

センサ/アクチュエータネットワークにおける ネットワーク切断を考慮した確率推論機構

間 博 人[†] 門 田 昌 哉[†]
中 澤 仁[†] 徳 田 英 幸[†]

人と環境の状態を取得可能な多数の小型無線センサノードを埋め込むことで、コンテキストに応じた機器を制御するセンサ/アクチュエータネットワークはシームレスコンピューティングの要素技術の1つである。シームレスコンピューティングを実現するうえで、センサからアクチュエータへのスムーズな結合は、必要不可欠である。本論文では、中継ノードの故障や移動で基幹サーバと切断したセンサノードがベイジアンネットワークを用いた推論、制御をセンサ/アクチュエータネットワーク内で実行する機構として Probabilistic Inference Mechanism for Sensor-Actuator Networks with Disconnected Operation (PIMSANDO) を提案する。基幹サーバでセンサノードのデータの収集・解析、アクチュエータに対する制御を行う中央集中型のセンサ/アクチュエータネットワークでは、中継ノードの故障や移動が生じた結果、基幹システムから断絶するとデータのモニタリング、データの解析、アクチュエータに対する制御といったすべての機能が停止してしまう。PIMSANDO はネットワーク切断に耐性のあるセンサ/アクチュエータネットワークを実現するため、基幹サーバで行っている推論機能をセンサノードが代替する手法をとる。センサノードは切断を検知すると一時的にセンサネットワークでの推論を実行し、アクチュエータに対する制御を行う。PIMSANDO のプロトタイプとして MICA Mote を用いた実証実験システムを構築し、推論するコンテキストの複雑度に応じた送信量と解析時間の検証を行い、提案した推論代替モデルの有効性と限界を示した。

A Probabilistic Inference Mechanism for Sensor-actuator Networks with Disconnected Operation

HIROTO AIDA,[†] MASAYA KADOTA,[†] JIN NAKAZAWA[†]
and HIDEYUKI TOKUDA[†]

Sensor-actuator network is a key component of seamless computing. This paper proposes a novel context inference mechanism for sensor-actuator networks called PIMSANDO, which enables the context inference based on Bayesian networks by sensor nodes that are disconnected from the sink node. In a statically centralized sensor-actuator network, the backbone server connected to the sink node is the single static node that is always responsible for inference and actuation. Therefore, if the sensor nodes are disconnected from the backbone server due to power failure, movement, or other events of relay nodes, they cannot keep their actuation capability since they rely context inference functionality on the backbone server. PIMSANDO achieves robustness against the disconnection by delegating a part of inference and actuation responsibility to the sensor nodes. When a sensor node detects a disconnection, meaning that the backbone server cannot continue to infer contexts, the sensor node temporarily starts inference and actuations based on inferred contexts. When the node detects a reconnection, it delegates the inference/actuation responsibility back to the backbone server. By delegating the responsibility back and forth, PIMSANDO can adapt to the network failure keeping a powerful context inference scheme based on Bayesian networks. This paper shows a prototype of the system using MICAz Motes and evaluates it with various contexts with different complexity to infer. The evaluation shows that our mechanism decreases the network/computational load in sensor nodes, thereby entailing energy efficiency with flexible and robust context inference using Bayesian networks.

1. はじめに

小型無線センサノードを家具、玩具、家電機器といった様々なものに埋め込み、環境の物理データをモ

[†] 慶應義塾大学大学院政策・メディア研究科
Graduate School of Media and Governance, Keio University

ニタリングするセンサネットワークは、様々なデバイスを透過的に接続するシームレスコンピューティングの要素技術の1つである^{6),13)}。センサ/アクチュエータネットワークでは、センサネットワークを介して得たデータから人や環境の状態をコンテキストとして抽出し、アクチュエータをシームレスに制御する必要がある。センサデータからコンテキストを抽出する手法の1つにベイジアンネットワークの利用がある。ベイジアンネットワークは、グラフ構造によって結合確率分布を表現するとともに視覚的にセンサデータ間の依存関係をモデル化できる。また、学習アルゴリズムを用いて条件付き確率分布の学習をすることで、不確実な要因下でも妥当な推論が可能となるアルゴリズムが確立している¹⁾。

センサ/アクチュエータネットワークの基本的な問題の1つに、潜在的なネットワーク切断の可能性がある。ネットワークを構築するノードは、中継ノードの移動、故障、電池切れにより基幹サーバとの切断が生じる。一方、センサ/アクチュエータネットワークにおけるコンテキストの推論およびアクチュエータに対する制御は、センサネットワーク外の基幹サーバで行う手法が一般的である¹²⁾。したがって、こうした中央集中型のセンサ/アクチュエータネットワークでは、ネットワーク切断時には機能が停止してしまう。

この問題を解決するために、本論文ではセンサ/アクチュエータネットワークにおけるネットワーク切断時の耐故障性の実現を目的とした Probabilistic Inference Mechanism for Sensor-Actuator Networks with Disconnected Operation (PIMSANDO) を提案する。PIMSANDO は、ネットワーク切断時に基幹サーバで実現していた推論機能をセンサノードが代替する機能を持つ。同機構では、センサノードが基幹サーバからの切断を検知し、その時点で接続可能なセンサノードのセンサデータを収集し、推論とアクチュエータの制御を行う。また基幹サーバへの接続が回復すると、推論とアクチュエーションの実行を同サーバへ戻す。これにより、中継ノードの移動や、突発的な故障によるネットワーク切断時にも、推論およびアクチュエーション機能の継続性を確保できる。

本論文の構成は、次章でネットワーク切断に対する解決方法を考察し、推論機能のセンサノードへの代替手法として3つのモデルを示す。3章ではPIMSANDOに必要な機能の要件について考察し、PIMSANDOの設計について詳述する。4章でPIMSANDOの実験評価を行い、5章で関連研究について述べる。最後に6章で本論文をまとめる。

2. ネットワーク切断と推論機能の代替

本章では、まずネットワーク切断の原因と結果を整理する。次に解決手法として推論機能の代替手法を3つに分類し比較検討を行う。

2.1 センサ/アクチュエータネットワークにおけるネットワーク切断

図1に示すようなセンサ/アクチュエータネットワークを想定する。センサネットワークを構成するセンサノードは、取得したデータをシンクノードを介して基幹サーバへ送信する。基幹サーバでは、システム要求に応じたコンテキストを検知するため確率推論を実行し、その結果に基づいてアクチュエータを制御する。こうした中央集中型のセンサ/アクチュエータネットワークでは、中継ノードの移動(ノードAなど)、バッテリー切れ、故障によるネットワーク切断の可能性がある。図1においてシンクノードが故障した場合、センサノードから基幹サーバへのデータ送信は失敗し、推論機能およびアクチュエータの制御も停止する。

こうしたネットワーク切断にともなう問題を解決するには、時間的、空間的冗長性を持たせる手法と、推論機能を代替させる手法が考えられる。空間的冗長性を持たせる例として、単一故障点となるシンクノードを複数配置する手法がある。しかし、この手法では中継ノードの故障や移動には対応できない。対象エリアに複数系統のセンサ/アクチュエータネットワークを敷設する手法も考えられるが、敷設コストの増大や無線帯域の消費といった問題が残る。時間的冗長性を持たせる例として、再送などのネットワーク技術を用いる方法がある。単純な再送では、中継ノードの故障といった本研究が想定する長期間のネットワーク切断への対応は難しい。こうした長期間の切断を想定した研究としては、Delay-Tolerant Network⁷⁾ などがある。

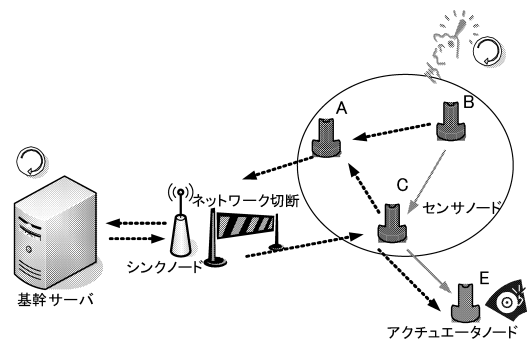


図1 センサ/アクチュエータネットワークにおけるネットワーク切断

Fig. 1 Network disconnection in sensor-actuator network.

またアドホックネットワークを対象としたものとして文献 10) や Message Ferry¹⁷⁾ がある。これらは、ネットワークが切断しても再接続時にデータを集めることや、データを宛先に届けることが課題である。しかしセンサ/アクチュエータネットワークでは、応答性が要求される。たとえば、1時間のネットワーク切断が発生し、再接続後に1時間前のコンテキストが抽出できたとしても、アクチュエータを制御する時期を逸している可能性が高い。結果として時間的冗長度を持たせる手法では、切断時に推論機能とアクチュエータの制御は停止してしまう。

推論機能の代替は、基幹サーバの推論機能およびアクチュエータに対する制御をセンサノードが代替する手法である。基幹サーバから切断したノード間で推論を実現することで、ネットワーク切断時にも推論機能とアクチュエータの制御を継続して実行可能である。次節において、推論機能の代替手法を3つに分類し比較検討する。

2.2 推論機能の代替

推論機能の代替とは、基幹サーバの推論機能をセンサノードが代替することで、基幹サーバとセンサノードが切断した際にセンサ/アクチュエータネットワークの機能継続を実現するための手法である。前提として、通常時はつねに基幹サーバ側で推論を行っているものとする。推論機能の代替手法においては、切断に対する準備と切断時の推論の手法を対象とする。

基幹サーバの機能をセンサノードが代替するにあたり、基幹サーバのどの機能をセンサノードで代替するかが問題となる。センサ/アクチュエータネットワークにおける基幹サーバの主な機能は、学習アルゴリズムを用いた条件付き確率分布の学習、確率推論の計算、コンテキストの検知に基づくアクチュエータノードの制御の3つである。このうちアクチュエータノードの制御は、ネットワーク切断時の機能継続を実現するためにはセンサノード側が必ず代替する必要がある。一方、条件付き確率分布の学習と確率推論の計算に関しては、センサノードが代替して行う手法のほかに、基幹サーバで切断前に計算した計算結果をセンサノードに渡すという手法も可能である。以下では、切断時にどの機能をセンサノードで代替するかという観点から、推論機能の代替手法を、a) 分散学習、分散確率推論型、b) 集中学習、分散確率推論型、c) 集中学習、集中確率推論型の3つに分類し議論する。

a. 分散学習、分散確率推論型 (DLDP)

DLDP は、切断時に条件付き確率分布の学習と確率推論の計算をセンサノードが代替する手法である。

センサノード側ですべてのデータの解析、制御を行うセンサネットワークはいくつか提案されている¹¹⁾ が、ここではネットワーク切断時の推論機能の代替という観点から論ずる。

センサノード側でのデータの解析・制御は、解析すべきデータや手法が固定で、データの解析が閾値による判定のような処理コストが小さい場合は有効である。しかし、データ解析手法に本論文で提案するようなベイジアンネットワークを用いる場合、計算コストと条件付き確率分布の学習に必要な多量の学習データの保存領域が問題となる。センサノードで条件付き確率分布の学習を行う場合、学習データ自身にコンテキストの教師例となるメタデータを付与しにくいという問題点もある。そのため、不完全な学習データを前提として条件付き確率分布の学習を行う必要があるが、Expectation-Maximization アルゴリズム¹⁾ のような山登り法に基づくアルゴリズムを適用する場合、繰り返しベイズ確率推論を実行する必要があり、その計算量も問題となる。

b. 集中学習、分散確率推論型 (CLDP)

CLDP は、確率推論の計算をセンサノードで代替し、条件付き確率分布は基幹サーバにおいて学習する手法である。センサノードは条件付き確率分布などのベイジアンネットワークのモデルを保持し、確率推論を行う。確率分布は基幹サーバにおいて学習し、基幹サーバは結果をセンサノードに配布する。ノード上で確率推論分布を算出するため、CLCP と比べセンサノード上での計算コストが高くなる。CLDP は、頻繁にモデルが更新される状況で配置性の面で有効だといえる。

c. 集中学習、集中確率推論型 (CLCP)

CLCP は、基幹サーバで確率分布の学習、および考えられる組合せに対しての確率推論分布を計算する。センサノードは、基幹サーバから配布された確率推論分布と実際の測定値を比較してコンテキストの検知を行うことで切断時の推論機能の継続を実現する。CPDP と比較し、非力なセンサノード上で確率推論分布を算出する必要がない反面、確率分布のプロパゲーションに必要なデータ量が問題となる。このため確率分布を頻繁に更新する必要がない場合、計算量の面で有効である。

2.3 推論機能の代替手法の比較

3つの推論機能の代替手法を効率性、欠損耐性、即応性、配置可能性の項目で比較したものを表1に示す。効率性は、ネットワーク切断時のセンサノードの計算コストを比較する項目である。推論機能の代替手法を

表 1 推論機能の代替手法の比較

Table 1 Comparison of inference delegation scheme.

	効率性	欠損耐性	応答性	配置性
DLDP	×	○	△	○
CLDP	△	○	△	△
CLCP	○	△	○	×

適用することで、ネットワーク切断時のセンサノードの計算コストは増加する。特にベイズ確率推論アルゴリズムは、センサデータを用いてコンテキストの事後確率を求める際繰り返しサンプリングすることで推論結果の変動を抑制するため、計算量は大きくなりがちである。3つの推論機能の代替手法のうち、確率推論の計算と条件付き確率分布の学習をセンサノードが代替して行うDLDPは、最もセンサノードの計算コストが大きくなると考えられる。確率推論の計算はセンサノードで代替し、条件付き確率分布の学習は基幹サーバで行うCLDPのセンサノードの計算コストは、条件付き確率分布の学習の処理コスト分DLDPよりも小さくなる。確率推論の計算と条件付き確率分布の学習を基幹サーバが計算するCLCPは、3つの手法の中で最も切断時のセンサノードの処理コストが小さくなると考えられる。

欠損耐性は、複数のセンサノードからセンサデータを集めて推論を行う際に、収集予定のセンサデータが欠損した場合の耐性について比較する項目である。たとえば、A, B, Cの3台のセンサノードの照度データをノードAに集めて確率推論を実行する際にノードCから応答がなくノードA, Bの照度データのみが収集できる場合についてを考える。DLDPおよびCLDPは、センサノード上で確率推論を計算する機能を持つため、ノードAはA, Bの照度データの組合せから推論結果を計算可能である。したがってDLDPおよびCLDPは、センサデータにどれほど欠損があっても推論プロセスは実行できるため、つねに欠損耐性があるといえる。これに対しCLCPでは、基幹サーバ側があらかじめノードA, B, Cの組合せで確率推論を計算し、ノードAはその計算結果を持つ。そのためAとBの照度データしか集まらなかった場合、推論プロセスは中断する。CLCPは想定した組合せのセンサデータが欠損していれば、推論プロセスが中断するため欠損耐性はないといえる。この問題に対処する方法として、基幹サーバ側があらかじめ欠損を織り込んだ推論結果をセンサノードへ送信するという方法が考えられる。上記の例の場合、基幹サーバがノードA, B, Cの組合せの結果とともにノードA, Bの組合せの推論結果をセンサノードに送ることで、ノードCの

照度データが欠損した場合にもノードAは推論プロセスが実行可能となる。

応答性は、センサノード上でコンテキストを検知するまでにかかる時間を比較する項目である。動的に変化するコンテキストに対し、アクチュエーションが追従するためには、コンテキストを検知する際の応答性が重要である。たとえば、「Working」コンテキストの抽出に数分を要した場合、推論結果の意味が失われてしまうおそれがある。CLCPは、基幹サーバから配布された事後確率分布と現在のセンサデータを比較することでコンテキストの抽出が完了するため、計算処理に要する時間が少ない。したがって、最新のセンサデータを収集してからコンテキストを検知するまでの時間は短く、動的に変化するコンテキストへの追従が容易である。これに対しDLDPおよびCLDPは確率推論の計算をセンサノードが代替して行うため、確率推論の計算時間の分だけコンテキストの検知に時間がかかる。したがって、最新のセンサデータを収集してからコンテキストを検知するまでの時間はCLCPと比較して長くなり動的に変化するコンテキストへの追従が難しくなる。

配置性は、基幹サーバに対する依存度の指標である。DLDPは、一度確率推論モデルを構築しセンサノードに配布しておけば、新しい確率推論モデルが必要にならない限り、センサノードが基幹サーバから長期間切断される環境下でも問題は生じない。したがって屋外などの電力供給が困難な環境でも配置が容易である。CLDPは、学習が収束した状況を想定すれば、DLDPと同様に基幹サーバから長期間切断されても問題は生じない。CLCPは、基幹サーバから配布された事後確率分布の組合せからコンテキストを検出するため、切断した状況下では、たとえば新たにセンサノードが追加されるといった変化に対し対応できない。したがって定期的に基幹サーバと接続し、事後確率分布を更新する必要がある。

3. PIMSANDO

本章では、推論機能の代替手法としてCLCPを採用することで、ネットワーク切断時の機能継続を実現するPIMSANDOの詳細を説明する。まず、いくつかPIMSANDOの前提となる事項を明らかにする。次にPIMSANDOにおける特徴的な機能として、1) 基幹サーバ側の機能である推論結果の圧縮、および2) センサノード側の機能として、ネットワーク内推論およびアクチュエーションのトリガーとなる切断、再接続の検知を提案する。

3.1 定義

PIMSANDO の前提として、利用機器、センシング、学習と推論、アクチュエーションの 4 つを定義する。

3.1.1 利用機器

センサまたはアクチュエータ機能、計算機能、通信機能、メモリ、バッテリーを持つ小さなノードがネットワークを構成する。各ノードが持つ計算資源、通信能力、メモリ、電力のリソースは非常に限られている。我々の身の回りの物や人に埋め込まれている環境を想定するため、固定ノードもあれば、移動するノードもある。また、各ノードは信頼性のあるデータ収集プロトコルと特定のノードへ通信可能な経路制御プロトコルを持つことを前提とし、本論文ではデータの配送方法に関する議論はしない。

3.1.2 センシング

抽象度に応じてデータを次の 3 段階に分ける。

- センサデータ
- 下位コンテキスト (例: dark または bright)
- 上位コンテキスト (例: Working, Meeting など)

センシングの結果得られるセンサデータは、センサの多様性から測定対象の持つ特徴に応じて、名義尺度、順序尺度、間隔尺度、比率尺度といった異なる尺度を持つ。下位コンテキストは、異なる尺度のセンサデータを同一の枠組みで扱うために、センサデータに 1 つないしは複数の閾値を設定して離散化したものである。たとえば、0 から 100 の値を出力する照度センサのデータにおいて閾値を 50 とすると、閾値 50 以上を bright、閾値 50 以下を dark と表すことができる。この照度センサの測定値が 60 であれば下位コンテキストは bright となり、10 であれば下位コンテキストは dark となる。この場合の下位コンテキストは 2 値だが、閾値を複数設定することで、より細粒度の照度を表現することも可能である。下位コンテキストの閾値は、一般的にシステムの要求やユーザの経験などにより決定するため本論文では既知のものとする。

上位コンテキストは、複数の下位コンテキストから抽出する。3.2 節で、具体的な上位コンテキストのモデル化の方法を説明する。

3.1.3 学習と推論

PIMSANDO における学習と推論の流れと特徴を明らかにする。まず、システムの要求するコンテキストをベイジアンネットワークを用いてモデル化する。モデル化の詳細は 3.2 節で述べる。次に、基幹サーバ側で条件付き確率分布の学習を行い、確率推論アルゴリズムを用い事後確率分布を算出する。第 3 に、事後確率分布をセンサノードへ送る。これは CLCP の手法

を用いる PIMSANDO の特徴の 1 つである。この際に事後確率分布のデータ量が問題となるため、3.3 節で事後確率分布の圧縮手法を提案する。第 4 に、切断時にセンサノードは、基幹サーバから送られた事後確率分布を観測値と単純に比較することでコンテキストを得る。コンテキストのモデルが複数のセンサノードのセンサデータから構成されている場合は、センサデータを収集する必要がある。センサデータの収集は、各ノードが測定したセンサデータを定期的にブロードキャストすることで行う。したがって、現状では推論に用いるセンサノードは互いに隣接している必要がある。これにより、隣接する複数センサノードのセンサデータを収集可能だが、収集したセンサデータを用いて推論するセンサノードは 1 台、制御するアクチュエータが 1 台であることを前提とする。DLDP や CLDP は、コンテキストを得る際に確率推論アルゴリズムを用いて事後確率分布を算出する必要があるが、CLCP の手法を用いる PIMSANDO は、短時間でコンテキストを検知できる。この際問題となる、センサノードでの切断、再接続の検知の方法に関しては、3.4 節で説明する。

3.1.4 アクチュエーション

アクチュエーションの種類には LED の制御、モータの制御、ブザーの制御のような基本的なものから、空調制御、照明制御といった高機能なものまで様々な種類がある。アクチュエータの種類に応じて具体的な制御の振舞いは異なるが、本質的にアクチュエーションとはアクチュエータに対するオンオフの制御である。したがって、本論文ではすべてのアクチュエーションを一義的にとらえ、アクチュエーションの種類に依存した議論はしない。

3.2 ベイジアンネットワークを用いたコンテキストのモデル化

コンテキストとは人や環境の状態であり、下位コンテキストの集合からモデル化する。ネットワーク内推論を実現するうえでコンテキストのモデル化において最も重要なのが、シンプルなモデルの構築である。なぜならば、モデルの複雑さは組合せ数の増加に直結し、データ送信量と計算量の増加につながるためである。

図 2 は、「仕事をしている」という上位コンテキストをモデル化したベイジアンネットワークの例である。このモデルにおいて、上位コンテキストである Working は、Lights, Temperature, Movement, Sound, Time の 5 つの下位コンテキストの状態に影響を与えており、それらはそれぞれ 2 値で離散化されている。このモデルでの下位コンテキストの組合せは、 $32 (2^5)$

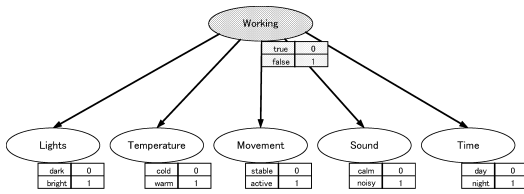


図 2 Working コンテキストを表すベイジアンネットワーク
Fig. 2 Bayesian Network for Working context.

通りある．各組合せにおける上位コンテキストの周辺確率分布は，あらかじめ学習した条件付き確率分布を用いてベイズ確率推論によって求める．条件付き確率分布は，収集したセンサデータ，およびユーザがタグ付けしたデータの比率から求められる．たとえば，ある組合せにおける Working コンテキストの周辺確率は次のように記述できる．

$$P(\text{Working} | \text{bright, cold, stable, calm, day}) \\ = \{80.2, 19.8\}$$

これは，下位コンテキストがそれぞれ bright, cold, stable, calm, day の状態のときに，「Working」である確率が 80.2%であることを示す．

3.3 推論結果の圧縮

ベイジアンネットワークのモデルは確率変数間の依存関係および条件付き確率分布から構成される．ベイズ推論から得る事後確率分布は，モデルが複雑になるに従い膨大な量になるため，センサノードへ送る際に推論結果を圧縮する必要がある．まず事後確率分布を格納するデータ構造の最適化について述べ，センサノードへの送信方法について説明する．

3.3.1 データ構造

ベイズ推論から得た事後確率分布を格納するデータ構造の最適化について述べる．事後確率分布を保存するテーブルを事後確率テーブルとし，上位コンテキストごとに作成する．表 2 は，Working コンテキストの事後確率分布の一部である．左列の下位コンテキストの組合せと右列の事後確率が対応する．Working コンテキストは，Lights, Temperature, Movement, Sound, Time の 5 つの下位コンテキストに依存している．下位コンテキストは，それぞれ 2 値の状態を持つ（たとえば Light の値は dark と bright）．そのため Working コンテキストの事後確率分布はヘッダの行を除いて 32 (2^5) 行必要となる．すなわち，下位コンテキストがそれぞれ 2 値の状態を持つ場合，下位コンテキストの数を n とすると事後確率分布のオーダは $O(2^n)$ で表すことができる．このように事後確率分布そのままの形では， $O(2^n)$ のデータ量をセンサノードへ送信する必要がある．したがって，事後確率

表 2 Working コンテキストの事後確率分布の抜粋

Table 2 A part of the marginal probability distribution associated with Working context.

Working コンテキスト					事後確率
Lights	Temperature	Movement	Sound	Time	true, false
dark	cold	stable	calm	day	0.30, 0.70
dark	cold	stable	calm	night	0.07, 0.93
dark	cold	stable	noisy	day	0.20, 0.80
dark	cold	active	calm	day	0.03, 0.97
dark	warm	stable	calm	day	0.30, 0.70
dark	warm	active	noisy	night	0.05, 0.95
bright	cold	stable	calm	day	0.92, 0.08
bright	cold	active	noisy	night	0.08, 0.92
bright	warm	stable	noisy	night	0.88, 0.12
bright	warm	active	calm	night	0.16, 0.84
bright	warm	active	noisy	day	0.08, 0.92
bright	warm	active	noisy	night	0.03, 0.97

分布を保存する段階で効率的なデータ構造を考える必要がある．

事後確率分布を事後確率テーブルとして保存する際に，データ量を削減するため「組合せインデックス」を導入する．組合せインデックスとは，事後確率分布における bright, cold, stable, calm, day といった下位コンテキストの組合せを，単一のインデックスに割り当てるものである．この組合せインデックスは，どのような上位コンテキストにも利用できて，必要なときに元の組合せに復元できる必要がある．

まず，上位コンテキスト番号と下位コンテキストを番号として定義する．たとえば，「Working」の上位コンテキスト番号を 0 と定義し，Lights, Temperature, Movement, Sound, Time の下位コンテキスト番号をそれぞれ 0, 1, 2, 3, 4 と定義する．次に，下位コンテキストの状態に離散値を設定する．Lights の場合，dark と bright 2 つの状態に対し，0, 1 という離散値を設定する（2 分割）．たとえば，下位コンテキストの 3 値の状態を持つ場合，0, 1, 2 と設定する（3 分割）．下位コンテキストの状態に対応するすべてのインデックスは，図 2 に示す．さらに，組合せの値に対応する数を最小ビット数にまとめることで，事後確率分布の各組合せをインデックスに割り当てる．最小ビット数は，下位コンテキストの分割数が 2^x または $2^x - 1$ のときに x ビット必要となる．

このアルゴリズムを，Working コンテキストの最初の行 (dark, cold, stable, calm, day) を用いて説明する．この場合，これらの下位コンテキストの状態値は 0, 0, 0, 0, 0 (図 2 参照) となる．これら下位コンテキストはすべて 2 分割のため，この値を格納するためそれぞれ 1 ビットが割り当てられる．つまり，表 2 の下位コンテキストの組合せを表すためには，各行ご

表 3 表 2 に対応する事後確率テーブルの例

Table 3 The combination indices corresponding to Table 2.

組合せインデックス	事後確率
0 (00000)	0.30
1 (00001)	0.07
2 (00010)	0.20
4 (00100)	0.03
8 (01000)	0.30
15 (01111)	0.05
16 (10000)	0.92
23 (10111)	0.08
27 (11011)	0.88
29 (11101)	0.16
30 (11110)	0.08
31 (11111)	0.03

表 4 組合せインデックスの要素

Table 4 The primitive contexts and their division number.

下位コンテキスト番号： 0, 1, 2, 3, 4
 分割数： 2, 2, 2, 2, 2

とに 5 ビットが必要となる（もしこれらの下位コンテキストの 1 つが 3 分割であった場合、6 ビット必要となる）。このように上記の組合せを、00000 の 5 ビットバイナリのインデックスとして表現できる。表 3 に表 2 に対応したインデックスを示す。この組合せインデックスは、下位コンテキストとその分割数がかかることで元の組合せに復元可能である。表 4 に Working コンテキストにおける、組合せインデックスの要素を示す。この組合せインデックスにより、 $O(2^n)$ のオーダであった事後確率分布を $O(n)$ まで圧縮できたことになる。

3.3.2 センサノードへの送信

基幹サーバは、アクチュエーションの起動を必要とするノードにベイズ推論の結果である事後確率分布を送信する必要がある。組合せインデックスの導入により、 $O(2^n)$ のオーダであった事後確率分布を $O(n)$ まで削減できたが、このままでは下位コンテキスト数の増加を補償するものではない。そこで次に、インデックスの選択送信を用いた送信データの削減を行う。

インデックスの選択送信とは、上位コンテキストの真となる確率が、任意に定めた閾値よりも高い列のみを選択しインデックスとしてノードへ送信する手法である。この方法は、アクチュエーションに必要な状況を満たした組合せに限り送信することになる。たとえば閾値が 0.8 だった場合、表 3 において 16 と 27 の 2 行のみを送る必要がある。これは、もし「Working」コンテキストの確率が 80% 以上のときにアクチュエータを起動することを意味する。このインデックスの選

択送信により、送信データのオーダを調整できる。実際のオーダは、コンテキストに依存するが閾値を調整することで制御できる。閾値を調整することで、テーブル行の多くを送信せずに済み、結果としてエネルギー消費を抑えセンサ/アクチュエータネットワークのライフタイムを延ばすことができる。加えて閾値の調整により、ユーザはセンサネットワークの感度を調整できる。ネットワークのセンサノードのいくつかは、センサデータにエラーが発生する場合がある。センサデータの不確実性をどの程度許容するかを閾値で調整できる。たとえばユーザが、こうした不確実性を除去したい場合、0.9 のような高い閾値を使うことができる。しかしながら、あまり高い閾値はコンテキストが成立しているにもかかわらず検知できない恐れがある。逆に不確実性を許容したい場合、0.6 といった低い閾値を使うことができる。この場合、実際には成立していないにもかかわらずコンテキストとして検知してしまう恐れがあるが、コンテキストを取りこぼす恐れは少なくなる。

基幹サーバは、組合せインデックスに加え、次の 3 つの情報をセンサノードに送る必要がある。(1) 上位コンテキスト番号：検知したい上位コンテキスト（「Working」など）、(2) 下位コンテキスト番号：上位コンテキストが推論時に使用する下位コンテキスト、(3) 各下位コンテキストの分割数。(2) および (3) は表 4 に示す。

また事後確率分布は、学習の進捗により動的に変化するため、事後確率テーブルを定期的送信し、更新する必要がある。事後確率テーブルを送信すべき間隔は、取得するコンテキストや学習の状況にもよるため一概にはいえない。事後確率分布の変化を、数量的に判断して定めた基準以上の変化が生じた場合にセンサノードに対し送信する仕組みが必要となる。この際、事後確率分布を送信するにあたっては、ユーザが明示的に指定している状況を想定し、所与のものとする。

3.4 切断、再接続の検知とアクチュエーション

アクチュエーションの主体が、サーバ側とノード側の間で切り替わるため各センサノードは基幹サーバとの切断と再接続を検知する必要がある。またセンサノードは、基幹サーバから配布された上位コンテキストの事後確率分布に基づいたアクチュエーション機構が必要となる。

3.4.1 センサノードにおける切断、再接続の検知

切断時と接続時で挙動が異なるため、センサノードの機能として切断、再接続の検知が必要である。切断、再接続を検知するには、シンクノードからの定期的なフ

ラッディングによる通知と経路制御との連動の2つの方法が考えられる。

シンクノードからの定期的なフラッディングによる通知は、各ノードが、シンクノードからの通知を受信することで接続を確認する手法である。各ノードは、通知がない場合タイムアウトにより切断状態を検知する。たとえば、基幹サーバが3秒ごとに通知パケットをフラッディングを行う。センサノードは、通知パケットを受信することでリセットされる9秒の受信タイムを持つ。通知パケットがセンサノードに3回届かないと受信タイムはタイムアウトし、切断状態へと移行する。この方法は、単純で実装が容易な反面、通信のオーバーヘッドが大きい。しかしながら、時間の同期など何らかの理由で定期的なフラッディングが必要となる場合、便乗してこの手法を採用できる。

シンクノードへのデータ送信時に信頼性のある経路制御を利用している場合、切断と再接続の検知に経路制御の情報を利用できる。具体例として TinyOS の標準的な経路制御である MintRoute¹⁶⁾ をデータ送信に利用している場合について考える。MintRoute は、多数のセンサノードから1台のシンクノードへデータ送信を行うことに特化したセンサネットワークの経路制御プロトコルである。センサノードは、定期的に制御パケットを送信し隣接ノードを管理するとともに、Distance Vector 方式でシンクノードへとつながる親ノードを決定する。したがって、MintRoute を利用した切断と再接続の検知を行う場合は、親ノードを参照しその有無を確認することで現在の状態を把握できる。この手法は、定期的なフラッディングといった余分なオーバーヘッドがない。しかしながら、MintRoute をはじめとする多くの信頼性のある経路制御では、ホップごとに切断検知のタイムアウトを行うためホップ数が大きくなると切断を検知するまでに時間がかかる。したがって、これら切断検知のメカニズムは、利用する環境に応じて適合する方式を採用すべきである。現状の実装では、MintRoute をデータ送信プロトコルとして用いているため、経路制御の情報を利用して切断と再接続を検知している。

3.4.2 コンテキストの検知とアクチュエーション

アクチュエータへの制御を実現するには、基幹サーバから配布される事後確率分布のほかいくつか必要なものがある。

まず、下位コンテキストごとにセンサデータの離散化のための閾値情報が必要である。これは、切断時にセンサノード上でセンサデータを下位コンテキストに変換する必要があるためである。このほか、上位コン

テキストごとに、送信先と送信すべき制御情報の定義が必要である。これらの情報は、学習の進捗により動的に変化する事後確率分布と比べ静的な情報であるため、各ノードが初めから持っていることを想定し、変更が必要な場合のみ送信する。

センサノードは、基幹サーバとの切断状態を検知すると、定期的にノードのコンテキストを検査する。図2の Working コンテキストを検知する場合を例として説明する。センサノードは Working コンテキストを構成する Lights, Temperature, Movement, Sound, Time のセンサデータを得る。次に bright, cold, stable, calm, day というように、閾値情報を元から取得したセンサデータを下位コンテキストへと変換する。検査するコンテキストが、複数ノードのセンサデータから構成される場合、変換した下位コンテキストを隣接ノードに送信し、隣接ノードのセンサデータを収集する。Working コンテキストでは単一のセンサノードから構成されているコンテキストであるため、他ノードのセンサデータの収集は行わない。収集した下位コンテキストから、1, 0, 0, 0, 0 (16) のように、現状の組合せインデックスを計算する。接続時に基幹サーバから同じ16という組合せインデックスが配布されている場合、センサノードは Working 状態だと判断しアクチュエーションが発生する。

4. 評価

PIMSANDO の有効性と実現可能性を実証するために、センサノードおよびアクチュエータノードとして、CrossBow の MICAz Wireless Mote³⁾ を採用し、プロトタイプの実装を行った。Mote では、次の機能を実装している。

- 音/光/温度/加速度/磁気/マイク/電力のセンシング
- センサデータをシンクノードへ送信
- センサデータを隣接ノードへ送信
- 切断、接続の検知
- 事後確率テーブルの送受信
- 事後確率テーブルに基づいたコンテキストの検知
- 制御情報の送受信
- アクチュエータとして LED, ブザーの制御

PIMSANDO は、事後確率分布を算出するための確率計算エンジンとして D-JENGA (図3) を用いる。D-JENGA は、ベイジアンネットワークの開発のための統合ソフトウェアツールキットである。D-JENGA は、確率推論アルゴリズムとして loopy belief propagation, logic sampling, likelihood weighting を提

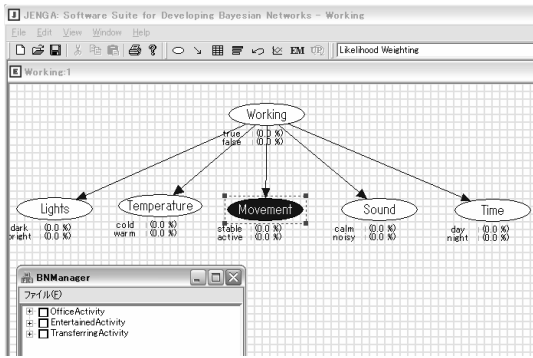


図 3 D-JENGA のスナップショット
Fig. 3 Snapshot of D-JENGA.

供する．また学習アルゴリズムとして Expectation-Maximization algorithm¹⁾ を提供している．これらのアルゴリズムは動的に変更可能であり，計算結果をテキストベースのクエリ言語としてネットワークに出力できる．

このプロトタイプにおいてどこまで複雑なコンテキストを扱うことが可能かを検証するため，コンテキストの複雑度に応じた送信量とコンテキストの複雑度に応じた解析時間の検証を行う．

4.1 コンテキストの複雑度に応じた送信量の検証

コンテキストの複雑度に応じた送信量について考える．本論文において，上位コンテキストは下位コンテキストの集合としてモデル化されているため，コンテキストの複雑度とは下位コンテキストの組合せ数を指す．

話を簡単にするため，今回の実験では，すべての下位コンテキストは 2 分割で表現しているものとする．したがって，上位コンテキストの複雑度は， 2^n で表現できる． n は，利用する下位コンテキストの数を表す．表 3 に示した事後確率テーブルをそのまま配布する場合，以下に示す情報が必要となる．

- (1) 事後確率テーブル
- (2) 宛先センサノードのアドレス
- (3) 上位コンテキスト識別子
- (4) 表 4 に示した利用する下位コンテキスト識別子と分割数

この場合，それぞれのデータサイズは 2^n byte，2 byte，1 byte， $2n$ byte となる．したがって，複雑度が 2^n の上位コンテキストにおける，送信すべきデータサイズは $2^n + 2n + 3$ byte である．たとえば，複雑度が $32 (2^5)$ の上位コンテキストでは送信すべきデータサイズは 45 byte となる．TinyOS の Active message を利用したデータ送信では 1 フレームの Data size が

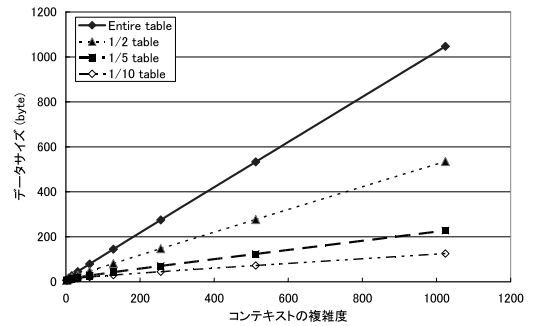


図 4 下位コンテキストの組合せ数とデータサイズの関係
Fig. 4 Complexity vs. data size.

29 byte と規定されているため，この場合は 2 フレームに分割して送信する必要がある．

基幹サーバ側でインデックスの選択送信を行うことで，事後確率テーブルのサイズは調整できる．事後確率テーブルのサイズを 5 分の 1 削減した場合（たとえば，閾値を 0.8 に設定した結果として），送信すべきデータサイズは $2^n/5 + 2n + 3$ byte となる．たとえば，複雑度が $32 (2^5)$ の上位コンテキストでは 17 byte となり，1 フレームで送信可能である．

図 4 は，事後確率テーブルを制限しない場合，インデックスの選択送信によりそれぞれ 1/2，1/5，1/10 に制限した送信すべきデータサイズと下位コンテキストの組合せ数の関係である．この図からインデックスの選択送信を用いた事後確率テーブルの削減の効果は，コンテキストの複雑度が増すにつれ大きくなることが分かる．要求するコンテキスト表現の自由度や電力効率を考慮してインデックスの選択送信の閾値を設定する必要があるといえる．

図 4 のデータサイズは，基幹サーバからセンサデータへ送信するデータ量であるとともに，センサノードで管理する情報量でもある．一方 DLDP，CLDP の場合，目的とするコンテキストを構成する確率変数の状態数分の事後確率を管理する．たとえば，図 2 の Working コンテキストの場合，10 エントリの事後確率を管理する．したがって，CLCP の管理する必要がある情報量は DLDP，CLDP と比較して大きくなる傾向があるが，選択送信により事後確率を削減することで，管理する情報量を抑えることができる．

4.2 コンテキストの複雑度に応じた解析時間の検証

Mote 上で事後確率テーブルを処理し，コンテキストを検知するのにかかる時間を測定した．Mote 側では，解析処理としてまず確率テーブルの組合せインデックスから下位コンテキストの組合せを復元し現状の下位コンテキストと比較している．Mote 上では細

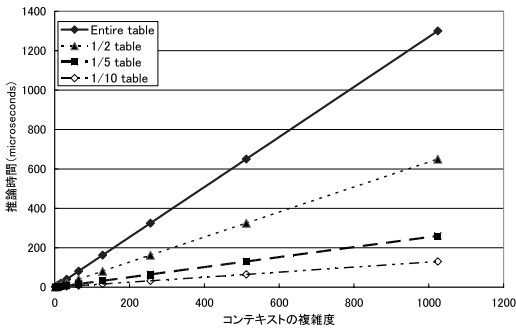


図5 下位コンテキストの組合せ数と計算時間の関係
Fig. 5 Complexity vs. inference overhead.

かい時間粒度での測定が難しいため、解析が終了しだいシリアルポート経由で終了信号を送信し、PC側での測定を行った。また事後確率時間の解析を行った測定結果から、解析を行わない場合の測定結果を引くことで、解析にかかった時間を概算している。それぞれの実験を1000回ずつ測定を行い、平均を求めた。

確率テーブルすべてを解析した場合、閾値により2分の1に削減した場合と5分の1に削減した場合の計算時間とコンテキストの複雑度の関係を図5に示す。この実験の結果からも、図4の結果と同様に閾値による計算量の削減が見てとれる。下位コンテキストの組合せ数が1024 (2^{10})のときに事後確率テーブルの全エントリを処理した場合でも1.3ミリ秒で終了している。

4.3 センサデータ収集および推論時の消費電力

ネットワーク切断時の通信コストと計算コストを明らかにするため、センサデータ収集および推論時の電力消費量を評価する。まず、通信コストとして複数センサノードからセンサデータを収集する際に消費する電力について考える。次の条件のもとで算出した通信の電力消費量を図6に示す。

- 15秒ごとに、下位コンテキストに変換したセンサデータ(36バイトのTinyOSフレーム)を隣接ノードへ送信
- コンテキスト検知を実行する1ノードのみが受信
- 理想的な状況下での通信を想定
- Micazの無線CC2420(250kbps)の消費電力
 - 送信: 17mA(送信電力1mW時)
 - 受信: 19.7mA
 - スリープ: 0.001mA

図6のX軸はセンサデータを収集するセンサノード数、Y軸はコンテキストを検知するセンサデータ収集ノードとセンサデータ送信ノードが15秒おきに消費する電力量をそれぞれ示す。センサデータ送信ノ

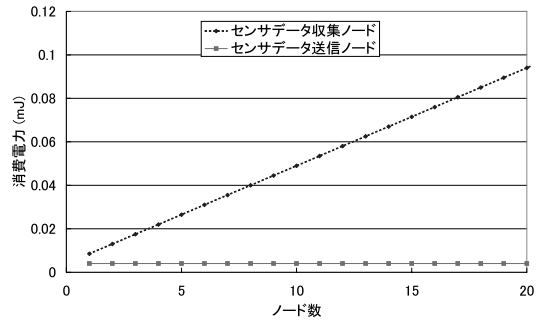


図6 センサデータ収集時の消費電力
Fig. 6 Energy consumption of data collection.

ドの消費電力は、ノードの増加に関係なく0.004mJで一定している。センサデータ収集ノードの消費電力は、受信電力量が増えるためノードの増加に比例して大きくなる。20台のセンサデータを収集する場合、0.09mJの電力を消費する。上述した条件で述べたとおり、この評価で算出した電力消費量は、理想的な環境で、かつすべてのセンサノードがシングルホップで届く隣接範囲にあることを前提としている。したがって、複数ホップ先のセンサノードからデータを収集する場合や、パケットロスの発生を考慮すると図6で示した電力消費量よりも大きくなると考えられる。

次にネットワーク切断時の計算コストとして、DLDP、CLDPのようにセンサノードで確率推論計算を実行する場合とCLCPのように確率推論計算は基幹サーバで行う場合の電力消費量について考える。センサノード上での確率推論計算はD-JENGAが提供するLikelihood Weighting⁸⁾を用い、サンプリング回数は2000回とする。消費電力を算出するにあたり必要となる、双方に共通する条件を以下に示す。

- 15秒ごとに上位コンテキストを検査
- 推論に用いる上位コンテキストの複雑度は1024で固定
- 処理時以外はスリープ状態
- MicazのMCU ATmega128Lの消費電力
 - 処理時: 12mA
 - スリープ時: 0.01mA

上記条件において、センサノードで確率推論計算を実行する場合と確率推論計算を基幹サーバで行う場合の15秒ごとに消費する電力は、それぞれ131.8mJ、0.047mJとなる。図7は、ネットワーク切断時の通信コストと計算コストをあわせた電力消費量を示すものである。X軸はセンサデータを収集するセンサノード数、Y軸はコンテキストを検知するセンサノードが15秒おきに消費する通信コストと計算コストの合計

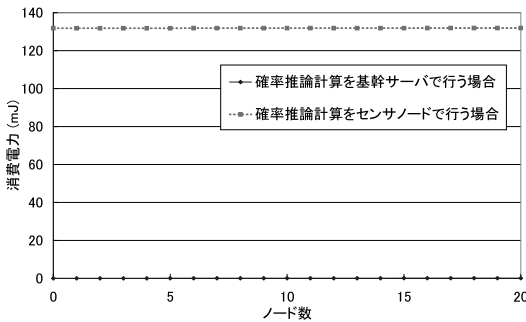


図 7 ネットワーク切断時の消費電力

Fig. 7 Energy consumption in a disconnected network.

の電力量をそれぞれ示す。

図 6, 7 から, 確率推論の計算コストが非常に大きいことが分かる。通信コストはセンサノード数と比例して増加するが, 確率推論の計算コストと比較すると問題にならないほど小さい。この結果から, 確率推論計算をセンサノードで行う DLDP, CLDP と比較して, CLCP は電力消費量の面で優れているといえる。

5. 関連研究

コンテキスト情報を利用したアプリケーションは, 数多く提案されている。Chen らは, これらコンテキストウェアアプリケーションの分類を行っている²⁾。コンテキストウェアの研究の多くは, センサ情報からのコンテキストのとり方や, コンテキストのモデル化手法に焦点が当てられている^{5), 14)}。しかしながら, ネットワーク切断に対する対応を考えていないため, たとえばセンサネットワークのシンクノードが故障してしまうとデータの解析, コンテキストの抽出, アクチュエータに対する制御といったすべての機能が停止してしまう。本研究では, こうした切断時における機能の継続性を目的とする。

ネットワークの切断性を背景とする研究は数多くある。Coda File System⁹⁾ は, ネットワーク切断時のファイルシステムの一貫性の保持を目的とし, Disconnected Operation という概念を創出した。ネットワーク分野では, 2.1 節であげた Delay-Tolerant Network⁷⁾, アドホックネットワークを対象とした Message Ferry¹⁷⁾ などがある。これらは, 断続的な長期間のネットワーク切断を背景として持つが, 再接続時にデータを集めることやデータを宛先に届けることが課題である。センサ/アクチュエータネットワークにおいては, センシングとアクチュエーションのサイクルが, つねに 1 組の動作として行われるためにデータをバックボーンサーバに配送するだけでは不十分であ

る。これに対し本論文では, データの解析とアクチュエータの制御に重点を置き, 耐故障性を実現するセンサ・アクチュエータネットワークを構築することが目的である。

センサ/アクチュエータネットワークの制御部分に重点を置いている研究としては, 文献 4) やモバイルロボットを対象とした RAMOS¹⁵⁾ がある。これらに対し本研究では, ベイジアンモデルを利用したネットワーク内でのコンテキスト検知に重点を置いている。

6. 結論と今後の課題

本論文では, ベイジアンモデルを使った推論と推論の結果に基づく制御を, ネットワーク内で実現する機構として PIMSANDO の提案を行った。PIMSANDO を用いることで, ノードの移動や, 突発的な故障によりネットワーク切断が生じて, 一時的に推論とアクチュエータへの制御をセンサノード側で実行できる。

本機構のデザインとして推論部分の代替手法, 推論結果の削減手法, センサノードにおける切断, 再接続の検知とアクチュエーション方法について説明した。Berkeley MICA-Mote 上で本機構のプロトタイプを実装し, コンテキストの複雑度に対する送信データ量と計算時間に関する検証を行い, 本機構の実現可能性と有効性を実証した。

今後の課題として, よりフレキシブルなネットワーク内アクチュエーションを考える。現在のデザインでは, 制御するセンサノードと, アクチュエータノードは 1 対 1 の関係を前提としている。今後は, 制御するノードとアクチュエータの関係を 1 対 N , N 対 1, N 対 N に拡張できる仕組みを考える必要がある。

参考文献

- 1) Bilmes, J.A.: A gentle tutorial of the em algorithm and its application to parameter-estimation for Gaussian mixture and hidden Markov models, Technical report, U.C. Berkeley (1998).
- 2) Chen, G. and Kotz, D.: A survey of context-aware mobile computing research, Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College (Nov. 2000).
- 3) Crossbow, MICAz ZigBee Series (MPR2400). <http://www.xbow.com/Products/productsdetails.aspx?sid=101>
- 4) Deshpande, A., Guestrin, C. and Madden, S.: Resource-Aware Wireless Sensor-Actuator Networks, *Proc. IEEE Data Engineering Bulletin* (2005).

- 5) Patterson, D.J., Liao, L., Fox, D. and Kautz, H.: Inferring high-level behavior from low-level sensors, *Proc. Ubicomp 2003* (Oct. 2003).
- 6) Estrin, D. and Govindan, R.: Next century challenges: scalable coordination in sensor networks, *Proc. ACM MOBICOM 1999*, pp.263–270 (Aug. 1999).
- 7) Fall, K.: A Delay-Tolerant Network Architecture for Challenged Internets, *Proc. ACM SIGCOMM 2003* (Aug. 2003).
- 8) Fung, R. and Chang, K.: Weighting and integrating evidence for stochastic simulation in Bayesian networks, *Proc. 5th Conference on Uncertainty in Artificial Intelligence* (1989).
- 9) Kistler, J.J. and Satyanarayanan, M.: Disconnected operation in the Coda file system, *Proc. 13th ACM Symposium on Operating Systems Principles* (Feb. 1992).
- 10) Li, Q. and Rus, D.: Sending Messages to Mobile Users in Disconnected Ad-hoc Wireless Networks, *Proc. ACM MOBICOM* (Aug.2000).
- 11) Liston, J.: Pushpin Computing: A Platform for Distributed Sensor Networks, Master's thesis, Massachusetts Institute of Technology (2002).
- 12) Ranganathan, A. and Campbell, R.H.: A Middleware for Context-Aware Agents in Ubiquitous Computing Environments, *Proc. ACM Middleware'03* (2003).
- 13) Sohrabi, K., Gao, J., Ailawadhi, V. and Potti, G.: Protocols for self-organization of a wireless sensor networks, *Proc. IEEE Personal Comm. Magazine*, Vol.7, No.5, pp.16–27 (Oct. 2000).
- 14) Sparacino, F.: Sto(ry)chastic: a Bayesian network architecture for user modeling and computational storytelling for interactive space, *Proc. Ubicomp 2003* (Oct. 2003).
- 15) Suzuki, R., Makimura, K., Saito, H. and Tobe, Y.: Prototype of a Sensor Network with Moving Nodes, *Proc. 1st International Workshop on Networked Sensing Systems (INSS 2004)*, Tokyo (June 2004).
- 16) Woo, A., Tong, T. and Culler, D.: Taming the Underlying Challenges of Reliable Multi-hop Routing in Sensor Networks, *Proc. ACM SenSys 2003* (Nov. 2003).
- 17) Zhao, W., Ammar, M. and Aegura, E.: A Message Ferry Approach for Data Delivery in Sparse Mobile Ad Hoc Networks, *Proc. ACM MobiCom'04*, pp.187–198 (Sep. 2004).

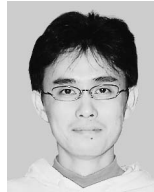
(平成 18 年 5 月 19 日受付)

(平成 18 年 11 月 2 日採録)



間 博人 (学生会員)

現在, 慶應義塾大学大学院政策・メディア研究科後期博士課程在学中。センサネットワーク, 無線通信, 通信プロトコル等の研究に従事。IEEE 学生会員。



門田 昌哉

現在, 慶應義塾大学院政策・メディア研究科後期博士課程に在学中。ベイジアンネットを用いた小型無線センサノード上での並列分散な確率推論機構の研究に従事。



中澤 仁 (正会員)

博士 (政策・メディア)。慶應義塾大学大学院政策・メディア研究科講師。日本ソフトウェア科学会, ACM 各会員。ミドルウェア, システムソフトウェア, ユビキタスコンピューティング等の研究に従事。



徳田 英幸 (正会員)

慶應義塾大学より工学修士。カナダ, ウォータールー大学より Ph.D. (Computer Science)。現在, 慶應義塾大学大学院政策メディア・研究科委員長, 同大学環境情報学部教授。分散リアルタイムシステム, マルチメディアシステム, 通信プロトコル, 超並列・超分散システム, モバイルシステム等の研究に従事。IEEE, ACM, 日本ソフトウェア科学会各会員。