial conditions of the system Σ_i and $\mathcal{X}^{\omega}_i$ is the space of all possible state signals of Σ_i. A dynamic system $\Sigma_i \subseteq 2^{\mathcal{I}_i} \times 2^{\mathcal{X}^{\omega}_i}$ is understood as a subset of possible input and state signal pairs.

Later in the paper we will consider the task of appending safety constraint to the dynamical system, such constraints are on the states of the subsystems.

***Remark* 1.** The results in this paper can easily be extended to the case of continuous-time dynamical systems. However, the methods we use to compute robust control invariant sets are most naturally presented in discrete-time, hence our choice.

B. *Power grid dynamics*

We now present the dynamic model of the power grid and control problem we seek to solve. This includes defining appropriate safety constraints.

Power system control is an important network control application, in this work we consider the problem of load-side primary frequency control [21], [22]. There has been a lot of effort focusing on the stability, optimality, and safety of power networks, see for example the survey papers [23], [24]. Specifically, the Optimal Load Control (OLC) algorithms in [21], [22] provide control laws that can asymptotically track an optimal load-control problem i.e., control policy method achieves good asymptotic performance that maximizes economic benefit [21], [22]. To be more specific, the virtual flow method proposed in [22] formulates an OLC problem and derives a control policy based on a primal-dual update of the Lagrangian. We shall use this controller as the legacy controller to demonstrate the capability of the CBF controller proposed in Section III-B. However, despite good asymptotic performance, it lacks a performance guarantee in the transient phase. In particular, when sudden changes such as failure of a component or a short circuit at one of the nodes. Drastic frequency changes should be avoided since it may lead to a severe damage to the system and a heavy economic loss. With increasing penetration of renewable and distributed energy resources, independent system operators will need to shift towards network based, i.e. distributed, algorithms with a focus on safety and performance constraints.

Robust control invariant set with control barrier functions is a good complement to the OPF controller since it guarantees set invariance with minimum intervention and preserves the good performance of the OPF controller when the violation of safety constraints is not imminent. The proposed procedure is to first compute an RCI for the grid dynamics, then implement a CBF based on the RCI as the supervisory controller. The CBF remains inactivated under normal situations and let the OPF controller operate, and will intervene when the state is about to leave the RCI, guaranteeing that the safety constraint is always satisfied.

We consider a transmission model of the power grid. The network consists of two types of buses, generator buses and load buses. Take the IEEE 9-bus test case shown in Fig. 2 as an example.

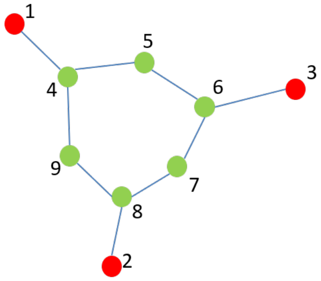

Fig. 2: IEEE 9 bus network (red nodes are generator buses, green nodes are pure load buses)

The generator buses are $\mathcal{G} = \{1, 2, 3\}$ and the load buses are $\mathcal{L} = \{4, 5, 6, 7, 8, 9\}$. Edges between adjacent nodes imply there is a transmission line connecting them. The dynamics of the grid can be described by the following model [22], [25]:

$$\begin{aligned} \dot{\theta}_i &= \omega_i, \\ M_i \dot{\omega}_i &= P^{in}_i - D_i \omega_i - r_i - u_i - \sum_{j \in \mathcal{N}_i} \frac{V_i V_j}{X_{ij}} \sin(\theta_i - \theta_j), i \in \mathcal{G} \\ 0 &= P^{in}_i - D_i \omega_i - r_i - u_i - \sum_{j \in \mathcal{N}_i} \frac{V_i V_j}{X_{ij}} \sin(\theta_i - \theta_j), i \in \mathcal{L}, \end{aligned} \tag{5}$$

where θ_i and ω_i are the phase angle and frequency respectively of the voltage at bus i, P^{in}_i and r_i are the input power and uncontrollable load at bus i. A sudden change to either P^{in}_i or r_i is the main source of disturbance to the system. We let u_i denote a controllable load, which is used to regulate bus i. For a generator bus, M_i is the inertia constant of generator i and D_i is the damping coefficient; for a load bus, there is zero inertia and ω_i is determined by an algebraic equation. A generator bus is modeled with 2 states ($x_i = [\theta_i, \omega_i]^\intercal$); and a load bus is modeled with 1 state ($x_i = \theta_i$). The voltage at bus i is V_i, which is assumed to be constant. X_{ij} is the reactance of the circuit between bus i and bus j, hence a smaller reactance leads to a stronger coupling. We choose the output to be $y_i = \theta_i$ since the coupling between buses occurs through the phase angles θ_i. In addition to the nonlinear model (5), we will also make use of a linearization about its steady state:

$$\begin{aligned} \delta\dot{\theta}_i &= \omega_i, \\ M_i \dot{\omega}_i &= -D_i \omega_i - r_i - u_i - \sum_{j \in \mathcal{N}_i} B_{ij}(\delta\theta_i - \delta\theta_j), i \in \mathcal{G} \\ 0 &= -D_i \omega_i - r_i - u_i - \sum_{j \in \mathcal{N}_i} B_{ij}(\delta\theta_i - \delta\theta_j), i \in \mathcal{L}, \end{aligned} \tag{6}$$

where $B_{ij} = \frac{V_i V_j}{X_{ij}} \cos(\theta^0_i - \theta^0_j)$ represents the sensitivity of the power flow to phase variations and θ^0_i is the steady-state phase angle at bus i. Note that B_{ij} is nonzero when bus i and bus j are neighbors. From (6) the subsystem dynamics Σ_i are given by

$$\begin{bmatrix}\delta\dot{\theta}_i\\ \dot{\omega}_i\end{bmatrix}=\begin{bmatrix}0 & 1\\ \frac{-\sum\limits_{j\in\mathcal{N}_i}B_{ij}}{M_i} & \frac{-D_i}{M_i}\end{bmatrix}\begin{bmatrix}\delta\theta_i\\ \omega_i\end{bmatrix}+\begin{bmatrix}0\\ -M_i^{-1}\end{bmatrix}u_i$$

$$+\begin{bmatrix}0 & \cdots & 0\\ \frac{B_{ij_1}}{M_i} & \cdots & \frac{B_{ij_{N_i}}}{M_i}\end{bmatrix}\begin{bmatrix}\delta\theta_{j_1}\\ \vdots\\ \delta\theta_{j_{N_i}}\end{bmatrix}+e_i, \qquad i\in\mathcal{G},$$

$$\delta\dot{\theta}_i=\frac{-\sum\limits_{j\in N_i}B_{ij}}{D_i}\delta\theta_i+\frac{-1}{D_i}u_i+\frac{\sum\limits_{j\in\mathcal{N}_i}B_{ij}\delta\theta_j}{D_i}+e_i,\quad i\in\mathcal{L}. \tag{7}$$

with output $y_i=\delta\theta_i$.

The **control objective** is to prevent large frequency deviation from a set value for the dynamics in (5). However, since the coupling is via the phase angle differences, in order to bound the frequency deviation, one needs to bound phase angle deviations as well. We will thus construct a robust control invariant set for both the phase angle and the frequency. The RCI should provide robustness to sudden changes of the input power P_i^{in}, uncontrollable load r_i and the coupling between neighboring buses. We will treat frequency deviation as the **safety constraint**, i.e., the danger set $\mathcal{X}_i^d$ for a generator bus Σ_i is defined as

$$\mathcal{X}_i^d=\left\{[\delta\theta_i,\omega_i]^\intercal \mid |\omega_i|\geq\omega^{\max}\right\}, \tag{8}$$

where $\omega^{\max}$ is the bound for frequency deviation.

The RCI is implemented with control barrier functions, which acts as a supervisory controller on top of the legacy controller. We choose the OLC controller presented in [22] as the legacy controller, but it should be noted that the CBF supervisory controller can work with any legacy controller and enforce safety on top of it.

For each node in the power grid, the coupling between neighboring nodes happens via a scalar output, the phase angle. If we can use assume-guarantee contract to put bound on the phase angle deviations, we can compute an RCI for each node, which in turn constitute an RCI for the whole power grid network.

III. Review of major tools

In this section, we review the major tools necessary to our approach, including the robust linear programming method for RCI computation, control barrier functions, and parameterized assume-guarantee contracts.

A. Robust linear programming for RCI computation

The key concept to guarantee safety is the robust control invariant set. We start with the definition.

Definition 1. Given a discrete-time dynamical system defined as

$$x^+=f(x,u,w), x\in\mathbb{R}^n, u\in\mathcal{U}, w\in\mathcal{W} \tag{9}$$

where x, u, and w are the state, control input, and disturbance. A set $\mathcal{S}\subseteq\mathbb{R}^n$ is *robust control invariant* if $\forall x(0)\in\mathcal{S},\ \forall\mathbf{w}\in\mathcal{W}^\omega$,

$$\exists\mathbf{u}\in\mathcal{U}^\omega\quad\text{s.t.}\quad\forall t=0,1,2,...,x(t)\in\mathcal{S}.$$

In particular, for a discrete-time dynamical system shown in (9), the forward invariance condition can be conveniently written as $\forall x\in\mathcal{S},\ \forall w\in\mathcal{W}, \exists u\in\mathcal{U},$ s.t. $x^+=f(x,u,w)\in\mathcal{S}$. In addition, we assume $w=[w^m;w^u]$, where w^m and w^u are the measured and unmeasured disturbances, respectively. The difference between them is that the control policy can depend on w^m, but not on w^u. In the power grid example, w consists of the exogenous disturbance d_i and coupling power flow from the neighboring nodes, and we will discuss more details about measurability later in this section.

There are two types of invariant sets that are relevant to control synthesis; the maximal control invariant set and the minimal control invariant set. The former was formally defined in [26], which can be thought of as the region of attraction for a controlled dynamic system. The minimal robust control invariant set (mRCI) describes the smallest invariant set a controller can maintain under disturbances and uncertainties. In the context of assume-guarantee contracts for network systems which we describe in Section III-D, an mRCI is clearly more relevant since the assumption about the coupling from neighboring nodes depends on the size of the invariant sets, and we want those bounds to be as small as possible.

We review the robust linear programming algorithm for minimal robust control invariant set computation, originally proposed in [27]. However, it should be noted that the contract-based framework and the epigraph algorithm introduced in Section V, are compatible with any RCI algorithm, we simply present the robust linear programming algorithm for completeness of the paper.

The original mRCI algorithm proposed in [27] involves a system identification step, which is not necessary if the model is known. In this paper, we assume that the model for the power grid dynamics is known, including the characterization of the model uncertainty, which simplifies the mRCI computation. We briefly review the method and present the setup for the mRCI computation.

The robust linear programming algorithm assumes a discrete-time linear model:

$$x^+=Ax+Bu+Ew. \tag{10}$$

The RCI takes a polytopic form $\mathbf{Poly}(P,q)$ with the hyperplane orientation fixed to P. The basic operation is called a one-step propagation, which computes a new polytope $\mathbf{Poly}(P,q^+)$ that contains all possible x^+ with all possible $x\in\mathbf{Poly}(P,q)$, and $w\in\mathcal{W}$ under the dynamics in (10). Robust linear programming [28] is used to obtain $\mathbf{Poly}(P,q^+)$. The exact formulation is given later in this section. We assume the control law takes the form $u=K_{ff}w^m+K_{fb}x$, but note that this can be easily changed without affecting the algorithm for computing the mRCI. Moreover, once the mRCI is computed, it is enforced by control barrier functions, the linear control law here is simply used to show that there exists a control strategy that renders the set robustly control invariant, it does not have to be implemented.

In the power grid case, the dynamics Σ_i for each subsystem is written in (7). The bound for $\delta\theta_i$ is given by the assume-guarantee contract, and is discussed in more detail in Section IV and V. We assume that phase angles of the neighboring nodes and the local generation and uncontrolled load are measured disturbances. The unmeasured disturbance is due to the communication delay between the neighboring nodes. For example, suppose the i^{th} node has one neighbor, the j^{th} node, the bound on frequency is $\omega^{\max}$, and the time delay of communication is τ. Then the maximum difference between the actual value of θ_j and the value used for feedback is $\omega^{\max}\tau$. The bound of the unmeasured disturbance for the i^{th} node w_i^u is then given as

$$|w_i^u| \le \left|B_i K_{ff}^i\right| \omega^{\max}\tau \tag{11}$$

where B_i K_{ff}^i is the input matrix and feedforward gain of the i^{th} node.

This continuous-time linear model is then discretized and fits the setup of the robust linear programming mRCI algorithm. The following one-step propagation solves for a polytopic set $\mathbf{Poly}(P, q^+)$ that contains all possible x^+ with $x \in \mathbf{Poly}(P, q)$ and $w \in \mathcal{W}$:

$$\begin{aligned} \min_{K_{ff},K_{fb},q^+} \quad & c^\intercal q^+ \\ & \text{s.t.} \forall x \in \mathbf{Poly}(P,q), \forall w \in \mathcal{W}, \\ & P\left(Ax + B\left(K_{ff}^\intercal w^m + K_{fb}^\intercal x\right) + Ew\right) \le q^+, \\ & K_{ff}^\intercal w^m + K_{fb}^\intercal x \in \mathcal{U}, \end{aligned} \tag{12}$$

which is solvable with linear programming after dualization [28].

Remark 2. We enforce an additional constraint that for the generator buses, the frequency stays bounded $|\omega_i| \le \omega^{\max}$, which is easily enforced as a constraint on q^+.

With the one-step propagation solvable, the iterative algorithm starts with a small q and iteratively updates q with q^+. If $q^+ \le q$, then the set is robustly control invariant, and the algorithm terminates, as shown in Algorithm 1. See [27] for detail.

Algorithm 1 Robust LP algorithm for mRCI

```
procedure RCI-IO($\Sigma$, $P$, $q^0$, $\mathcal{W}$, $\mathcal{U}$, $\epsilon$)
    $q \leftarrow q^0$
    do
        Find $\left[q^+, K_{ff}, K_{fb}\right]$ s.t.
        $\forall x \in \mathbf{Poly}(P,q), \forall w \in \mathcal{W}, K_{ff}w^m + K_{fb}x \in \mathcal{U}$,
        $x^+ \in \mathbf{Poly}(P, q^+ - \epsilon\mathbf{1}_L)$
        $q \leftarrow q^+$
    while $q^+ \le q + \epsilon\mathbf{1}_L$
    return $[q, K_{ff}, K_{fb}]$
end procedure
```

B. Control barrier function

The computed robust control invariant set will be enforced with a control barrier functions (CBF). Control barrier functions were first proposed in [29], and improved in [30], where the authors proposed a quadratic programming framework that keeps the system safe with minimum intervention. Specifically, consider the dynamic system described in (7). Suppose there exists a function $b : \mathbb{R}^n \to \mathbb{R}$ that satisfies

$$\begin{array}{ll} \forall x \in \mathcal{X}_0, & b(x) \ge 0 \\ \forall x \in \mathcal{X}_d, & b(x) < 0 \\ \forall x \in \{x \mid b(x) \ge 0\}, \forall w \in \mathcal{W}, & \exists u \in \mathcal{U} \text{ s.t.} \\ & \qquad \dot{b} + \alpha(b) \ge 0, \end{array} \tag{13}$$

where $\mathcal{X}_0$ is the set of initial states and $\mathcal{X}_d$ is the danger set that we want to keep the state away from. $\alpha(\cdot)$ is a class-$\mathcal{K}$ function, i.e., $\alpha(\cdot)$ is strictly increasing and satisfies $\alpha(0) = 0$. Then, for any legacy controller, the CBF controller is a supervisory controller that enforces the state to stay inside $\{x \mid b(x) \ge 0\}$ for all possible disturbance $w \in \mathcal{W}$ with the following quadratic programming:

$$\begin{aligned} u^\star = \arg\min_{u\in\mathcal{U}} & \left\|u - u^0\right\|^2 \\ \text{s.t.} & \nabla b \cdot f(x, u, w) + \alpha(b) \ge 0, \end{aligned} \tag{14}$$

where u^0 is the input of the legacy controller. It can be shown that under mild conditions, one can construct a CBF from an RCI that contains $\mathcal{X}_0$ and not intersecting with $\mathcal{X}_d$.

The robust optimization algorithm (Algorithm 1) generates a polytopic RCI $\mathbf{Poly}(P, q)$, where P is a constant $m \times n$ matrix and $q \in \mathbb{R}^m_{>0}$. Note that the origin is always contained in the interior of the RCI. The CBF is defined as

$$b(x) = \min_k \frac{q_k - P_k x}{q_k}.$$

Let $\mathcal{X}_0$ be the origin and let $\mathcal{X}_d$ be defined in (8) for the generator buses (there is no $\mathcal{X}_d$ for pure load buses), it is easy to verify that the CBF defined above satisfies (13). Then, the RCI can be enforced with the convex quadratic programming in (14).

C. Parameterized Signal Temporal Logic

To break the “curse of dimensionality” for large network systems, we use assume-guarantee contract to decompose the synthesis problem for the whole network into smaller subproblems for the subsystems. This work differs from the assume-guarantee approach used in the synthesis for transition systems [15], [31] in that here we deal with a continuous input and state space rather than discrete states and actions. The language for writing the specification is Signal Temporal Logic (STL), which is an extension of Linear Temporal Logic that allows for real time and predicates over reals [32], [33], [34]. A Signal Temporal Logic formula $\phi : \mathcal{X}^\omega \to \mathbb{B}$ is written using the following grammar:

$$\phi = \top \mid \mu \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \mathbf{U}_I \phi_2, \tag{15}$$

where $\top$ is the logical tautology, $\mu : \mathcal{X} \to \mathbb{B}$ is a logic proposition, $\neg$ is Boolean negation, $\wedge$ is the Boolean **AND**, and I is an interval of time. The validity of a formula μ with respect to a signal $\mathbf{x}$ at time t can be determined as

$$\begin{array}{lll}
(\mathbf{x},t)\models\mu & \text{iff} & x(t) \text{ satisfies } \mu \\
(\mathbf{x},t)\models\neg\phi & \text{iff} & x(t)\not\models\phi \\
(\mathbf{x},t)\models\phi_1\wedge\phi_2 & \text{iff} & x(t)\models\phi_1 \text{ and } x(t)\models\phi_2 \\
(\mathbf{x},t)\models\phi_1\mathbf{U}_{[a,b]}\phi_2 & \text{iff} & \begin{array}{l}\exists t'\in t+[a,b] \text{ s.t. } x(t)\models\phi_2 \\ \text{and } \forall t''\in[t,t'], x(t)\models\phi_1\end{array}
\end{array}$$

where $\models$ stands for "satisfy". A signal $\mathbf{x}\models\mu$ if $(\mathbf{x},0)\models\mu$.

From the above basic grammar, one can derive additional temporal operators $\Diamond_I\phi=\top\mathbf{U}_I\phi$, which means "$\phi$ is eventually true during I," and $\Box_I\phi=\neg(\Diamond_I\neg\phi)$, which means "$\phi$ is always true in I". When I is not specified, it is assumed that by default $I=[0,\infty)$.

Remark 3. An STL formula is extended to discrete-time signals by considering the sampling instances, as discussed in [35].

Given an STL formula ϕ, $L(\phi)=\{\mathbf{x}\in\mathcal{X}^\omega\mid\mathbf{x}\models\phi\}$ is the language of the formula. A partial order is defined among STL formulas as $\phi_1\preceq\phi_2$ if $\forall\mathbf{x}\in\mathcal{X}^\omega,(\mathbf{x}\models\phi_1)\Rightarrow(\mathbf{x}\models\phi_2)$, or equivalently, $L(\phi_1)\subseteq L(\phi_2)$.

A Parameterized Signal Temporal Logic (pSTL) formula is an STL formula with parameters. For example, $\phi=\Box_{[a,b]}(x\geq c)$ can be represented as the following pSTL: $\varphi(a,b,c)=\Box_{[a,b]}(x\geq c)$, where a,b and c are the parameters and $\varphi:\mathbb{R}^3\to(\mathcal{X}^\omega\to\mathbb{B})$ is the pSTL template. For the rest of the paper, it is assumed that all the pSTL formulas are defined on partially ordered parameter domains. Given a parameter domain $\mathcal{P}$, the partial order is denoted as $\leq_{\mathcal{P}}$. We adopt the definition of monotonicity of a pSTL formula from [36].

Definition 2. A pSTL formula $\varphi(p)$ is *monotonically increasing* if

$$\forall p_1,p_2\in\mathcal{P},\quad p_1\leq_{\mathcal{P}}p_2\Rightarrow\varphi(p_1)\preceq\varphi(p_2), \tag{16}$$

and *monotonically decreasing* if the inequality holds in the opposite direction.

For example, consider a formula that requires x to be larger than zero at some time during $t\in[0,p]$, where p is the parameter. It is written as $\varphi(p)=\Diamond_{[0,p]}(x\geq 0)$, which is monotonically increasing and $\varphi(p)=\Box_{[0,\infty)}(x\geq p)$, interpreted as x should always be positive during $t\in[0,p]$, is monotonically decreasing in p.

For a pSTL φ with parameter domain $\mathcal{P}_1$, if $\mathcal{P}_1$ is a subspace of $\mathcal{P}_2$, then $\forall p\in\mathcal{P}_2$, $\varphi(p)=\varphi(p_{\downarrow}\mathcal{P}_1)$, where $\downarrow$ denotes the projection of p onto $\mathcal{P}_1$.

D. Assume-Guarantee Contract for Network Systems

Finally, we present a framework that builds a large assume-guarantee contract from small subcontracts, which is then used for the computation of the RCI for the power network. First, we adopt the definition of assume-guarantee contract from [37]:

Definition 3 (Assume-Guarantee Contract)**.** An assume-guarantee contract $\mathcal{C}$ for the dynamic system Σ is a pair $[\phi_a,\phi_g]$ consisting of an assumption ϕ_a and a guarantee ϕ_g that encode the requirement that the logical implication $\phi_a\to\phi_g$ holds.

An assume-guarantee contract $\mathcal{C}=[\phi_a,\phi_g]$ is true for a dynamic system Σ if $\Sigma\cap L(\phi_a)\subseteq L(\phi_g)$, or written compactly as $\phi_a\wedge\Sigma\to\phi_g$ with a slight abuse of notation. Note that Σ here is understood as a proposition, interpreted as "a trace satisfies the system dynamics".

Definition 4 (Parameterized Assume-Guarantee Contract)**.** An assume-guarantee contract $\mathcal{C}=[\phi_a,\phi_g]$ is in parameterized form if there exists a pSTL $\phi_a=\varphi_a(p_a)$, a pSTL $\phi_g=\varphi_g(p_g)$ and a mapping $\lambda:\mathcal{P}_a\to\mathcal{P}_g$ such that $\mathcal{C}(p_a)=[\varphi_a(p_a),\varphi_g(\lambda(p_a))]$.

Here we want to emphasize the importance of λ, which maps the parameter for the assumption to the parameter for the guarantee. In particular, ϕ_a consists of two parts:

$$\phi_a=\phi_{ae}\wedge\phi_{af}=\varphi_{ae}\left(p_{ae}\right)\wedge\varphi_{af}\left(p_{af}\right), \tag{17}$$

where ϕ_{ae} is the specification for exogenous environment behavior and ϕ_{af} is the feedback specification, which is understood as the specification that changes with other contracts.

Definition 5 (Parameterized Network Assume-Guarantee Contract)**.** For a network defined in (1), a parameterized network assume-guarantee contract consists of individual parameterized assume-guarantee contracts $\mathcal{C}_i$ for each subsystem Σ_i. Let $p_{ae}\in\mathcal{P}_{ae}$, $p_{af}\in\mathcal{P}_{af}$ and $p_g\in\mathcal{P}_g$ be the parameters for φ_{ae}, φ_{af} and φ_g. Each subcontract $\mathcal{C}_i$ consists of $\phi_a^i=\varphi_{ae}^i(p_{ae}^i)\wedge\varphi_{af}^i(p_{af}^i)$ and $\phi_g^i=\varphi_g^i(p_g^i)$. where $p_{ae}^i=p_{ae}\downarrow\mathcal{P}_{ae}^i$, $p_{af}^i=p_{af}\downarrow\mathcal{P}_{af}^i$ and $p_g^i=p_g\downarrow\mathcal{P}_g^i$. Then the network assume-guarantee contract is defined as $\mathcal{C}=[\phi_{ae}\wedge\phi_{af},\phi_g]$ with the parameter mapping $\Lambda:\mathcal{P}_{ae}\times\mathcal{P}_{af}\to\mathcal{P}_g$ and

$$\begin{array}{rlll}
\phi_{ae}= & \varphi_{ae}(p_{ae})= & \displaystyle\bigwedge_{i=1}^{N}\phi_{ae}^i= & \displaystyle\bigwedge_{i=1}^{N}\varphi_{ae}^i(p_{ae}^i) \\
\phi_{af}= & \varphi_{af}(p_{af})= & \displaystyle\bigwedge_{i=1}^{N}\phi_{af}^i= & \displaystyle\bigwedge_{i=1}^{N}\varphi_{af}^i(p_{af}^i) \\
\phi_{g}= & \varphi_{g}(p_{g})= & \displaystyle\bigwedge_{i=1}^{N}\phi_{g}^i= & \displaystyle\bigwedge_{i=1}^{N}\varphi_{g}^i(p_{g}^i),
\end{array} \tag{18}$$

Remark 4. Note that some parameters may appear in more than one subcontract, the network contract parameters p_{ae}, p_{af} and p_g remove the repetition.

E. The Big Picture

Pulling all of the previous results and frameworks together we can summarize the problem and our solution as follows. Given a large-scale networked dynamical system, we would like to be able to compute a controller and verify that the system is robust to external perturbations and can satisfy various safety constraints. Achieving such an objective is computationally intractable in general. Our approach is to use assume-guarantee contracts to isolate subsystems in the network. We do this by assuming that the disturbance signals from neighboring subsystems caused by dynamic coupling satisfy a certain bound, and in return the given subsystem will guarantee not to output a signal that exceeds a given bound. Controllers that satisfy these safety and robustness guarantees that we specify using pSTL formulae can then be synthesized independently of each other by the RCI and CBF techniques. Moreover, upon interconnection, the resulting closed-loop system will be provably safe and robust.

IV. Set invariance with assume-guarantee contract

We now present one of the main results of this paper, which utilizes a network assume-guarantee contract to prove set invariance for network systems.

Theorem 1 (Assume-guarantee reasoning)**.** *Consider the network system in* (2) *associated with a parameterized network assume-guarantee contract defined in Definition 5 with parameter mapping* Λ*. Suppose the following are satisfied:*

1. Under the local mapping $\lambda_i : \mathcal{P}^i_{ae} \times \mathcal{P}^i_{af} \to \mathcal{P}^i_g$ *for each subsystem, the following subcontract* $\mathcal{C}_i : \Sigma_i \wedge \varphi^i_{ae}(p^i_{ae}) \wedge \varphi^i_{af}(p^i_{af}) \to \varphi^i_g(\lambda_i(p^i_{ae}, p^i_{af}))$ *is satisfied for all* $p^i_a \in \mathcal{P}^i_a \doteq \mathcal{P}^i_{ae} \times \mathcal{P}^i_{af}$*,*

2. There exists a mapping $\Gamma : \mathcal{P}_g \to \mathcal{P}_{af}$ *such that*

$$\varphi_g(p_g) \preceq \varphi^i_{af}(\gamma_i(p_g)), \tag{19}$$

where $\gamma_i(p_g) = \Gamma(p_g) \downarrow \mathcal{P}^i_{af}$*.*

3. There exists environment parameters $p_{ae} \in \mathcal{P}_{ae}$ *such that* $\varphi_{ae}(p_{ae})$ *is true.*

4. There exists an initial feedback parameter $p_{af}[0] \in \mathcal{P}_{af}$ *such that* $\varphi_{af}(p_{af}[0])$ *is true.*

Given p^i_{ae}*, define* $\hat{\lambda}_i(\cdot) = \lambda_i(p^i_{ae}, \cdot)$*. Let*

$$\hat{\Lambda}(p_{af}) = [\hat{\lambda}_1(p^1_{af})^\intercal,\ \hat{\lambda}_2(p^2_{af})^\intercal,\ \ldots\ \hat{\lambda}_N(p^N_{af})^\intercal]^\intercal, \tag{20}$$

then define recursively

$$\begin{aligned} p_g[k] &= \hat{\Lambda}(p_{af}[k]) \\ p_{af}[k+1] &= \Gamma(p_g[k]). \end{aligned} \tag{21}$$

Under these conditions, the network system satisfies

$$\hat{\phi}_g = \bigwedge_{k=0}^{\infty} \varphi_g(p_g[k]). \tag{22}$$

Proof. By assumption 3 and 4, p^i_{ae} and $p^i_{af}[0]$ exists so that ϕ^i_{ae} and $\phi^i_{af}[0]$ are satisfied. Therefore, we can build the following infinite sequence of pSTL that the network system satisfies from assumption 1 and 2 with (21):

$$\begin{aligned} &\bigwedge_{i=1}^{N} \varphi^i_{ae}(p^i_{ae}) \wedge \bigwedge_{i=1}^{N} \varphi^i_{af}(p^i_{af}[0]) \wedge \\ &\left(\bigwedge_{i=1}^{N} \varphi^i_{ae}(p^i_{ae}) \wedge \bigwedge_{i=1}^{N} \varphi^i_{af}(p^i_{af}[0]) \Rightarrow \bigwedge_{i=1}^{N} \varphi^i_g(p^i_g[0]) \right) \wedge \\ &\left(\bigwedge_{i=1}^{N} \varphi^i_g(p^i_g[0]) \Rightarrow \bigwedge_{i=1}^{N} \varphi^i_{af}(p^i_{af}[1]) \right) \wedge \\ &\ldots \end{aligned} \tag{23}$$

which implies (22). □

Theorem 19 can be viewed as the logic analogy of set invariance. If we have the recursive reasoning that propagates forward, and the initial logic proposition is satisfied, then all the subsequent propositions are satisfied. Here we use it on network assume-guarantee contracts where the subcontracts are for individual subsystems yet the recursive reasoning happens on the network level, i.e., the guarantees on subsystems' behavior are shared across the network as assumptions for the next iteration.

Next, we apply Theorem 1 to show set invariance of a network system. Consider the network system described in (2), suppose that all subsystem outputs, y_i, are scalars, and for each subsystem Σ_i, $y_{\mathcal{N}_i}$ is treated as a disturbance. Then given a bound on $y_{\mathcal{N}_i}$: $|y_{\mathcal{N}_i}| \le y^{\max}_{\mathcal{N}_i}$, a bound $\mathcal{D}_i$ on d_i and a bound $\mathcal{U}_i$ on u_i, we can apply standard RCI algorithm to compute an RCI $\mathcal{S}_i$ for Σ_i that satisfies

$$\begin{aligned} &\forall x_i \in \mathcal{S}_i, \quad \forall d_i \in \mathcal{D}_i, \quad \forall\, |y_{\mathcal{N}_i}| \le y^{\max}_{\mathcal{N}_i}, \\ &\exists u_i \in \mathcal{U}_i \ s.t. \quad x^+_i = f_i\left(x_i, y_{\mathcal{N}_i}, u_i, d_i\right) \in \mathcal{S}_i. \end{aligned}$$

Assume that $\mathcal{D}_i$ and $\mathcal{U}_i$ are given as part of the problem specification for all subsystems, the only information needed for RCI computation is $y^{\max}$. Let $\mathscr{F}$ be such a procedure that takes $y^{\max}$ as input, and computes an RCI. For clarity, we let $\mathscr{F}_i(y^{\max}_{\mathcal{N}_i}) \subseteq \mathcal{X}_i$ be an RCI computed by $\mathscr{F}$ for the i^{th} subsystem Σ_i, and let $\mathscr{F}(y^{\max}) \doteq \mathscr{F}_1(y^{\max}_{\mathcal{N}_1}) \times \ldots \times \mathscr{F}_N(y^{\max}_{\mathcal{N}_N})$ be the products of all the individual RCIs.

Remark 5. Given a fixed procedure $\mathscr{F}$, it can be thought of as a mapping from the parameter $y^{\max}$ to the RCIs for the subsystems, which is then used to enforce constraints on the state. Note that $\mathscr{F}(y^{\max})$ is simply the product of RCIs for the subsystems, not necessarily an RCI for the network system. It has to satisfy the validity condition defined later to be an RCI for the network system.

Definition 6. $\mathscr{F}$ is *monotonic* w.r.t. $y^{\max}$ if given $y^{\max,1} \ge y^{\max,2} \ge 0$, $\mathscr{F}(y^{\max,2}) \subseteq \mathscr{F}(y^{\max,1})$. The inequality is defined element-wise.

Lemma 1. *There always exists a* $\mathscr{F}$ *that is monotonic w.r.t.* $y^{\max}$*.*

Proof. $y^{\max,1} \ge y^{\max,2}$ implies that the uncertainty set for $\mathscr{F}(y^{\max,1})$ is a superset of the uncertainty set for $\mathscr{F}(y^{\max,2})$, so $\mathscr{F}(y^{\max,1})$ is also robust control invariant under $|y| \le y^{\max,2}$. Therefore, picking $\mathscr{F}(y^{\max,2}) = \mathscr{F}(y^{\max,1})$ completes the proof. □

The lemma above is intuitive since the size of the RCI should monotonically grow with the size of the disturbance bound.

Assumption 1. The RCI computation procedure $\mathscr{F}$ considered in this paper is monotonic.

Note that Assumption 1 can be made without loss of generality due to lemma 1.

Given a procedure $\mathscr{F}$ that computes RCIs for subsystems given $y^{\max}$ as described above, define the local mapping λ_i:

$$\begin{aligned} \lambda_i(y^{\max}_{\mathcal{N}_i}) &= \max_{x_i \in \mathscr{F}_i(y^{\max}_{\mathcal{N}_i})} \left| h_i(x_i) \right|, \\ \Lambda(y^{\max}) &= [\lambda_1(y^{\max}_{\mathcal{N}_1}); \lambda_2(y^{\max}_{\mathcal{N}_2}); \ldots; \lambda_N(y^{\max}_{\mathcal{N}_N})]. \end{aligned} \tag{24}$$

Note that $\Lambda(y^{\max})$ has the same dimension as $y^{\max}$. Then we have our main theorem.

Theorem 2 (Set invariance of a network system with assume-guarantee contract)**.** *Given an RCI computation procedure $\mathscr{F}$ and let Λ be defined in* (24)*. If there exists a $y^{\max} \in \mathbb{R}^N_{\geq 0}$ such that*

$$\Lambda(y^{\max}) \leq y^{\max}, \tag{25}$$

then $\mathscr{F}(y^{\max})$ is a robust control invariant set for the network system.

Proof. Let $\mathcal{S}_i = \mathscr{F}_i(y^{\max}_{\mathcal{N}_i})$, and define a network assume-guarantee contract with

$$\begin{aligned}\phi^i_{ae} = (x_i(0) \in \mathcal{S}_i) \wedge \Box\,(d_i \in \mathcal{D}_i)\\ \wedge\Box\,(u_i = k_i(x_i, y_{\mathcal{N}_i}, d_i))\,,\end{aligned} \tag{26}$$

$$\phi^i_{af} = \varphi^i_{af}(T) = \Box_{[0,T]}\,|y_{\mathcal{N}_i}| \leq y^{\max}_{\mathcal{N}_i}, \tag{27}$$

$$\phi^i_g = \varphi^i_g(\hat{T}) = \Box_{[0,\hat{T}]} x_i \in \mathcal{S}_i; \tag{28}$$

where k_i is the feedback law that keeps x_i within $\mathcal{S}_i$. By the definition of an RCI, the existence of k_i is guaranteed. Let $\hat{\Lambda}(T) = T + T_s$, $\Gamma(\hat{T}) = \hat{T}$, where T_s is the time step of the discrete dynamics in (2).

Among the 4 assumptions of Theorem 1, Assumption 1 is satisfied by the definition of an RCI. With (25), Assumption 2 is satisfied with Γ defined above. Assumption 3 is satisfied by (26) and Assumption 4 is satisfied by setting $T = 0$ in (27). Then, by Theorem 1, the guarantee for the network system is

$$\hat{\phi}^i_g = \bigwedge_{k=0}^{\infty} \Box_{[0,k\cdot T_s]} x_i \in \mathcal{S}_i, \tag{29}$$

which is simplified to

$$\forall i = 1, ..., N, \Box_{[0,\infty)} x_i \in \mathcal{S}_i. \tag{30}$$

□

The condition in (25) is the critical condition to show invariance, from hereon we refer to it as the "*validity condition*". It can be interpreted as the condition such that each node can satisfy what other nodes assume of it. In the next section we will describe an algorithm that searches for a $y^{\max}$ that satisfies the validity condition, or else returns an infeasibility certificate.

V. Search for Assume-Guarantee Contract with Epigraph Method

In this section, we present the epigraph algorithm that searches for an assume-guarantee contract that meets the validity condition if one exists. In particular, we show that the epigraph algorithm can be viewed as an extension of the classic small gain theorem to network systems with nonlinear "gains" and multiple interconnected systems.

A. Epigraph representation of the validity condition

Recall that given a function $g : \mathbb{R}^m \to \mathbb{R}$, the epigraph of g is defined as

$$\mathbf{epi}(g) := \{(x, t) \mid x \in \mathbf{dom}\, g, \quad g(x) \leq t\},$$

where $\mathbf{dom}\, g$ denotes the domain of g.

The idea behind our algorithm is to look at each local $\lambda_i : \mathbb{R}^{N_i}_{\geq 0} \to \mathbb{R}_{\geq 0}$ as a function and consider its epigraph. The condition in (25) is equivalent to the following condition:

$$[y^{\max}_{\mathcal{N}_i}; y^{\max}_i] \in \mathbf{epi}(\lambda_i).$$

Suppose the epigraph of each λ_i is known, the search for a valid contract can be formulated as the following optimization:

$$\begin{aligned}\min_{y^{\max}\geq \mathbf{0}} \; &\sum_{i=1}^{N} y^{\max}_i\\ \text{s.t. } \forall i = 1, ..., N, &\left[y^{\max}_{\mathcal{N}_i}; y^{\max}_i\right] \in \mathbf{epi}(\lambda_i).\end{aligned} \tag{31}$$

If $\mathbf{epi}(\lambda_i)$ does not have a simple explicit form, one can replace $\mathbf{epi}(\lambda_i)$ in (31) with a tractable inner approximation and the optimization would still generate a valid contract if a solution is obtained.

Example 1. Consider the two systems Σ_1 and Σ_2 interconnected as shown in Fig. 3.

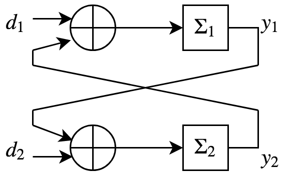

Fig. 3: Two systems interconnection network

Suppose that there exist constants $\mu_1, \mu_2, \nu_1, \nu_2 \geq 0$ such that

$$\begin{aligned}\|y_1\|_\infty \leq \mu_1\|d_1\|_\infty + \nu_1\|y_2\|_\infty,\\ \|y_2\|_\infty \leq \mu_2\|d_2\|_\infty + \nu_2\|y_1\|_\infty.\end{aligned} \tag{32}$$

If, in addition, the small gain condition is satisfied, i.e.,

$$\nu_1 \cdot \nu_2 < 1,$$

then the small gain theorem tells us that the interconnection is stable and

$$\begin{aligned}\|y_1\|_\infty \leq \quad & \frac{\mu_1}{1-\nu_1\nu_2}\|d_1\|_\infty + \frac{\mu_2\nu_1}{1-\nu_1\nu_2}\|d_2\|_\infty,\\ \|y_2\|_\infty \leq \quad & \frac{\mu_1\nu_2}{1-\nu_1\nu_2}\|d_1\|_\infty + \frac{\mu_2}{1-\nu_1\nu_2}\|d_2\|_\infty.\end{aligned} \tag{33}$$

The proof can be found in [38]. The same result can be obtained by considering the epigraph.

Proposition 1. *Given* (32) *and bounded $\|d_i\|_\infty > 0, i = 1, 2$, there exists an assume-guarantee contract that guarantees* (33) *if $\nu_1 \cdot \nu_2 < 1$.*

Proof. Given (32), $\|d_1\|_\infty$ and $\|d_2\|_\infty$, the mappings λ_1, λ_2 can be easily found to be

$$\begin{aligned}\lambda_1(\|y_2\|_\infty) = \quad & \mu_1\,\|d_1\|_\infty + \nu_1\,\|y_2\|_\infty\\ \lambda_2(\|y_1\|_\infty) = \quad & \mu_2\,\|d_2\|_\infty + \nu_2\,\|y_1\|_\infty\,.\end{aligned} \tag{34}$$

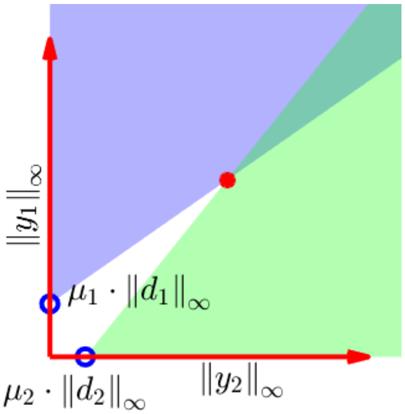

Fig. 4: Epigraph of $\lambda_{1,2}$ for the interconnected system from Example 1

The epigraphs of λ_1 and λ_2 are shown in Fig. 4, where the blue region shows $\mathbf{epi}(\lambda_1)$ and the green region shows $\mathbf{epi}(\lambda_2)$. A contract is valid if the point $[\|y_1\|_\infty, \|y_2\|_\infty]^\intercal$ lies within the intersection of the two epigraphs. When $\|d_1\|_\infty$ and $\|d_2\|_\infty$ are not both zero, the two epigraphs have a nonempty intersection if and only if $\nu_1 \cdot \nu_2 < 1$. When the intersection is nonempty, the contract with the minimum $\|y_1\|_\infty$ and $\|y_2\|_\infty$ is depicted as the red dot, which equals to the result in (33). □

Remark 6. The small gain theorem is a special case of the epigraph interpretation. In cases when λ_i are nonlinear functions and when there are more than 2 interconnected subsystems, the epigraph method is still applicable.

B. Grid Sampling for epigraph approximation

Next, we show a grid sampling approach to compute an inner-approximation of $\mathbf{epi}(\lambda_i)$. For the simplicity of notation, we consider a scalar function $g : \mathbb{R}^n \to \mathbb{R}$, with input x and output $y = g(x)$.

The epigraph of a function is not bounded since it is defined as the area above the function graph in $[x; g(x)]$ space, as shown in Fig. 5. In addition, the domain of x may be unbounded as well.

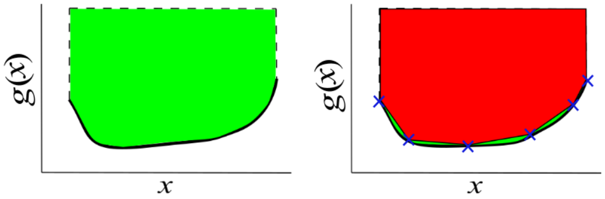

Fig. 5: Epigraph of a function and its polytopic approximation

Therefore, to obtain a tractable representation of the epigraph, we first need to fix the domain of x to be a compact set $\mathcal{X}$ of interest. Notice that by definition, when g is a convex function, its epigraph is a convex set. If one picks a finite set $S = \{x_1, x_2, ..., x_n\}$ and evaluates the function at every point in S, the convex hull of the point set $[x_1; g(x_1)], [x_2; g(x_2)], ..., [x_n; g(x_n)]$ can then be computed, denoted as H, where H is convex and $H \subseteq \mathbf{epi}(g)$, as shown in the second figure in Fig. 5. Therefore, for a convex function, we can simply sample the input and use the convex hull of the sampled points with their function values as the approximation of $\mathbf{epi}(g)$. Computing the convex hull of an n-point set of dimension d can be done in $O(n \log n + n^b)$ time, where $b = \lfloor d/2 \rfloor$ [39].

When g is not convex, a decomposition algorithm is developed to inner approximate $\mathbf{epi}(g)$ with a union of polytopes. We omit the details, and instead provide a sketch of the algorithm: Suppose $\mathbf{epi}(g)$ is approximated by $\bigcup_{j=1}^{M} P_j$, where P_j are polytopes, then $[x; g(x)] \in \mathbf{epi}(g)$ is encoded with the following mixed integer constraint:

$$[x; g(x)] \in \bigcup_{j=1}^{M} P_j \Leftrightarrow \left(\begin{array}{l} \mathbb{1}([x; f(x)] \in P_j) - s_j \geq 0, \\ s_j \in \{0, 1\}, \sum_{j=1}^{M} s_j = 1, \end{array} \right) \tag{35}$$

where s_j are the binary variables and $\mathbb{1}(\cdot)$ is the indicator function.

Definition 7. Two or more disturbance signals are *summable* if they have the same input dynamics. To be specific, consider $x^+ = f(x, u, d)$, where $d = [d_1, ..., d_l]^\intercal \in \mathbb{R}^l$ is the disturbance. The individual disturbances $\{d_i\}$ are summable if $\exists \bar{f}$ such that $f(x, u, d) \equiv \bar{f}(x, u, \sum_i d_i)$.

Summable disturbance inputs can be combined and viewed as one disturbance since they invoke the same disturbance dynamics and their bounds are summable, i.e.,

$$(|d_1| \leq \alpha) \wedge (|d_2| \leq \beta) \Rightarrow |d_1 + d_2| \leq \alpha + \beta.$$

Since the number of samples needed grow exponentially with the number of disturbance inputs for each node, combining summable disturbance inputs reduces the complexity of the epigraph algorithm.

Algorithm 2 Epigraph algorithm for valid assume-guarantee contracts

```
procedure EPIGRAPH_SEARCH($\{\lambda_i\}_{1:N}$)
    for do $i = 1 : N$
        Calculate inner approximations of $\mathbf{epi}(\lambda_i)$
    end for
    Solve (31) with $\mathbf{epi}(\lambda_i)$ for $y^{\max}$
    return $y^{\max}$ if (31) is feasible, otherwise return infeasible.
end procedure
```

VI. Robust Control Invariant Set for Power Grid

In this section, we apply the proposed method to the fixed point tracking control of a power network and present the simulation result. We use the IEEE 9-bus case introduced in Section II-B, see Fig. 2 for the network topology.

A. Search for valid assume-guarantee contract with epigraph

We use the power grid dynamics given in Section II-B. Since the goal is fixed point tracking, we use the linearized dynamics presented in (6), and include the linearization error in the disturbance term. The assume-guarantee contract in this example follows the form introduced in Section III-D. Each bus takes the bound on the phase angle deviation of its neighbors as the assumption, and guarantees that its own phase angle deviation stays bounded. The contract parameters are the bounds on phase angle deviation for each bus $\theta^{\max}$.

The computation of the RCI follows the robust linear programming algorithm reviewed in Section III-A, c.f. Algorithm 1. For each bus, the RCI is computed with the linearized model in (7) after discretization. The inputs to the RCI computation of the i^{th} bus are the input sets $\mathcal{U}_i$, exogenous disturbance bounds $\mathcal{D}_i$, and bounds on the phase angle deviations of neighboring buses $\theta^{\max}_{\mathcal{N}_i}$, where $\mathcal{U}_i$ and $\mathcal{D}_i$ are determined by the environment assumptions ϕ^i_{ae} and are assumed to be given, and $\theta^{\max}_{\mathcal{N}_i}$ is given as the feedback assumption ϕ^i_{af}.

As discussed before, we assume that $\mathcal{U}_i$, $\mathcal{D}_i$, and the dynamics Σ_i for each bus is given. Let $\mathscr{F}$ be the RCI computation procedure, and define

$$\lambda_i(\theta^{\max}_{\mathcal{N}_i}) = \max_{x_i \in \mathscr{F}(\theta^{\max}_{\mathcal{N}_i})} |\theta_i|. \tag{36}$$

In the IEEE 9 bus example (as shown in Fig. 2), bus 1,2, and 3 are generator buses, and the rest are pure load buses. Here we add an additional constraint to $\mathscr{F}$ such that for each RCI $\mathcal{S}_i$ computed for the generator buses,

$$\max_{x_i \in \mathcal{S}_i} |\omega_i| \leq \omega^{\max},$$

so that $x_i \in \mathcal{S}_i$ implies that the safety constraint is satisfied.

By Assumption 1, λ_i is clearly monotonic. The evaluation of λ_i is done in two steps. First, with $\theta^{\max}_{\mathcal{N}_i}$ fixed, $\mathscr{F}$ is called to compute an RCI $\mathcal{S}_i$, then $\theta^{\max}_i$ is obtained through (36). Next, the inner approximation of $\mathbf{epi}(\lambda_i)$ is computed for each bus with the grid sampling algorithm, Fig. 6 shows two computed epigraph as examples:

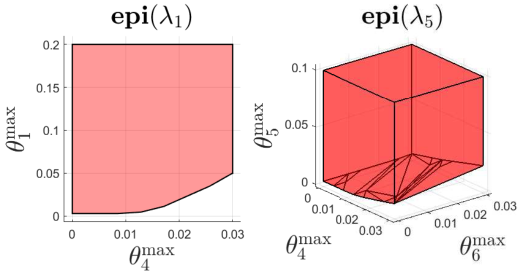

Fig. 6: Inner approximations of $\mathbf{epi}(\lambda_1)$ and $\mathbf{epi}(\lambda_5)$

As shown in Fig. 2, bus 1 has one neighbor (bus 4) and bus 5 has two neighbors (bus 4 and 6), therefore $\mathbf{epi}(\lambda_1)$ is in 2d while $\mathbf{epi}(\lambda_5)$ is in 3d. Since some of the epigraphs are not convex, a mixed integer programming as formulated in (35) is solved. Once a $\theta^{\max}$ that satisfies the validity condition is found, it leads to a valid network assume-guarantee contract, and an RCI can be obtained via $\mathscr{F}$.

Remark 7. If one takes P_{ij}, the power flow from between bus i and bus j as the disturbance inputs, they are summable (has the same input dynamics). Moreover, it is easy to see that combining P_{ij} into one disturbance input is lossless since P_{ij} are scalars. We use Fig. 6 to conceptually show the scenario with multiple disturbance inputs, the actual computation of the epigraphs for the power grid network can be simplified by combining the power flow from neighboring buses for each bus.

Fig. 7 shows the robust invariant sets for the generator buses under the assume-guarantee contract.

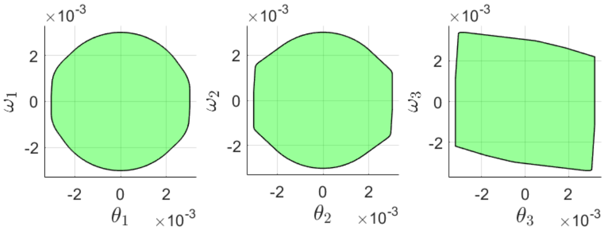

Fig. 7: Robust control invariant sets for the generator buses

B. Simulation result

For each bus, the computed robust control invariant set is then enforced with a control barrier function (CBF), as reviewed in Section III-B. In this example the primal-dual controller introduced in [22] is used as the legacy controller u^0.

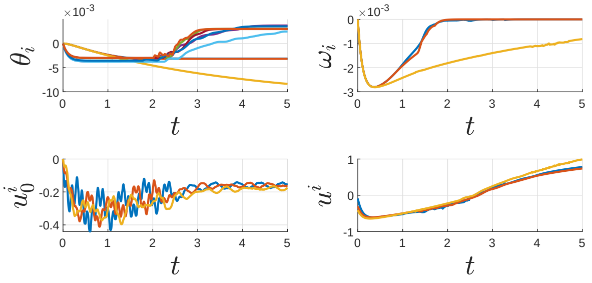

Fig. 8: Simulation with CBF as supervisor

In Fig 8 we show a simulation trace of the 9-bus system with the CBF controller as the supervisory controller, implementing (14). The bound on frequency deviation is set at $\omega^{\max} = 5 \times 10^{-3} rad/s$ and it is never breached.

Fig. 9 shows the phase angles with and without the CBF supervisor. Under the CBF supervisory controller (magenta plots), all phase anngles are within their respective bound determined by the contract; on the other hand, without CBF control (blue plots), there is no guarantee that the phase angles stay within bounds under u^0.

VII. Model Predictive Control for contingency recovery

We have shown how to compute an RCI for the network system with an assume-guarantee contract, which is sufficient to guarantee the satisfaction of the safety constraint if the network operates around a fixed operating point θ^0 (around which the dynamics are linearized). However, when a severe contingency occurs such as a change in the network topology, a

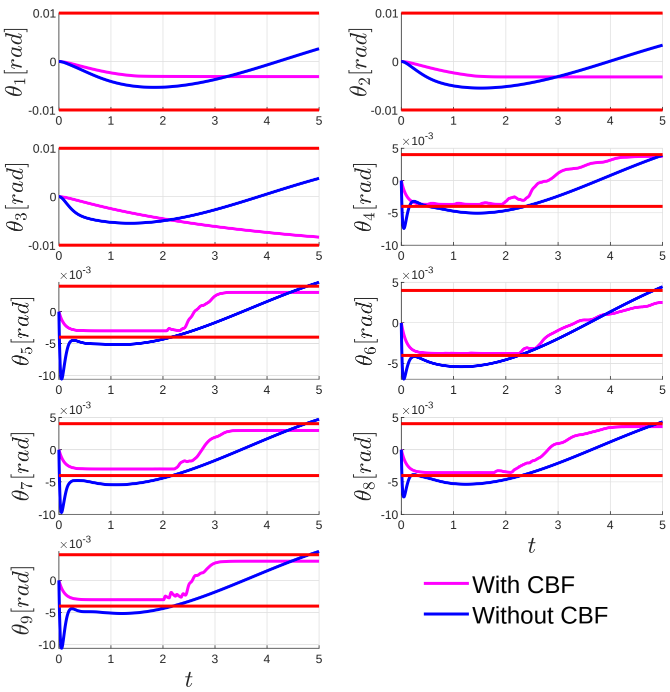

Fig. 9: phase angle plot with and without the CBF supervisor

bus disconnects, or a line shorted, the RCI around the original operating point can no longer be maintained with the available control input, and the operating point has to change. This calls for an alternative controller that deals with the transient.

In this section we propose a contingency tube model predictive control scheme that can handle the transient caused by contingency cases based on the mechanism developed for fixed point control.

A. Model Predictive Control for Reference Trajectory Generation

The proposed MPC scheme is slightly different from the classic MPC (see for example [40]), here we briefly review some concepts from the MPC literature and introduce our contingency tube MPC scheme.

There are two important horizons for MPC, the prediction horizon T_p and the control horizon T_c. An MPC controller looks ahead T_p steps and represents the future state trajectory as a function of the input sequence, then solves for the optimal control sequence w.r.t. a cost function and some state and input constraints. The control sequence will be executed for T_c steps, at which point another MPC iteration is executed, and a new control law computed. Traditional MPC schemes typically have $T_c \ll T_p$, often choosing $T_c = 1$, which requires the controller to have access to the state information without delay.

In the network setting, distributed MPC schemes have been proposed [41], [42] that depends on fast communication and distributed optimization techniques. However, when the communication delay is not negligible, the receding horizon scheme is likely to be infeasible. Instead, we consider a contingency tube MPC scheme that is triggered only when a contingency occurs, and we do not update the control policy until the end of the prediction horizon or another contingency occurs. Obviously, such an MPC scheme is equivalent to feedforward control once the MPC input is solved, and would not work without feedback. We use CBF at each node of the network as the feedback controller to guarantee the tracking performance of the reference trajectory generated by the MPC. The contingency tube MPC is designed to guarantee the safe transition of the network to the new operating point after the contingency.

Three requirements for the MPC should be considered:

- Computation of the MPC solution should be fast enough to allow real-time implementation.
- Safety constraints should be satisfied.
- Communication limitations (constraints) should be respected.

Computation limitations differ with applications. In the power grid example shown in VII-C, in order to speed up the computation, we use the linearized model shown in (6) and the nonlinearity is treated as bounded disturbance. With a linear discrete-time model, quadratic costs and linear state and input constraints, the MPC can be solved by convex quadratic programming over the input sequence $\hat{u}(0 : T_p - 1)$ with T_p being the prediction horizon. The MPC controller is triggered when any bus detects a contingency that exceeds the capability of the fixed point controller, such as connecting or disconnecting a bus or a loss of line. To get the reference trajectory, the following optimization is solved.

$$\begin{aligned} &\min\ \mathcal{J}(\hat{u}, \hat{x}, x^\star) \\ &\quad s.t. \hat{x}(t+1) = \hat{f}(\hat{x}(t), \hat{u}(t), \hat{d}(t)), \\ &\qquad \forall i \in \mathcal{G}, t = 0, 1, ..., T_p - 1, |\omega_i| \le \omega^{\max,ff}, \\ &\qquad \mathcal{C}\left(\hat{u}(0 : T_p - 1)\right) = 0, \end{aligned} \tag{37}$$

where $\mathcal{J}$ is the cost function, which penalizes $\hat{u}$ and the distance between $\hat{x}$ and $x^\star$, the new operating point under the contingency. $\omega^{\max,ff}$ is the bound on the bus frequencies for the MPC. Later we show that with CBF, the frequency tracking error is bounded by $\omega^{\max,fb}$. Let $\omega^{\max} = \omega^{\max,ff} + \omega^{\max,fb}$, then the total frequency deviation is bounded by $\omega^{\max}$. $\hat{x}$ and $\hat{u}$ are the reference state and input trajectories and $\hat{d}$ is the predicted disturbance sequence, which depends on the knowledge of the contingency. $\hat{f}$ is the nominal dynamic model (linearized model) with the following form:

$$\hat{f}(x, u, d) = \begin{bmatrix} \hat{f}_1(x_1, y_{\mathcal{N}_1}, u_1, d_1) \\ \vdots \\ \hat{f}_N(x_N, y_{\mathcal{N}_N}, u_N, d_N) \end{bmatrix}, \tag{38}$$

each $\hat{f}_i$ is the approximation of the dynamics in (4). In the power grid case, the linear model is represented in (6). $\mathcal{C}$ is the constraint on the input caused by communication delay, which will be discussed later. The proposed scheme is based on the assumption that the network is close to a steady state when the contingency happens, therefore we can compute the reference trajectory for the whole network assuming that the system is at steady state without real-time state information. Once the MPC obtains a solution, the solution is sent to each node as the reference trajectory for the whole prediction horizon.

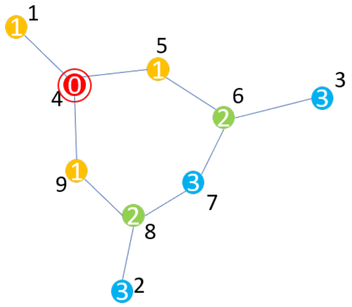

Fig. 10: Contingency positions. Colors refer to delay constraints: yellow (1-step delay), green (2-step delay), blue (3-step).

Each node then uses a local feedback controller to track the reference trajectory.

Since the transmission of the reference trajectory is also subject to communication delay, we need the additional constraint $\mathcal{C}$ on the input. Take the 9 bus test case as an example, suppose a contingency is detected at bus 4 and the MPC is computed at node 4. Assuming that the signal travels one edge per time-step, then the delay at each node is shown in Fig. 10, and $\mathcal{C}$ would enforce the following input structure:

$$\begin{bmatrix} \hat{u}_1(0:T_p-1) \\ \hat{u}_2(0:T_p-1) \\ \hat{u}_3(0:T_p-1) \\ \hat{u}_4(0:T_p-1) \\ \hat{u}_5(0:T_p-1) \\ \hat{u}_6(0:T_p-1) \\ \hat{u}_7(0:T_p-1) \\ \hat{u}_8(0:T_p-1) \\ \hat{u}_9(0:T_p-1) \end{bmatrix} = \begin{bmatrix} 0 & * & * & * & * \\ 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \\ 0 & 0 & * & * & * \\ 0 & * & * & * & * \end{bmatrix}, \tag{39}$$

which restricts the input to be zero before the reference trajectory signal arrives. Note that we assume the communication delay is the same across all lines and that the shortest path is always chosen.

B. Contingency tube MPC with CBF

As mentioned above, a local feedback controller is needed to track the reference trajectory generated by the MPC. The idea of centralized tube MPC was discussed in [43], [44], and was extended to distributed tube MPC for multiple subsystems without coupling in the dynamics [45]. There exist, however, strong coupling between nodes in the models of the grid dynamics that we consider. We use the assume-guarantee contract method proposed previously to handle the trajectory tracking problem for networks with strong coupling.

We assume that there exists a nominal dynamic model $\hat{f}_i$ for each subsystem in the network, and the difference between the model and the actual dynamics as described by(2) is bounded:

$$f_i\left(x_i, y_{\mathcal{N}_i}, u_i, d_i\right) - \hat{f}_i\left(x_i, y_{\mathcal{N}_i}, u_i, d_i\right) \in \mathcal{W}_{f_i}, \tag{40}$$

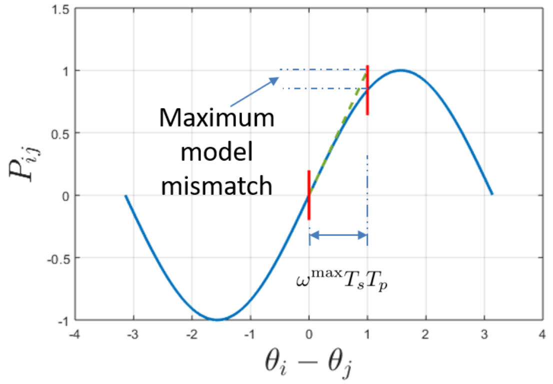

Fig. 11: Linearization error

where $\mathcal{W}_{f_i}$ is the bound for model mismatch. The goal is to track a reference trajectory $\hat{x}(1:T_p)$ that satisfies

$$\begin{aligned} \hat{x}_i(t+1) &= \hat{f}_i\left(\hat{x}_i(t), \hat{y}_{\mathcal{N}_i}(t), \hat{u}_i(t), \hat{d}_i\right), \\ \hat{y}_i(t) &= h(\hat{x}_i(t)), i = 1, ..., N, t = 0, ..., T_p - 1 \end{aligned} \tag{41}$$

and keep the tracking error bounded.

For the grid dynamics, $\hat{f}_i$ is linear, therefore we can write it as

$$\hat{f}_i(x_i, y_{\mathcal{N}_i}, u_i, d_i) = A_i x_i + B_i u_i + E_i^1 y_{\mathcal{N}_i} + E_i^2 d_i,$$

where (A_i, B_i, E_i^1, E_i^2) are easily obtained from (6). Define the error $e_i = x_i - \hat{x}_i$, then the error evolves as

$$e_i^+ = A_i e_i + B_i \Delta u_i + E_i^1 (y_{\mathcal{N}_i} - \hat{y}_{\mathcal{N}_i}) + E_i^2 (d_i - \hat{d}_i) + \Delta f_i(t) \tag{42}$$

where $\Delta u \doteq u_i - \hat{u}_i$ denotes the feedback part of the input,

$$\Delta f_i(t) \doteq f_i\left(\hat{x}_i(t), \hat{y}_{\mathcal{N}_i}(t), \hat{u}_i(t), \hat{d}_i\right) - \hat{f}_i\left(\hat{x}_i(t), \hat{y}_{\mathcal{N}_i}(t), \hat{u}_i(t), \hat{d}_i\right),$$

is the modelling error, which satisfies $\forall t = 0, 1, ..., T_p - 1, \Delta f_i(t) \in \mathcal{W}_{f_i}$, and is considered as an additional disturbance to the system.

The modeling error caused by linearization can be clearly seen by looking at (5) and (6). Since the reference trajectory has a finite duration $T_s T_p$ and ω_i is bounded by $\omega^{\max}$ for every bus, the bound on the modelling error can be obtained, as shown in Fig. 11. We can now state the main result of this section.

To this point, the trajectory tracking problem can be handled by the machinery developed for fixed point tracking in Section IV.

Theorem 3. *Consider the power system dynamics in* (5), *denoted as f, and the linearized model in* (6), *denoted as $\hat{f}$. For a reference state and input trajectory $[\hat{x}(1:T_p), \hat{u}(0:T_p-1)]$ that satisfies* (41), *suppose there exists a feedback controller $\Delta u_i = k_i(x_i - \hat{x}_i, y_{\mathcal{N}_i} - \hat{y}_{\mathcal{N}_i}, d_i - \hat{d}_i)$ for each bus such that for a given bound $\Delta\mathcal{D}_i$ of $d_i - \hat{d}_i$, a given set $\mathcal{S}_i \subseteq \mathcal{X}_i$ for each* *bus and a given bound on $|y - \hat{y}| \leq \Delta y^{\max}$, the following is true:*

$$\begin{gathered}\forall x_i(t) \in \hat{x}_i(t) + \mathcal{S}_i, d_i(t) \in \hat{d}_i(t) + \Delta\mathcal{D}_i \\ \forall \left|y_{\mathcal{N}_i}(t) - \hat{y}_{\mathcal{N}_i}(t)\right| \leq \Delta y^{\max}_{\mathcal{N}_i}, \\ x_i(t+1) = f_i\left(x_i, y_{\mathcal{N}_i}, u_i, d_i\right) \in \hat{x}_i(t+1) + \mathcal{S}_i \\ \max_{x_i(t) \in \hat{x}_i(t) + \mathcal{S}_i} \left|h_i(x_i) - \hat{y}_i\right| \leq \Delta y^{\max}_i, t = 0, ..., T_p - 1,\end{gathered}$$

where $\Delta y^{\max}_{\mathcal{N}_i}$ is a projection of $\Delta y^{\max}$ onto $\mathcal{Y}_{\mathcal{N}_i}$ and the "+" signs between vectors and sets denote direct sums. Then let $u_i = \hat{u}_i + k_i(x_i - \hat{x}_i, y_{\mathcal{N}_i} - \hat{y}_{\mathcal{N}_i}, d_i - \hat{d}_i)$, for any $x(0)$ satisfying $x_i - \hat{x}_i \in \mathcal{S}_i$, disturbance satisfying $d_i(t) \in \hat{d}_i(t) + \Delta\mathcal{D}_i$, the closed loop trajectory stays inside the tube defined as $\{x(1 : T_p) \mid x(t) \in \hat{x}(t) + \mathcal{S}_1 \times ... \times \mathcal{S}_N\}$.

Proof. The proof can be obtained by directly applying Theorem 2 on the error dynamics in (42). □

To implement the contingency tube MPC, we first compute an RCI for the error dynamics taking the bound on disturbance and model mismatch into account. Once a contingency occurs, the MPC scheme in (37) is solved to obtain a reference trajectory $\hat{x}$, then at each node, the following CBF supervisory control is implemented:

$$\begin{aligned} u^i(t) = &\arg\min_{u \in \mathcal{U}_i} \left\|u - u^0_i(t)\right\|^2 \\ &\text{s.t.} \quad \dot{b}_i(x_i - \hat{x}_i, u) + \kappa b_i(x_i - \hat{x}_i) \geq 0, \end{aligned} \tag{43}$$

where b_i is the CBF for the i^{th} node defined based on the RCI $\mathcal{S}_i$, u^0_i is the nominal control signal for the i^{th} node, which can be simply chosen as $\hat{u}_i$, or alternatively chosen as $\hat{u}_i$ plus a feedback part. In Section VII-C, u^0_i is picked as $\hat{u}_i$ plus an LQR feedback component.

C. Simulation of MPC for contingency

To validate the proposed contingency tube MPC scheme, we use the high-fidelity power grid simulator PST [19] as the simulation environment. PST allows several types of contingency cases, such as the 3-phase error, loss of line and loss of load. The New England network from PST is picked for demonstration, which contains 39 buses with 10 of them generator buses, as shown in Fig. 12. The red nodes are the generator buses and the green nodes are the pure load buses. The two tested contingencies are:

- **Case 1:** Bus loss at bus 7
- **Case 2:** Line between bus 3 and 4 trips

The locations of the failures are shown in Fig. 12. When bus 7 disconnects, the network is able to find a new set point without changing the generation. When the line between bus 3 and 4 disconnects, since the line is located in the center of the network and causes a significant change to the network topology, the network cannot balance itself with the original generation. So an optimal power flow (OPF) routine (AC OPF routine in Matpower toolbox [46]) is performed to get the new generation together with the new operating point, and the contingency tube MPC is used to complete the transition to the new operating point. The sampling time and horizon for the contingency tube MPC is set at 50ms and 2.5s ($T_p = 50$).

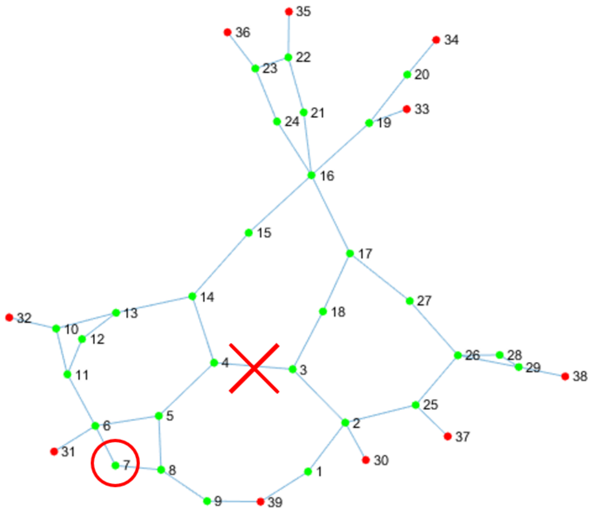

Fig. 12: New England grid network structure

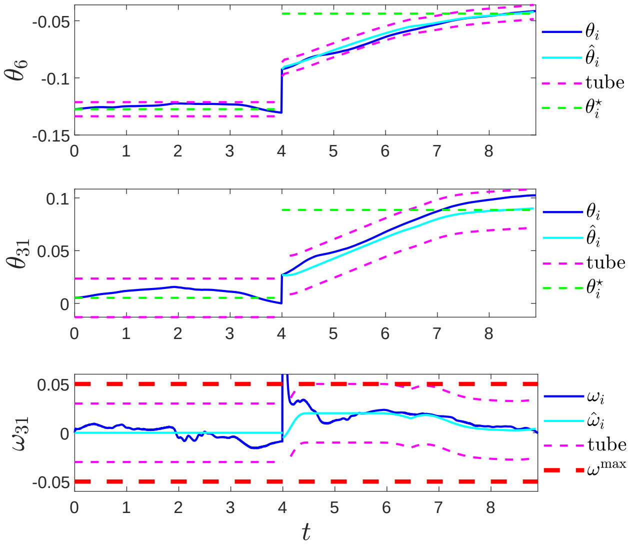

Fig. 13: Case 1: Bus failure contingency

We insert sinusoidal load fluctuation with the maximum magnitude allowed by the RCI at every bus to simulate the effect of uncontrolled load disturbance. Once the contingencies (bus loss in case 1 and line loss in case 2) are detected, the contingency tube MPC kicks in at the nearest node to the contingency (bus 6 in case 1 and bus 4 in case 2) to compute the nominal trajectory for the transition to the new operating points. Then the plan is sent out to the rest of the network via communication. In both cases, the signal is assumed to travel two edges per sampling interval.

Fig. 13 shows the PST simulation of case 1. The blue line is the state, the magenta line represents the tube (i.e. the region the state is confined to lie in), the green line represents the new set point for the phase angle and the red line represents the bound for frequency. We show the state trajectory of bus 6, the bus closest to the contingency, and bus 31, the closest generator bus to the contingency. When the contingency happens, the frequency breached the constraint

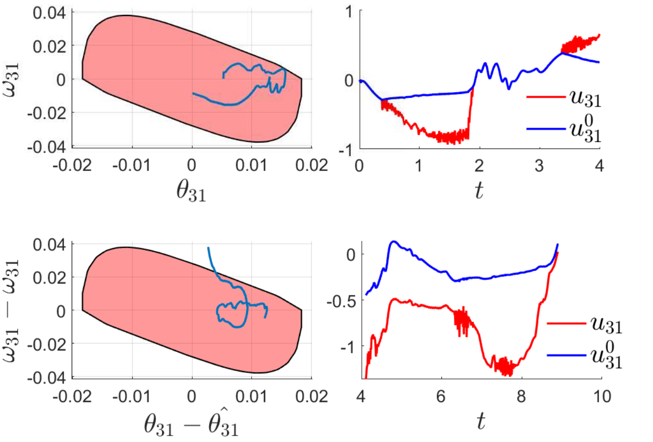

Fig. 14: Robust Control Invariant set with state and input trajectories at bus 31

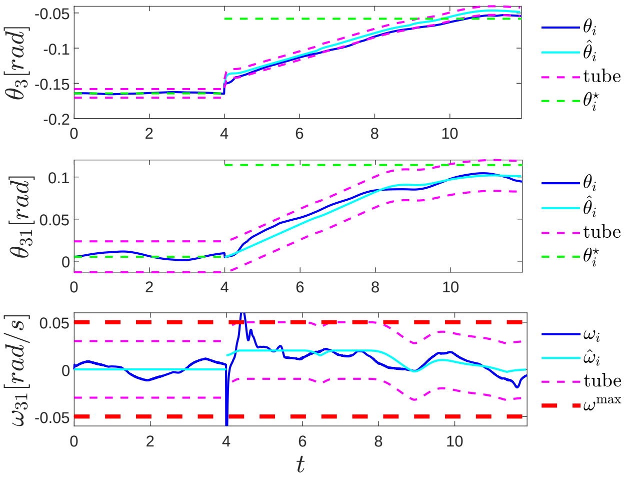

Fig. 15: Case 2: Line failure contingency

for a slight moment, the reason for this violation of the safety constraint are (i) the dynamics under the contingency is not modelled accurately (ii) the violation happened instantly after the loss of bus 7, before the contingency tube MPC is able to kick in and react (iii) there may be some simplification made by the PST toolbox which ignores some inertia in the system. Once the contingency tube MPC scheme kicks in, the state trajectory was kept within the tube and the network eventually reaches the new operating point without violating the safety constraint. Moreover, Fig. 14 shows the state trajectory w.r.t. the RCI and the inputs to the system at bus 31 (generator bus). The two figures on top shows the state and input trajectories before the contingency at $t = 4s$. Due to the sinusoidal fluctuation of the load, the phase angle also fluctuates, but it never left the RCI, as shown in Fig. 14(a). In Fig. 14(b), the blue curve shows the legacy controller input, and the red curve shows the CBF controller input. The timing of the interventions coincide with the timing when the state is close to the boundary of the RCI. Fig. 14(c) and (d) show the state and input trajectories after the contingency. Note that in the contingency tube MPC scheme, we require the error state $x - \hat{x}$ instead of the state x to stay inside the RCI. After the temporary deviation right after the contingency, $x - \hat{x}$ stays inside the RCI due to the CBF controller.

Fig. 15 shows the simulation for the line loss case. Similarly, the contingency tube MPC together with the CBF controller is able to keep the system trajectory within the tube and take the whole network to the new operating point.

VIII. Conclusion

We consider the application of robust control invariant set and control barrier functions on network systems to prevent large deviations from the desired working condition. The key idea is to use assume-guarantee contracts to break the large network into small subsystems. The coupling between subsystems are treated as bounded disturbances and is handled with a network assume-guarantee contract. We show that a network assume-guarantee contract satisfying the validity condition guarantees robust set invariance for the whole network system. Furthermore, we propose an epigraph algorithm that is capable of searching for a valid contract, which enjoys linear complexity when the network is sparse or the coupling terms are summable. Based on the network assume-guarantee contract idea, we further propose a contingency tube MPC scheme that is capable of handling contingencies with changing operating points while respecting communication limitations. The proposed method is demonstrated with two power grid control examples and tested with high-fidelity simulations. The results validate the proposed methods and show their capability to prevent large deviations from the operating point and handle severe contingencies.

References

[1] P. Nilsson, O. Hussien, Y. Chen, A. Balkan, M. Rungger, A. Ames, J. Grizzle, N. Ozay, H. Peng, and P. Tabuada, "Preliminary results on correct-by-construction control software synthesis for adaptive cruise control," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*. IEEE, 2014, pp. 816–823.

[2] Y. Chen, H. Peng, and J. W. Grizzle, "Validating noncooperative control designs through a lyapunov approach," *IEEE Transactions on Control Systems Technology*, no. 99, pp. 1–13, 2018.

[3] Y. Chen, H. Peng, and J. Grizzle, "Obstacle avoidance for low-speed autonomous vehicles with barrier function," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 1, pp. 194–206, 2018.

[4] F. Blanchini, "Set invariance in control," *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999.

[5] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games," *IEEE Transactions on automatic control*, vol. 50, no. 7, pp. 947–957, 2005.

[6] M. V. Khlebnikov, B. T. Polyak, and V. M. Kuntsevich, "Optimization of linear systems subject to bounded exogenous disturbances: The invariant ellipsoid technique," *Automation and Remote Control*, vol. 72, no. 11, pp. 2227–2275, 2011.

[7] J. Anderson and A. Papachristodoulou, "Advances in computational lyapunov analysis using sum-of-squares programming," *Discrete & Continuous Dynamical Systems-Series B*, vol. 20, no. 8, 2015.

[8] S. Prajna, P. A. Parrilo, and A. Rantzer, "Nonlinear control synthesis by convex optimization," *IEEE Transactions on Automatic Control*, vol. 49, no. 2, pp. 310–314, 2004.

[9] J. Anderson and A. Papachristodoulou, "A decomposition technique for nonlinear dynamical system analysis," *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1516–1521, 2011.

[10] O. Hussien, A. Ames, and P. Tabuada, "Abstracting partially feedback linearizable systems compositionally," *IEEE Control Systems Letters*, vol. 1, no. 2, pp. 227–232, 2017.

[11] S. W. Smith, P. Nilsson, and N. Ozay, "Interdependence quantification for compositional control synthesis with an application in vehicle safety systems," in *Decision and Control (CDC), 2016 IEEE 55th Conference on*. IEEE, 2016, pp. 5700–5707.

[12] P. Nilsson and N. Ozay, "Control synthesis for large collections of systems with mode-counting constraints," in *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control*. ACM, 2016, pp. 205–214.

[13] Y. Zheng, G. Fantuzzi, A. Papachristodoulou, P. Goulart, and A. Wynn, "Chordal decomposition in operator-splitting methods for sparse semidefinite programs," *Mathematical Programming*, pp. 1–44, 2019.

[14] A. A. Ahmadi and A. Majumdar, "DSOS and SDSOS optimization: more tractable alternatives to sum of squares and semidefinite optimization," *SIAM Journal on Applied Algebra and Geometry*, vol. 3, no. 2, pp. 193–230, 2019.

[15] R. Alur and T. A. Henzinger, "Reactive modules," *Formal methods in system design*, vol. 15, no. 1, pp. 7–48, 1999.

[16] K. Chatterjee and T. A. Henzinger, "Assume-guarantee synthesis," in *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 2007, pp. 261–275.

[17] M. G. Bobaru, C. S. Păsăreanu, and D. Giannakopoulou, "Automated assume-guarantee reasoning by abstraction refinement," in *International Conference on Computer Aided Verification*. Springer, 2008, pp. 135–148.

[18] Y. Chen, J. Anderson, K. Kalsi, S. H. Low, and A. D. Ames, "Compositional set invariance in network systems with assume-guarantee contracts," *arXiv preprint arXiv:1810.10636*, 2018.

[19] J. H. Chow and K. W. Cheung, "A toolbox for power system dynamics and control engineering education and research," *IEEE transactions on Power Systems*, vol. 7, no. 4, pp. 1559–1564, 1992.

[20] N. Sandell, P. Varaiya, M. Athans, and M. Safonov, "Survey of decentralized control methods for large scale systems," *IEEE Transactions on automatic Control*, vol. 23, no. 2, pp. 108–128, 1978.

[21] C. Zhao, U. Topcu, N. Li, and S. Low, "Design and stability of load-side primary frequency control in power systems," *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1177–1189, 2014.

[22] E. Mallada, C. Zhao, and S. Low, "Optimal load-side control for frequency regulation in smart grids," *IEEE Transactions on Automatic Control*, vol. 62, no. 12, pp. 6294–6309, 2017.

[23] D. K. Molzahn, F. Dörfler, H. Sandberg, S. H. Low, S. Chakrabarti, R. Baldick, and J. Lavaei, "A survey of distributed optimization and control algorithms for electric power systems," *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 2941–2962, 2017.

[24] A. Giani, S. Sastry, K. H. Johansson, and H. Sandberg, "The viking project: an initiative on resilient control of power networks," in *Resilient Control Systems, 2009. ISRCS'09. 2nd International Symposium on*. IEEE, 2009, pp. 31–35.

[25] P. Kundur, N. J. Balu, and M. G. Lauby, *Power system stability and control*. McGraw-hill New York, 1994, vol. 7.

[26] D. Bertsekas, "Infinite time reachability of state-space regions by using feedback control," *IEEE Transactions on Automatic Control*, vol. 17, no. 5, pp. 604–613, 1972.

[27] Y. Chen, H. Peng, J. Grizzle, and N. Ozay, "Data-driven computation of minimal robust control invariant set," in *Decision and Control (CDC), 2018 IEEE 57th Annual Conference on*. IEEE, 2018.

[28] D. Bertsimas, D. B. Brown, and C. Caramanis, "Theory and applications of robust optimization," *SIAM review*, vol. 53, no. 3, pp. 464–501, 2011.

[29] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*. IEEE, 2014, pp. 6271–6278.

[30] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017.

[31] C. S. Păsăreanu, D. Giannakopoulou, M. G. Bobaru, J. M. Cobleigh, and H. Barringer, "Learning to divide and conquer: applying the l* algorithm to automate assume-guarantee reasoning," *Formal Methods in System Design*, vol. 32, no. 3, pp. 175–205, 2008.

[32] E. Asarin, A. Donzé, O. Maler, and D. Nickovic, "Parametric identification of temporal properties," in *International Conference on Runtime Verification*. Springer, 2011, pp. 147–160.

[33] G. Bombara, C.-I. Vasile, F. Penedo, H. Yasuoka, and C. Belta, "A decision tree approach to data classification using signal temporal logic," in *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control*. ACM, 2016, pp. 1–10.

[34] V. Raman, A. Donzé, M. Maasoumy, R. M. Murray, A. Sangiovanni-Vincentelli, and S. A. Seshia, "Model predictive control with signal temporal logic specifications," in *53rd IEEE Conference on Decision and Control*. IEEE, 2014, pp. 81–87.

[35] G. E. Fainekos and G. J. Pappas, "Robustness of temporal logic specifications for continuous-time signals," *Theoretical Computer Science*, vol. 410, no. 42, pp. 4262–4291, 2009.

[36] X. Jin, A. Donzé, J. V. Deshmukh, and S. A. Seshia, "Mining requirements from closed-loop control models," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 11, pp. 1704–1717, 2015.

[37] E. S. Kim, M. Arcak, and S. A. Seshia, "A small gain theorem for parametric assume-guarantee contracts," in *Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control*. ACM, 2017, pp. 207–216.

[38] C. A. Desoer and M. Vidyasagar, *Feedback systems: input-output properties*. Siam, 1975, vol. 55.

[39] B. Chazelle, "An optimal convex hull algorithm in any fixed dimension," *Discrete & Computational Geometry*, vol. 10, no. 4, pp. 377–409, 1993.

[40] B. Kouvaritakis and M. Cannon, "Model predictive control," *Switzerland: Springer International Publishing*, 2016.

[41] J. F. Mota, J. M. Xavier, P. M. Aguiar, and M. Püschel, "Distributed admm for model predictive control and congestion control," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. IEEE, 2012, pp. 5110–5115.

[42] A. N. Venkat, I. A. Hiskens, J. B. Rawlings, and S. J. Wright, "Distributed mpc strategies with application to power system automatic generation control," *IEEE transactions on control systems technology*, vol. 16, no. 6, pp. 1192–1206, 2008.

[43] S. V. Raković, B. Kouvaritakis, M. Cannon, C. Panos, and R. Findeisen, "Fully parameterized tube mpc," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 197–202, 2011.

[44] S. Yu, C. Maier, H. Chen, and F. Allgöwer, "Tube mpc scheme based on robust control invariant set with application to lipschitz nonlinear systems," *Systems & Control Letters*, vol. 62, no. 2, pp. 194–200, 2013.

[45] P. Trodden and A. Richards, "Robust distributed model predictive control using tubes," in *2006 American Control Conference*. IEEE, 2006, pp. 6–pp.

[46] R. D. Zimmerman, C. E. Murillo-Sánchez, and D. Gan, "Matpower: A matlab power system simulation package," *Manual, Power Systems Engineering Research Center, Ithaca NY*, vol. 1, 1997.