A Distributed Logical Filter for Connected Row Convex Constraints

T. K. Satish Kumar, Hong Xu, Zheng Tang, Anoop Kumar, Craig Milo Rogers, and Craig A. Knoblock

Information Sciences Institute, University of Southern California, Marina del Rey, California, USA

tkskwork@gmail.com, hongx@usc.edu, {zhengtan, anoopk, rogers, knoblock}@isi.edu

***Abstract*—Filtering denotes any method whereby an agent updates its belief state—its knowledge of the state of the world—from a sequence of actions and observations. Popular filtering techniques like Kalman and particle filters maintain compact representations of the belief state at all times. However, these techniques cannot be applied to situations where the world is described using constraints instead of stochastic models. In such cases, the belief state is a logical formula describing all possible world states. In this paper, we first review a logical filtering algorithm for connected row convex (CRC) constraints. CRC constraints are representationally very powerful; and the filtering algorithm for CRC constraints is a logical equivalent of the Kalman filter. We later study the CRC filtering algorithm in distributed settings where nodes of a network are interested in different subsets of variables from a larger system. We deduce its reducibility to the problem of distributed path consistency (PC) and prove the compactness of the belief state representations maintained at each node at all times.**

I. Introduction

When an agent operates in a partially observable or uncertain environment, it must maintain a representation of its knowledge about the world (*belief state*). Filtering denotes any method whereby an agent updates its belief state from a sequence of *actions* and *observations*. For stochastic models with Gaussian noise in linear transitions and observations, the *Kalman filter* [1] maintains a multivariate Gaussian belief state over N system variables. In each timestep of the Kalman filter, the time complexity for updating the belief state is $O(N^3)$, and the space complexity for maintaining the belief state is $O(N^2)$. These complexities are polynomial in N and independent of the timestep. Therefore, a Kalman filter can run indefinitely.

The Kalman filter has also been decentralized to be made applicable to many large-scale dynamical systems [2]. For example, the distributed Kalman filter has been used with Wireless Sensor Networks (WSNs) to monitor power grid systems, and in weather forecast, earthquake, and target tracking systems [3]. Here, geographically distributed sensors take measurements of system variables; but each sensor is typically able to observe the values of only a subset of the variables.

State estimation in this distributed setting not only requires local information processing capabilities at each individual sensor but also requires inter-sensor communication capabilities. This networking aspect of the problem—that is absent from centralized state estimation—raises the need for an integrated design of local processing and network communication operations to optimize metrics related to energy consumption, bandwidth requirements, and system efficiency [4]. To address the challenge of limited communication, an *innovation* factor is used to characterize the importance of communicating a certain observation over a network [5]. The innovation of an observation simply measures the information content in it.

Despite the wide use of the centralized/distributed Kalman filter, it is not applicable to domains that are described using logical formulas or constraints instead of stochastic models. The reason is that the assumptions of the Kalman filter, as well as other popular filters such as the *particle filter*, are not met by such domains. In logical domains, the belief state should describe all possible *world states*. The problem of how to represent such belief states compactly using logical formulas is commonplace in automated planning, game playing [6], etc.

In logical domains, efficient logical filtering refers to maintaining a compact representation of the belief state with a (potentially unbounded) sequence of actions and observations. In a general version of the logical filtering problem, the initial state may be only partially known, the transition model that allows for actions by the agent may be nondeterministic, and the observation model may be partial or nondeterministic.

With increasing timesteps, logical filtering faces the challenge of having to maintain a compact representation of a growing number of world states. Although efficient filtering is possible for very restrictive classes [7]–[9], it is hard in general for nondeterministic domains even in propositional logic. In general, for a logical filtering algorithm, it is unviable to update each world state separately [10], expensive to record the sequence of actions and observations [11], and *incomplete/unsound* to approximate the belief state [12].

A logical equivalent of the centralized Kalman filter with analogous time and space complexities is provided in [13]. Here, *connected row convex* (CRC) constraints play the role of Gaussians; and the belief state is updated using *path consistency* (PC). CRC constraints are representationally powerful and occur commonly in many real-world domains [14], [15]. In this paper, we extend this CRC filter to distributed settings where nodes of a network are interested in different subsets of variables from a larger system. We deduce the reducibility of this problem to the problem of distributed PC and, more importantly, prove the compactness of the belief state representations maintained at each node at all times.

© 2017 IEEE. The final published version is available from IEEE via https://doi.org/10.1109/ICTAI.2017.00026

X_i \ X_j	d_{j1}	d_{j2}	d_{j3}	d_{j4}	d_{j5}
d_{i1}	0	1	0	0	0
d_{i2}	1	1	0	0	0
d_{i3}	0	1	1	0	1
d_{i4}	0	0	1	0	0
d_{i5}	0	0	1	0	0

X_i \ X_j	d_{j1}	d_{j2}	d_{j3}	d_{j4}	d_{j5}
d_{i1}	0	1	0	0	0
d_{i2}	1	0	0	0	0
d_{i3}	0	0	1	1	1
d_{i4}	0	0	1	0	0
d_{i5}	0	0	1	0	0

Fig. 1: Illustrates the (0,1)-matrix representation of constraints as well as the properties of CRC constraints. The constraint on the left qualifies as a CRC constraint. The 4^{th} column in it has only '0's. After its removal, the '1's appear consecutively, as a single band, in each row and column. Moreover, in consecutive rows and columns, these bands of '1's touch each other. The constraint on the right does not qualify as a CRC constraint. Here, although the '1's appear consecutively in each row and column, the bands in consecutive rows and columns don't touch each other. In particular, the band of '1's in the second row/column does not touch the band in the third row/column.

II. Preliminaries and Background

A *constraint satisfaction problem (CSP)* is defined by a triplet $\langle \mathcal{X}, \mathcal{D}, \mathcal{C} \rangle$, where $\mathcal{X} = \{X_1, X_2 \dots X_N\}$ is a set of variables and $\mathcal{C} = \{C_1, C_2 \dots C_M\}$ is a set of constraints. Each variable X_i is associated with a discrete-valued domain $D_i \in \mathcal{D}$, and each constraint C_i is a pair $\langle S_i, R_i \rangle$, where $S_i \subseteq \mathcal{X}$ is a subset of variables (called the *scope* of C_i) and $R_i \subseteq D_{S_i}$ ($D_{S_i} = \times_{X_j \in S_i} D_j$) denotes all compatible tuples of D_{S_i} allowed by the constraint. $|R_i|$ is called the *arity* of the constraint C_i. A *solution* to a CSP is an assignment of values to all variables from their respective domains such that all constraints are satisfied. In a *binary* CSP, the arity of any constraint is 2. Binary CSPs are representationally as powerful as CSPs and are NP-hard to solve in general.

A network of binary constraints is said to be *path consistent* iff, for any three distinct variables X_i, X_j and X_k, and for each pair of consistent values of X_i and X_j that satisfies the direct constraint $C(X_i, X_j)$, there exists a value of X_k such that the constraints $C(X_i, X_k)$ and $C(X_j, X_k)$ are also satisfied. Conceptually, algorithms that enforce PC work by iteratively "tightening" the binary constraints of a CSP [16]. When binary constraints are represented as matrices, PC algorithms employ the three basic operations of *composition*, *intersection* and *transposition*. The (0,1)-matrix representation of a constraint $C(X_i, X_j)$ between the variables X_i and X_j consists of $|D_i|$ rows and $|D_j|$ columns when orderings on the domain values of X_i and X_j are imposed. The '1's and '0's in the matrix respectively indicate the allowed and disallowed tuples. Figure 1 presents examples of such (0,1)-matrix representations of binary constraints.

A binary constraint represented as a (0,1)-matrix is *row convex* iff, in each row, all the '1's are consecutive. It has been shown in [17] that if there exists an ordering of the variables and a domain ordering for each variable in a path consistent network of binary constraints such that each constraint $C(X_i, X_j)$, with X_i appearing after X_j in the ordering, can be made row convex, then the network is also *globally consistent* along this ordering of the variables. That is, a solution can be found using a backtrack-free search that instantiates variables in that ordering. The orderings on the domain values of all variables are critical to establishing row convexity in path consistent networks, for which the result of [18] can be used.

Although row convexity implies global consistency in path consistent networks, the very process of achieving PC may destroy the property of row convexity. This means that row convexity alone does not necessarily imply global consistency. CRC constraints avoid this problem by imposing a few additional restrictions. A (0,1)-matrix is CRC if, after removing rows or columns with only '0's in them, it is row convex and *connected*, that is, the positions of the '1's in any two consecutive rows/columns intersect, or are consecutive. Unlike row convex constraints, CRC constraints are closed under composition, intersection and transposition—the three basic operations employed by algorithms that enforce PC—hence establishing that enforcing PC over CRC constraints is sufficient to ensure global consistency [19]. Figure 1 shows examples to illustrate the properties of CRC constraints.

III. The Centralized CRC Filter

In this section, we review the logical filtering algorithm for CRC constraints that is based on PC and presented in [13]. Here, the belief state can be maintained indefinitely (at all times) using a compact representation. In fact, the space complexity of this representation is $O(N^2 D^2)$, where N is the number of system variables and D is the largest domain size of any of these variables. This space complexity matches the quadratic space complexity of the Kalman filter. Moreover, the time complexity of updating the belief state at each timestep is $O(N^3 D^2)$. This is, in fact, the time complexity of the PC algorithm applied to CRC constraints. While PC takes $O(N^3 D^3)$ time in general [16], the special structure of the CRC constraints has been exploited in [20] to reduce this time complexity to $O(N^3 D^2)$. The cubic time complexity in terms of N also matches that of the Kalman filter. The CRC filter is therefore the logical equivalent of the Kalman filter with the CRC constraints playing the role of Gaussians in Kalman filtering.

Figure 2 shows a dynamically evolving system. The system state at time t is defined by an assignment of values to all N variables in $\{X_1^t, X_2^t \dots X_N^t\}$. Without loss of generality, we assume that the system is *Markovian*—i.e., its state at time $t+1$ is independent of its states before time t given its state at time t. The initial state can be a unique state specified as an assignment of values to all variables in $\{X_1^0, X_2^0 \dots X_N^0\}$; or uncertainty in the initial state is allowed as far as the possible initial states are the set of all solutions to CRC constraints specified on the variables in $\{X_1^0, X_2^0 \dots X_N^0\}$. The possible transitions of the system from time t to time $t+1$ are also defined similarly—i.e., as the set of solutions to CRC constraints specified on the variables in $\{X_1^t, X_2^t \dots X_N^t\} \cup \{X_1^{t+1}, X_2^{t+1} \dots X_N^{t+1}\}$. Finally, the observations made at time t can either be indicated using unique values to a subset

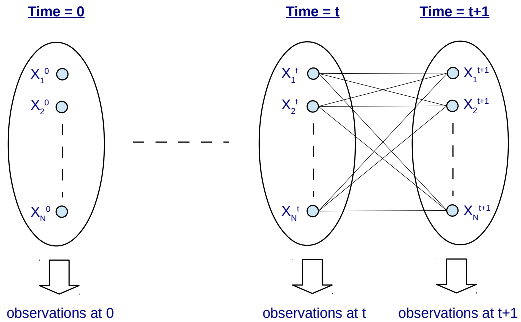

Fig. 2: Shows a dynamically evolving system. The system state at time t is defined by an assignment of values to the N variables $\{X_1^t, X_2^t \dots X_N^t\}$. The possible initial states are the set of all solutions to the constraints specified on $\{X_1^0, X_2^0 \dots X_N^0\}$. The possible transitions from time t to time $t+1$ are the set of solutions to the constraints specified on $\{X_1^t, X_2^t \dots X_N^t\} \cup \{X_1^{t+1}, X_2^{t+1} \dots X_N^{t+1}\}$. The observations made at time t can be specified as unique values to the variables in $\{X_1^t, X_2^t \dots X_N^t\}$ or more generally as constraints on $\{X_1^t, X_2^t \dots X_N^t\}$.

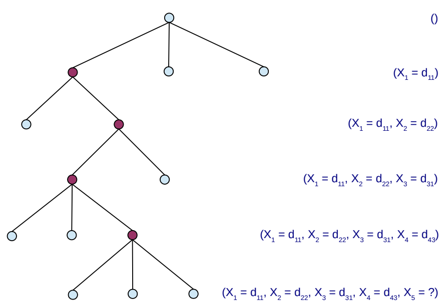

Fig. 3: Illustrates a CSP search tree. At level 1 of the search tree, X_1 is assigned the value d_{11}. At level 2, X_2 is assigned the value d_{22} known to be consistent with X_1's value. Similarly, X_3 is then assigned the value d_{31} and X_4 is assigned the value d_{43} known to be consistent with all previous commitments. The figure shows a snapshot of this search process when we are currently searching for a domain value to assign to X_5 that should be consistent with the values of all variables instantiated higher up in the search tree.

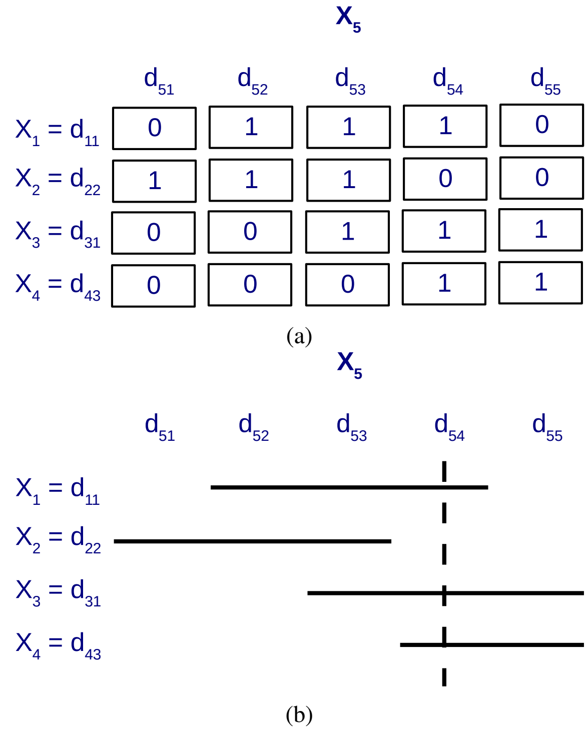

Fig. 4: Illustrates the argument of backtrack-free search for path consistent CRC constraints using the example from Figure 3. The ordered domain values of X_5 are, say, $d_{51}, d_{52} \dots d_{55}$. Each of the previously instantiated variables, $X_1 = d_{11}, X_2 = d_{22}, X_3 = d_{31}, X_4 = d_{43}$, is consistent with a continuous range of X_5's domain values that can be visualized as a horizontal line segment. A required domain value of X_5 corresponds to a vertical line that goes through all these horizontal line segments that represent bands of '1's. If such a vertical line does not exist, then simple geometry indicates that some two of the horizontal line segments do not overlap with each other. This means that some two previous commitments, $X_2 = d_{22}$ and $X_4 = d_{43}$ in this case, annihilate all domain values of X_5. But this would contradict the assumption of PC, because PC would have marked $X_2 = d_{22}$ and $X_4 = d_{43}$ as being incompatible.

of the variables in $\{X_1^t, X_2^t \dots X_N^t\}$ or be specified more generally as CRC constraints on $\{X_1^t, X_2^t \dots X_N^t\}$.

Figures 3 and 4 illustrate the basic arguments in proving that path consistent row convex constraints are also globally consistent—i.e., any partial assignment to a subset of the variables that satisfies all the direct constraints between them can also be extended to a consistent assignment to any other variable. Since CRC constraints are row convex after establishing arc consistency (AC) and are closed under the operations of PC, they can be solved using backtrack-free search if enforcing *strong PC* (PC and AC) does not lead to the annihilation of any variable's domain.

Figure 5 shows how the above properties of CRC constraints can be exploited in logical filtering. In the CRC filter, the belief state at time t is represented using a set of CRC constraints on $\{X_1^t, X_2^t \dots X_N^t\}$. Since a CRC constraint is binary, the space complexity is $O(N^2D^2)$. Maintaining the belief state using a set of CRC constraints is beneficial since we can represent an exponentially large number of possible world states as solutions to these CRC constraints. Filtering approaches that represent the belief state as a collection of complete assignments to the variables $\{X_1^t, X_2^t \dots X_N^t\}$ fail with an unwieldy space complexity. Even approaches that try to approximate the set of all solutions using ellipsoids or bounding hyperplanes fail since there exist cases where the

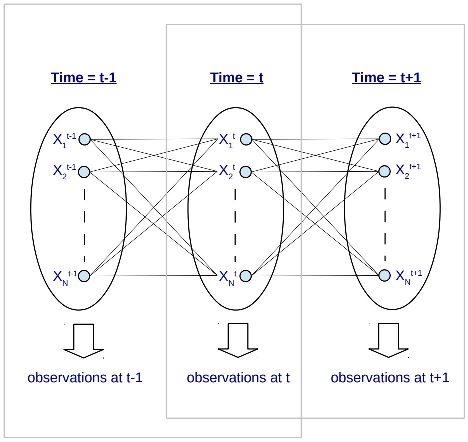

Fig. 5: Illustrates the correctness arguments of the CRC filter. The belief state at time $t+1$ is the set of all possible assignments to the variables in $\{X_1^{t+1}, X_2^{t+1} \ldots X_N^{t+1}\}$ that are consistent with the observations at time $t+1$ and have a consistent extension to the belief state at time t. Assuming, by induction, that the belief state at time t is represented as a set of CRC constraints on $\{X_1^t, X_2^t \ldots X_N^t\}$, the belief state at time $t+1$ can also be represented as a set of CRC constraints on $\{X_1^{t+1}, X_2^{t+1} \ldots X_N^{t+1}\}$. This is because the transition constraints between $\{X_1^t, X_2^t \ldots X_N^t\}$ and $\{X_1^{t+1}, X_2^{t+1} \ldots X_N^{t+1}\}$ are also CRC. The two bounding rectangles help visualize the inductive arguments.

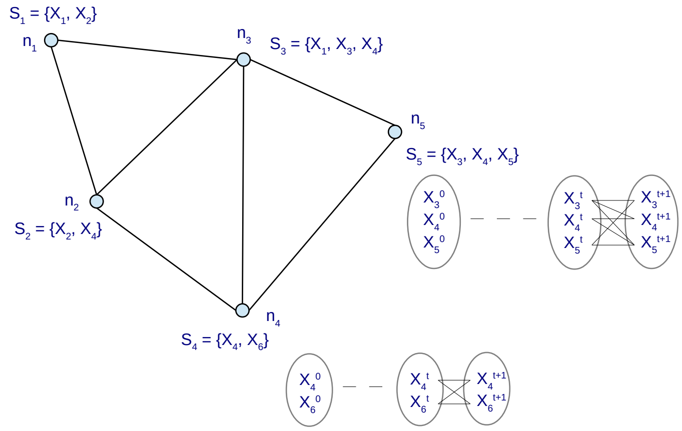

Fig. 6: Shows a network of 5 nodes $\{n_1, n_2, n_3, n_4, n_5\}$ and 6 system variables $\mathcal{X} = \{X_1, X_2, X_3, X_4, X_5, X_6\}$. Each node n_i is associated with a subset $S_i \subseteq \mathcal{X}$ of system variables that it is interested in. An assignment of values to these variables constitutes an S_i-partial state. n_i maintains an S_i-partial belief state $B(S_i^t)$ at all times: $B(S_i^t)$ is the set of all valid S_i-partial states at time t.

solutions to a set of CRC constraints cannot be represented compactly using geometrically closed regions [21].

The belief state at time $t+1$ is the set of all possible assignments to the variables in $\{X_1^{t+1}, X_2^{t+1} \ldots X_N^{t+1}\}$ that are not only consistent with the observations at time $t+1$ but also have a consistent extension to the belief state at time t through the transition constraints. Since the observations, transitions, and by induction, the belief state at time t are all expressed using CRC constraints, establishing PC on these CRC constraints induces a new set of CRC constraints on the variables in $\{X_1^{t+1}, X_2^{t+1} \ldots X_N^{t+1}\}$. These induced CRC constraints are retained as a representation of the belief state at time $t+1$. After establishing PC, any solution to these CRC constraints can be extended to the variables in $\{X_1^t, X_2^t \ldots X_N^t\}$, which, in turn by induction, can be extended to all previous variables as well. This proves the correctness of the belief state update procedure that employs PC. The time complexity of updating the belief state is therefore $O(N^3 D^2)$ since it requires establishing PC on CRC constraints over $2N$ variables [20].

IV. The Distributed CRC Filter

The distributed state estimation problem arises in many real-world applications. Appropriate distributed filtering techniques are therefore needed to address this problem. The distributed Kalman filter, for example, is used for distributed state estimation in WSNs [2], [22], which in turn can be used for monitoring large-scale systems [3]. Many real-world problem domains, however, are better described using constraint models instead of stochastic models required for the distributed Kalman filter. One such example is in distributed monitoring of plan execution. Here, plans are naturally described using preconditions and effects of actions which can be easily converted to constraint models. We therefore develop the distributed CRC filter to serve as a logical analogue of the distributed Kalman filter. We show that the same intuitions about CRC constraints and the remarkable effects of establishing PC on them can be extended to distributed settings as well.

In a distributed setting, as shown in Figure 6, there are L nodes $\{n_1, n_2 \ldots n_L\}$ on a network. We assume that the network is connected so that any two nodes can communicate with each other by staying oblivious to load balancing algorithms implemented in the physical layer. Each node n_i is interested in a subset $S_i \subseteq \{X_1, X_2 \ldots X_N\}$ of the system variables. If $S_i = \{X_{i_1}, X_{i_2} \ldots X_{i_{|S_i|}}\}$, we define $S_i^t = \{X_{i_1}^t, X_{i_2}^t \ldots X_{i_{|S_i|}}^t\}$ to reason about the dynamically evolving subsystems. Since the variables in S_i^t do not suffice to define the global state of the system at time t, we refer to an assignment of values to the variables in S_i^t as an S_i-*partial state* at time t. An assignment of values to all variables in S_i^t is a *valid* S_i-partial state at time t if there exists a consistent extension of this assignment to all variables in $\bigcup_{t' \leq t}\{X_1^{t'}, X_2^{t'} \ldots X_N^{t'}\} \setminus S_i^t$. Each node is required to maintain $B(S_i^t)$, the S_i-*partial belief state* at time t, that comprises of all valid S_i-partial states at time t. Without loss of generality, we also assume that at time t, a node n_i can make observations only on variables in S_i^t. If node n_i receives observations on variables not in S_i^t, it simply forwards them to the relevant nodes interested in them.

Figure 7 shows how properties of CRC constraints can

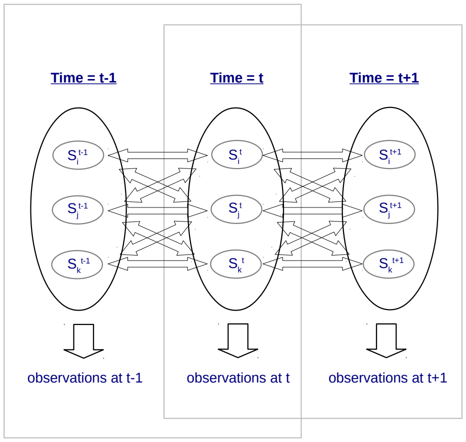

Fig. 7: Illustrates the correctness arguments of the distributed CRC filter. The large ellipses indicate the global states at times $t-1$, t and $t+1$. The smaller ellipses within each of them indicate the partial states at these times. Three partial states, corresponding to subsets S_i, S_j and S_k, are shown for each of these times. By induction, we assume that all $S_{i/j/k}$-partial belief states at time t' are represented as CRC constraints on the variables in $S_{i/j/k}^{t'}$ for $t' \leq t$. Since all other constraints between time t and time $t+1$ are also CRC, the idea is to establish PC on all these constraints using a distributed algorithm and retain only the CRC constraints between the variables in $S_{i/j/k}^{t+1}$ as the $S_{i/j/k}$-partial belief state at time $t+1$. The two bounding rectangles help visualize the inductive arguments.

be used to design an efficient distributed logical filter. In this distributed CRC filter, $B(S_i^t)$, the S_i-partial belief state maintained by node n_i at time t, is represented using a set of CRC constraints on the variables in S_i^t. The space complexity for representing $B(S_i^t)$ is therefore only quadratic in the largest domain size of any variable and the total number of variables in S_i^t. The S_i-partial belief state at time $t+1$ is the set of all possible assignments to the variables in S_i^{t+1} that are not only consistent with the observations at time $t+1$ but also have a consistent extension to all other variables in $\bigcup_{t' \leq t+1}\{X_1^{t'}, X_2^{t'} \dots X_N^{t'}\} \setminus S_i^{t+1}$.

Assume by induction that the set of CRC constraints between the variables in S_i^t represents $B(S_i^t)$ and the set of CRC constraints between all variables in $\{X_1^t, X_2^t \dots X_N^t\}$ represents the global belief state at time t, i.e., the set of all assignments to variables in $\{X_1^t, X_2^t \dots X_N^t\}$ that have a consistent extension to all other variables in $\bigcup_{t'<t}\{X_1^{t'}, X_2^{t'} \dots X_N^{t'}\}$. If the possible initial states are specified using CRC constraints on $\{X_1^0, X_2^0 \dots X_N^0\}$, this property required for the base case of the induction can be enforced simply by establishing PC on $\{X_1^0, X_2^0 \dots X_N^0\}$ and retaining only the CRC constraints between the variables in S_i^0. $B(S_i^{t+1})$ can be obtained by establishing PC on all the CRC constraints between the variables in $\{X_1^t, X_2^t \dots X_N^t\} \cup \{X_1^{t+1}, X_2^{t+1} \dots X_N^{t+1}\}$ and retaining only the CRC constraints between the variables in S_i^{t+1}. The reason for this follows from the fact that all constraints are CRC, and after PC, any consistent assignment of values to the variables in S_i^{t+1} can be extended to all other variables in $\{X_1^t, X_2^t \dots X_N^t\} \cup \{X_1^{t+1}, X_2^{t+1} \dots X_N^{t+1}\} \setminus S_i^{t+1}$. Moreover, by induction, we also know that any consistent assignment of values to $\{X_1^t, X_2^t \dots X_N^t\}$ can be extended to all other variables in $\bigcup_{t'<t}\{X_1^{t'}, X_2^{t'} \dots X_N^{t'}\}$. Put together, therefore, any consistent assignment of values to variables in S_i^{t+1} can be extended to all other variables in $\bigcup_{t' \leq t+1}\{X_1^{t'}, X_2^{t'} \dots X_N^{t'}\} \setminus S_i^{t+1}$. This proves the correctness of the PC-based procedure for updating the S_i-partial belief states at each node. A similar argument also proves that the set of CRC constraints between the variables in $\{X_1^{t+1}, X_2^{t+1} \dots X_N^{t+1}\}$ retained after PC represents the global belief state at time $t+1$.

We note that each node n_i maintains not only the current set of CRC constraints between the variables in S_i^t but also any other CRC constraint that involves at least one of the variables in S_i^t. This is because the global belief state at time t relies on all these CRC constraints—including the ones that involve variables from two different S_i^t and S_j^t. These CRC constraints between any two variables in $\{X_1^t, X_2^t \dots X_N^t\}$ are all important for updating the partial belief states at time $t+1$.

From the foregoing arguments, it is clear that the problem of computing partial belief states for each node is now reduced to the problem of establishing PC on CRC constraints using a distributed algorithm. Although distributed PC algorithms are not well studied, some attempts can be found in [23], [24]. The difficulties faced in designing an efficient distributed PC algorithm are analogous to the ones faced in designing a distributed Kalman update procedure for stochastic models where heuristic notions akin to *innovations*, *signs of innovations*, etc. have to be used [4], [5], [22]. While the space and time complexities of the distributed CRC filter are clearly upper bounded by those of the centralized CRC filter, the message complexity of the distributed CRC filter matches the message complexity of the distributed PC algorithm that it relies on. We also note that, unlike in a distributed Kalman filter, retrieving a valid state from the global belief state does not require distributed PC since CRC constraints can be solved directly using a very efficient randomized distributed algorithm [25].

V. Application Domains

There are many problem domains in which logical equivalents of the Kalman filter can be applied. For example, both the centralized as well as the distributed CRC filters are particularly important in monitoring the executions of plans. Although the planning problem itself is typically hard, the execution of an already generated valid plan is usually flexible under some simple temporal constraints [23]. These simple temporal constraints are, in fact, known to be CRC [26]. In temporal reasoning, CRC constraints also arise in scheduling problems that are described using restricted disjunctive temporal constraints [26], temporal constraints with domain rules [14], or temporal constraints with taboo regions [27]. Another domain in which CRC constraints arise frequently is in geometric reasoning with max-distance constraints [15]. Yet another example where CRC filtering is important is in multi -robot localization [13].

In some sense, the CRC filter can be deemed as being important simply because it is the logical analogue of the Kalman filter. Even if the constraint-based description of the world does not make use of only CRC constraints, the CRC filter can still serve as an approximation or pave the way for a more expressive logical filter. The same is true for the distributed CRC filter as well.

VI. Conclusions and Future Work

We studied the CRC filter as a logical equivalent of the Kalman filter. In centralized settings, the CRC constraints play the role of Gaussians; and the belief state is updated using PC. In distributed settings, where nodes of a network observe different subsets of variables from a larger system, we showed that PC can be used again to extend a centralized CRC filter to a distributed CRC filter. Our distributed CRC filter maintains compact representations of the partial belief states at each node at all times and relies on distributed PC algorithms. The message complexity of updating the partial belief states at each node matches that of the distributed PC algorithm.

There are many directions for future work. One direction is to apply CRC filters as approximations to general constraint models (in the same sense that Kalman filters are used with approximations). Another interesting direction is to generalize CRC filters based on PC to richer classes of constraints based on higher levels of local consistency. Yet another direction is to combine CRC filters with Kalman filters when the world is described using constraints as well as stochastic models.

VII. Acknowledgment

This material is based upon work supported by the Air Force Research Laboratory (AFRL) and the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-15-C-0138. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the official views or policies of the Department of Defense or the U.S. Government.

References

[1] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.

[2] B. S. Y. Rao, H. F. Durrant-Whyte, and J. A. Sheen, "A fully decentralized multi-sensor system for tracking and surveillance," *International Journal of Robotics Research*, vol. 12, no. 1, pp. 20–44, 1993.

[3] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[4] J.-J. Xiao, A. Ribeiro, Z.-Q. Luo, and G. B. Giannakis, "Distributed compression-estimation using wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 23, no. 4, pp. 27–41, 2006.

[5] A. Ribeiro, G. B. Giannakis, and S. Roumeliotis, "SOI-KF: Distributed Kalman filtering with low-cost communications using the sign of innovations," *IEEE Transactions on Signal Processing*, vol. 54, no. 12, pp. 4782–4795, 2006.

[6] S. Russell and J. Wolfe, "Efficient belief-state AND-OR search with application to Kriegspiel," in *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, 2005, pp. 278–285.

[7] E. Amir and S. Russell, "Logical filtering," in *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, 2003, pp. 75–82.

[8] R. Fikes, P. Hart, and N. Nilsson, "Learning and executing generalized robot plans," *Artificial Intelligence*, vol. 3, pp. 251–288, 1972.

[9] F. Lin and R. Reiter, "How to progress a database," *Artificial Intelligence*, vol. 92, no. 1–2, pp. 131–167, 1997.

[10] A. Cimatti and M. Roveri, "Conformant planning via symbolic model checking," *Journal of Artificial Intelligence Research*, vol. 13, pp. 305–338, 2000.

[11] R. Reiter, *Knowledge in Action*. MIT Press, 2001.

[12] T. C. Son and C. Baral, "Formalizing sensing actions—A transition function based approach," *Artificial Intelligence*, vol. 125, no. 1–2, pp. 19–91, 2001.

[13] T. K. S. Kumar and S. Russell, "On some tractable cases of logical filtering," in *Proceedings of the 16th International Conference on Automated Planning and Scheduling*, 2006, pp. 83–92.

[14] T. K. S. Kumar, "Tractable classes of metric temporal problems with domain rules," in *Proceedings of the 21st AAAI Conference on Artificial Intelligence*, 2006, pp. 847–852.

[15] ——, "On geometric CSPs with (near)-linear domains and max-distance constraints," in *Proceedings of the ECAI-2004 Workshop on Modeling and Solving Problems with Constraints*, 2004.

[16] R. Dechter, *Constraint Processing*. Morgan Kaufmann, 2003.

[17] P. van Beek and R. Dechter, "On the minimality and global consistency of row convex constraint networks," *Journal of the ACM*, vol. 42, no. 3, pp. 543–561, 1995.

[18] K. Booth and G. Lueker, "Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms," *Journal of Computer and System Sciences*, vol. 13, no. 3, pp. 335–379, 1976.

[19] Y. Deville, O. Barette, and P. Van Hentenryck, "Constraint satisfaction over connected row-convex constraints," *Artificial Intelligence*, vol. 109, no. 1–2, pp. 243–271, 1999.

[20] Y. Zhang and S. Marisetti, "Solving connected row convex constraints by variable elimination," *Artificial Intelligence*, vol. 173, no. 12–13, pp. 1204–1219, 2009.

[21] L. El Ghaoui and G. Calafiore, "Confidence ellipsoids for uncertain linear equations with structure," in *Proceedings of the 38th IEEE Conference on Decision and Control*, vol. 2, 1999, pp. 1922–1927.

[22] R. Olfati-Saber, "Distributed Kalman filtering for sensor networks," in *Proceedings of the 46th IEEE Conference on Decision and Control*, 2007, pp. 5492–5498.

[23] J. C. Boerkoel Jr., L. R. Planken, R. J. Wilcox, and J. A. Shah, "Distributed algorithms for incrementally maintaining multiagent simple temporal networks," in *Proceedings of the 23rd International Conference on Automated Planning and Scheduling*, 2013, pp. 11–19.

[24] F. Fioretto, T. Le, W. Yeoh, E. Pontelli, and T. C. Son, "Improving DPOP with branch consistency for solving distributed constraint optimization problems," in *Proceedings of the 20th International Conference on Principles and Practice of Constraint Programming*, 2014, pp. 307–323.

[25] T. K. S. Kumar, D. T. Nguyen, W. Yeoh, and S. Koenig, "A simple polynomial-time randomized distributed algorithm for connected row convex constraints," in *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, 2014, pp. 2308–2314.

[26] T. K. S. Kumar, "On the tractability of restricted disjunctive temporal problems," in *Proceedings of the 15th International Conference on Automated Planning and Scheduling*, 2005, pp. 110–119.

[27] T. K. S. Kumar, M. Cirillo, and S. Koenig, "Simple temporal problems with taboo regions," in *Proceedings of the 27th AAAI Conference on Artificial Intellige*, 2013, pp. 548–554.