

平成 31 年 4 月 30 日現在

機関番号：14401

研究種目：基盤研究(C) (一般)

研究期間：2016～2018

課題番号：16K00125

研究課題名(和文) モデル・情報の不完全な環境下における自己モデル化を用いたネットワーク制御手法

研究課題名(英文) Network Control based on self modeling for the case of uncertain model and information

研究代表者

大下 裕一 (Ohsita, Yuichi)

大阪大学・情報科学研究科・准教授

研究者番号：80432425

交付決定額(研究期間全体)：(直接経費) 3,500,000円

研究成果の概要(和文)：近年、ネットワーク機能の仮想化技術の進展や、IoT機器と呼ばれるネットワークに接続される機器の多様化により、ネットワークに接続される制御される対象が拡大している。そのような機器の中には、制御対象の正確なモデル化が困難なものも存在する。この問題に対して、本研究課題では、(1)制御対象を制御しながら、制御対象や把握される情報に含まれる誤差のモデル化を行う手法、(2)制御対象が数学的にモデル化できない場合であっても、適切な制御入力を求める手法を確立した。これにより、制御対象があらかじめ正確にモデル化できず、得られた情報が不正確であっても適切な制御を行うことが可能となる。

研究成果の学術的意義や社会的意義

本研究では、正確なモデル化が行われていない機器について、当該機器を実際に操作しながらモデルを構築し、操作を行い、さらに、そのモデル化をタスク実行の初期段階で完了する手法を確立している。この手法は、複雑な機器や機器の制御誤差を考慮しながら制御を行うような機器をネットワーク経由で制御すること活用することができると考えられ、ネットワーク経由で行われる制御の範囲の拡大に寄与できると考えられる。

研究成果の概要(英文)：Many devices are becoming operated via network. Some of them are not accurately modeled. As a result, it is difficult to calculate suitable input for them. In this research, we propose a method to control such devices. We propose (1) a method to model the control errors of the devices by operating and monitoring the device, and (2) a method to obtain a suitable input for the device.

研究分野：情報ネットワーク

キーワード：ネットワーク 制御 バイズ推定 進化計算

1. 研究開始当初の背景

近年、ネットワーク機能の仮想化技術の進展や、IoT 機器と呼ばれるネットワークに接続される機器の多様化により、ネットワークに接続される制御される対象が拡大している。その一方で、ネットワークや制御対象の状態といった制御に必須な情報を正確に把握することは困難であるのみならず、制御対象の正確なモデル化も困難であり、どのような制御を行ったら、どのような状態に達するのかといったことも事前に把握することは困難である。そこで、本課題では、このような、状態の正確な把握も、制御対象の正確なモデル化も困難な環境下において、適切な制御を実現するための手法を検討する。

2. 研究の目的

本研究課題では、制御に必要な現在の状況把握、制御対象の正確なモデル化が困難な環境下においても、適切な制御を実現するネットワーク制御手法を確立することを目的とする。

3. 研究の方法

本研究課題の目的を達成するにあたり、本研究では

(1) 制御対象を制御しながら、制御対象や把握される情報に含まれる誤差のモデル化を行う手法

(2) 制御対象が数学的にモデル化できない場合であっても、適切な制御入力を求める手法を確立する。これにより、(1)のモデル化と(2)の制御を組み合わせることにより、制御対象があらかじめ正確にモデル化できず、得られた情報が不正確であっても適切な制御を行うことが可能となる。

4. 研究成果

(1) 制御対象や把握される情報に含まれる誤差のモデル化手法

本項目では、制御入力と結果の対応が正確にモデル化できない状況においても、制御対象の状況をモデル化しながら、意図通り制御対象を制御する仕組みについて検討を行う。本検討では、ネットワーク経由で操作される移動ロボットを対象として検討を行った。ネットワーク経由で行うロボット制御では、ネットワークの遅延を考慮して制御することが必要となる。すなわち、コントローラにおいて、制御コマンドが到達する将来のロボットの状況を予測し、予測した将来のロボットに対して適した制御コマンドを計算し、制御コマンドをロボットに向かって送出する。この手順においては、将来のロボットの状況の予測精度が、制御性能に大きな影響を及ぼす。将来のロボットの予測は、それまでに送出した制御コマンドをもとに、その制御コマンドにより制御が行われた結果、ロボットが到達する状態を予測する形で行われるが、車輪がスリップする、遅延のゆらぎにより制御コマンドの到達が遅れる等により、将来のロボットの状態が意図通りの状態に到達しているとは限らない。その結果、予測した状態と実際の状態の誤差が大きく異なり、意図通りの制御ができない。

本研究では、上記のようなネットワーク・ロボットの動作とともに不安定で、制御に誤差が生じる環境下において、ロボットの環境を同定し、制御に用いる手法について検討する。本研究では、不安定な環境下におけるロボットの状況の推定は、確率モデルで表す。この確率モデルは、遠隔制御ロボットを制御しながら、行った制御とその制御後に遠隔制御ロボットが到達する状態から求めることができる。しかし、この方法では、行った制御、制御後に到達したロボットの状態に関する情報が少ない、タスク実行初期の段階では、適切な環境同定ができないといった問題や、ロボットを制御する上で未経験の突発的な大きな誤差を考慮することができず、適切な制御を行うことができないという可能性がある。

本研究では、上記のような遠隔操作ロボットに対して、得られた観測結果が少ないタスク実行初期の段階であっても、適切な制御が可能となるような環境同定手法を提案する。文献 [1] では、生物が少ない経験であっても、適切な認知を行うことができる仕組みを持つように進化をすることが示唆されており、進化の結果、自身が持つ事前分布をもとに観測情報を用いてベイズ推定を行い、発生しうる状況を認知するのに適した事前分布を持つような進化が行われる可能性を明らかにしている。そこで、機器の遠隔制御においても、ベイズ推定の認知を行うとともに、その事前分布を遺伝的アルゴリズムにより得ることにより、少ない観測情報しか得られていないタスク実行初期の段階であっても、現在の環境下で生じる誤差を適切に予測し、制御を行うことが期待できる。

以降、上記のベイズ推定にもとづく誤差のモデル化手法と、遺伝的アルゴリズムを用いた事前分布の定め方について述べる

誤差のモデル化手法：

本研究では、遠隔制御ロボットを用いたタスクを実行しながら、行った制御とその制御後に遠隔制御ロボットが到達する状態をもとにロボットの移動距離あたりに含まれる制御誤差の確率分布 $P(\cdot)$ を推定することにより、現在、ロボットが動作する環境の同定を行う。タスクを実行しながら、 $P(\cdot)$ の推定を行う場合、これまでの観測結果には含まれないような大きな誤差が生じる可能性についても考慮することが求められ、このような場面においても、ロボットに対

して、誤差の影響が大きくなってしまふような不適切な制御入力を与えることは防ぐ必要がある。

本研究では、ベイズ推定により、 $P(\theta)$ の推定を行う。これにより、事前分布を適切に定めることができれば、観測結果が不十分な場合であっても、不適切な制御入力を与えてしまうことを避けることができる。

本研究における、コントローラが $P(\theta)$ を逐次推定する手順を以下に示す。

手順 1. ロボットから受信した観測値 $A(t-1)$ と直近の制御入力をもとに、現在の時刻 t における誤差 $e(t)$ を求める。

手順 2. 直近 K タイムスロットの $(t-K)\dots (t-1)$ をもとに、 $P(\theta | A(t-K:t-1))$ をベイズ推定 $P(\theta | A(t-K:t-1)) = P(\theta | (t-K:t-1)) P^{\text{prior}}(\theta)$

ただし、 $P^{\text{prior}}(\theta)$ は事前に定めた事前分布である。

進化計算による事前分布の算出：

ベイズ推定を用いた環境同定では、事前分布の影響が大きく、事前分布を適切に定めていないと、誤差の影響が大きくなるような不適切な制御入力を与えてしまうことも考えられる。そのため、タスクを実行しながら、ベイズ推定により環境の同定を行う手法においては、適切な事前分布を定めることが必要となる。生物が進化の過程でベイズ推定における適切な事前分布を獲得した可能性が示唆されている。そこで、本稿においても、遺伝的アルゴリズムにより、事前分布を定めるものとする。

本研究では、 $P^{\text{prior}}(\theta)$ の分布の形状は既知であるとし、そのパラメータを進化させる。具体的な手順は、以下の通りである。

(1) 初期化： $P^{\text{prior}}(\theta)$ の分布を定めるパラメータをランダムに設定与えた個体を N 個生成する。

(2) 評価：各個体について、当該個体が保持するパラメータの $P^{\text{prior}}(\theta)$ を事前分布として持つコントローラにより、移動ロボットを遠隔制御するタスクを想定しうる環境下で実行した場合の挙動をシミュレーションにより得る。そして、そのタスクの正確性、速度といった指標で、各個体を評価する。

(3) 選択・交叉・突然変異： 評価値をもとに選択、交叉、突然変異を行い、新たな個体を生成する。

(4) 2 へ戻る

上記の手順では、適切にタスクを実行するのに有用な事前分布を持つように選択・交叉が行われる。そのため、上記の手順による進化を行うことで、適切な事前分布を得ることができると考えられる。

評価：

本研究では、上記の手法を対向二輪型ロボットのネットワーク経由での制御に応用した場合についてシミュレーションで評価を行った。対向二輪型ロボットでは、左右の車輪の回転速度を入力として与えることにより、移動・回転を行う。本制御では、ロボットの移動にかかる誤差は、左右の車輪の回転数に比例した誤差が生じるものとし、ロボットの精度を確保するために、精度が一定以下となるように速度の上限を定めて、制御をするものとした。

本シミュレーションにおいては、10ms ごとにコントローラはロボットに送出する指示を計算、ロボットは自身の座標、向きの情報をコントローラに送出するものとする。また、簡単のため、ロボットが把握している自身の位置は正確であるものとする。ロボット・コントローラ間には、往復遅延が存在するものとし、遅延時間は平均 100ms、分散 0.05 および 0.25 の正規分布に従う環境で動作させた。また、本シミュレーションにおいて、各車輪が進む速度は、スリップ等の影響により、制御で投入された速度に、平均 1、標準偏差が 0.05 から 0.2 の正規分布に従う値を乗じた速度になるという環境下で動作するものとした。

本シミュレーションでは、座標位置 (0,0) から (5000,0) までの直線を移動するタスクを課した。本シミュレーションの環境では、ロボットの左右の車輪の移動距離に誤差が生じる。そのため、大きな誤差が生じた際には、ロボットの角度がずれ、意図した直線から離れた箇所を経由する。そのため、本シミュレーションでは、本タスクにおいて、実際にロボットが経由した軌跡の Y 軸からのずれに注目した。

本評価では、以下の手法を比較する。

提案手法：遺伝的アルゴリズムにより求めたパラメータを持つ事前分布と、直近 20 タイムスロットの誤差の観測地をもとに、ベイズ推定により誤差を把握する。本評価においては、各世代 50 個体を生成、50 世代進化を行った結果得られたパラメータをもつ事前分布を初期事前分布として与えた。

統計的手法：事前分布を用いずに、制御中に得られた直近 20 タイムスロットの誤差を用いて誤差を学習し、制御に用いる。本手法では、制御開始時点では、誤差についての情報を持たず、制御を行いながら、制御結果をもとに左右の車輪の制御に含まれる誤差を計算し、その平均、分散を計算する。そして、正規分布に従う想定して、ロボットの状態の予測・制御を行う。本手法と比較することにより、事前分布を用いて、逐次ベイズ推定により現在の環境で生じる誤

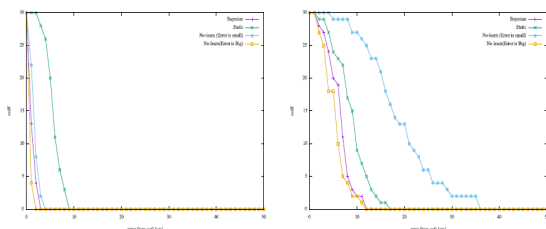
差を推定することの効果を確認する。

固定の誤差モデルを用いる手法：制御開始前に、ロボットの移動時に生じる誤差のモデルに関するパラメータをすべて固定値として与える。具体的には、平均0の正規分布（ただし、分散はパラメータとして事前に与える）としてモデル化し、このモデルを用い、ロボットの状態の予測・制御する。本評価では、 $\sigma=0.05$ 、 $\sigma=1.0$ の2種類のモデルを用いた。本手法との比較により、誤差モデルのパラメータを環境に合わせて学習することの効果を確認する。

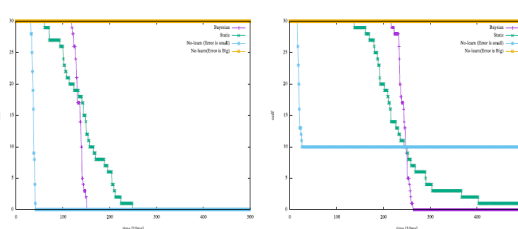
本評価では、車輪の移動距離の誤差の分散値がそれぞれ0.025、0.04の環境において、各制御手法でロボットを制御するタスクを30回行い、各タスクの試行において、目標軌跡からのずれの最大値を調べた。図1a、1bに、その累積補分布を示す。図より、いずれの環境下においても、ベイズ推定により誤差を推定する手法は、目標軌跡からのずれを小さく抑えることができていることが分かる。

図2a、2bに、各手法を用いて制御を行った際のタスク完了までにかかる時間の累積補分布を示す。横軸はタスク完了までにかかる時間を表しており、縦軸はロボットの累積補分布を表す。図より、ベイズ推定を行う手法によるタスク完了までの時間は、車輪の移動距離の誤差の分散値が小さい場合には、短い時間でタスクを完了することができることが分かる。

つまり、ベイズ推定を行うことにより、各環境の誤差を推定し、その環境にあった制御が可能であるといえる。



(a) 遅延の分散、車輪速度の誤差が小 (b) 遅延の分散、車輪速度の誤差が大



(a) 遅延の分散、車輪速度の誤差が小 (b) 遅延の分散、車輪速度の誤差が大

図1 経路からのずれの累積補分布

図2 制御の時間の累積補分布

(2) 制御入力を定める手法

上述の、制御対象や把握される情報に含まれる誤差のモデル化手法により得られる制御入力と誤差との関係は、数式等で表されたモデルと異なり、複雑なモデルとなることもありうる。そのため、制御を行う際には、入力と出力の対応が複雑なモデルでモデル化した場合であっても、適切な制御入力を求める手法が必要となる。

本検討では、複雑なモデルに対して、適切な制御入力を求める手法の検討にあたり、遅延、故障発生時の迂回経路の確保と電力の3つの性能を考慮に入れ、必要な遅延性能、故障発生時の迂回経路の確保の条件を満たしつつ、消費電力を最小化するネットワークの経路、スリープ制御を対象として検討を行った。

迂回経路を多く確保しようとするればオンになるノードやリンクが増加するため消費電力は増加し、性能を確保しようとしても消費電力は増加する、といったように、消費電力、信頼性、性能の指標は、ある指標を向上させると、別の指標が悪化してしまう密接な関係にある。本研究では、複数の指標を考慮して最適化を行うパレート最適制御をネットワークに適用する。パレート最適制御においては、多目的進化計算 [2] によりパレート最適解の集合（パレートフロント）を求める。本研究では、消費電力、信頼性、性能の3つの指標についてのパレートフロントを進化計算により求める。そして、SLAを保証できる範囲内で最も低消費電力な解をネットワークに投入する。

本制御では、集中制御コントローラは、定期的にネットワーク内のトラヒック観測情報を収集、各時刻のトラヒック状況に応じて、制御対象のネットワークを経路、機器の電源の投入状況を制御することにより、ネットワークの低消費電力化を行う。本計算では、毎回パレートフロント解を導出し、パレートフロント解で、ネットワーク管理者が定めた性能、信頼性の制約条件を満たすもののうち、最も消費電力が小さい解を投入する。これにより、ネットワークの性能、信頼性の条件を満たしつつ、消費電力を最小化することが可能となる。

以下に、各時刻において集中制御コントローラが行う手順を示す。

- (1) 環境取得：現在の環境を取得
- (2) 進化計算実行：パレートフロントを導出
- (3) 制御：2で得られたパレートフロントの中から、制約条件を満たす解をネットワークに投入

上記の手順のうち、進化計算によるパレートフロント解の導出は進化計算により行う。進化計算では、各個体に該当する解を遺伝子としてコーディングする。以下にアルゴリズム全体の流れを示す。

- (1) 初期化：N個の個体群Pの生成。
- (2) 評価：個体群Pの評価、非優越ソートによる各個体のランク付け実行、各ランクにおける各個体の密度計算。

- (3) 子の生成：遺伝子操作（選択、交叉、突然変異）を実行。N 個の子集団 Q を生成。
- (4) 世代交代：P、Q から解を選出し、N 個を P として構成。P とは別に、ランク 1 の個体をパレートアーカイブと呼ばれる領域に保存。
- (5) 終了判定：終了条件を満たした場合終了。この際、パレートアーカイブに保存されていた解が求まるパレートフロントとなる。終了でない場合は 3 へ。

上記の手順では、個体のランク付けに非優越ソート [3] を用いる。非優越ソートでは、ある個体において比較対象となる個体とどの評価値においても優れていることを「支配している」と呼び、各個体について、当該個体を支配している個体数を基準としてランク付けを行う。以下に非優越ソートの手順を示す。

- (1) n を 1 に初期化する。
- (2) 各個体に対して、当該個体が支配している個体の数と、当該個体を支配している個体の数を数える
- (3) いずれのランクのリストにも含まれていない各個体について、当該個体を支配している個体数を調べ、その値が 0 の場合は、ランク n の個体のリスト F_n に加える
- (4) F_n 中の各個体が支配している個体について、当該個体を支配している個体の数から 1 を引く

上記の手順では、各ステップにおいて、パレートフロントとなる解を複数保持、更新し、そのうち、実際に投入する設定を決定することにより、性能指標が複雑であっても、適切な制御入力を求めることができる。

ただし、大量の通信を行う機器間の組み合わせが変わるといった大きな環境変動が生じた際には、環境変動後のパレートフロントは、環境変動前のパレートフロントとは大きくかけ離れたものとなる。この場合、前の時刻のパレートフロントを初期解とした進化計算では、適切なパレートフロント解が求まるまで時間を要する。そこで、本研究では、前の時刻のパレートフロント解に加えて、様々な環境変動に対応可能だと考えられる多様な解を保存した解アーカイブを準備する。多様な解を準備することにより、このアーカイブには、環境変動発生時にパレートフロントに近い解も含まれている可能性が高くなる。その結果、環境変動発生時にも短時間で適切な解を発見することが可能となる。多様性維持のための解アーカイブは、パレートフロントを求める進化計算の各世代について、以下の処理を加えることにより、構築される。

- (1) 進化計算を 1 世代分行う
- (2) 現在の世代の解集合 P と、前の時刻に保存された解アーカイブ Q の和集合 P ∪ Q の各要素について、他の解との類似度とパレートフロント解からの性能指標の差の両面を考慮して評価を行う
- (3) 評価値が小さいものから順に解アーカイブ Q に保存

評価：

本評価では、典型的なデータセンター内ネットワーク構造である FatTree 型トポロジを用い、各スイッチあたりのリンク数を 8 とした。

本評価では、以下の 3 つの場合を比較する。

多様解保持あり (w/ DS ; with Diverse Solutions) : 本研究で提案する、前の時刻のパレートアーカイブと多様性の確保のための解アーカイブをもとに初期解を生成し、進化計算を行うことにより、パレート最適制御を行う手法。

多様解保持なし (w/o DS ; without Diverse Solution) : 前の時刻のパレートアーカイブを初期解とし、進化計算を行うことにより、パレート最適制御を行う手法。

ランダムな初期解 (R ; Random) : 毎回、ランダムに初期解を生成、進化計算を行うことで、パレート最適制御を行う手法。

図に結果を示す。図において、横軸はタイムスロット、縦軸は消費電力、遅延、確保された迂回経路数を示す。遅延、迂回経路数は、本制御にあたり、目標として投入した値に線を引いている。図より、w/ DS、w/o DS のいずれも、トラヒック量の変動に追従して、トラヒック量の少ない時間帯には、消費電力を抑え、トラヒック量の大きな時間帯には消費電力が高くなっていることが分かる。これは、トラヒック量が少ない時間帯は、少数の機器の電源を投入するだけで、性能要件を満たすことができるのに対し、トラヒックが多い時間帯には、多くの機器の電源を投入し、トラヒックを分散させないと、必要な制約を満たすことができないためである。w/ DS や w/o DS を用いた場合は、性能の確保を行った上で、消費電力を抑えるような経路を発見することができており、トラヒックの変動に追従したネットワーク低消費電力化を実現できている。しかし、初期解をランダムとして与えた R では、いずれの時間帯も同様の消費電力となっている。これは、ランダムな初期解では、49 世代までに、消費電力を削減できるような適切なパレートフロントが見つからないためである。このため、各時刻で短い世代数の進化計算によりパレートフロントを導出しつつ、環境変動に追従した低消費電力化を実現するためには、前の時刻のパレートフロントを用いることが有効であると考えられる。

また、遅延、確保された迂回経路数を比較すると、w/ DS を用いることで全ての時刻において、性能、耐故障の要件を満たす解を投入することができている。これは、w/ DS で求めたパレートフロントにおいて、性能、耐故障の要件を満たす解が存在し、本手法では、要件を満たす解からネットワーク設定を投入する解を選択するためである。w/o DS を用いた場合の時刻 175・

318・461 以降の性能において、制約を満たさず、解がほとんど改善されていない様子が見られる。時刻 117 付近のような環境では、トラヒック量が 0 に近い値に近づき、ほとんどのパレートフロント解が遅延の面で優劣のない解となってしまう。これにより、その後の進化計算で世代を重ねたパレートフロント解は、トラヒック量が増加した際に輻輳してしまう解が多くなり、制約を満たせる解がほとんど残っていない状況になる。その結果、進化をしようとしても、なかなか時刻 175・318・461 のようにトラヒック量が多い時間帯の制約を満たすような解に到達できない状況に陥る。よって、w/ DS のようにパレートフロント解のみだけでなく以前省電力になったような多様な解が残ることで、高速に適切なパレートフロントを取得し、よりトラヒックの増減に追従した制御が可能になると言える。

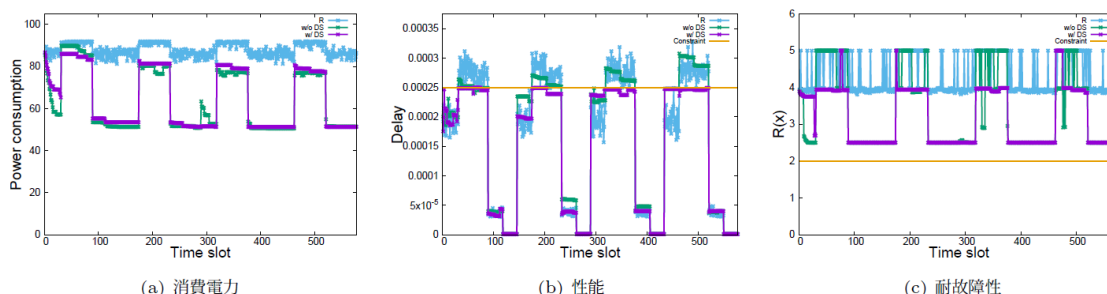


図3 各時刻における性能指標の値

参考文献

- [1] J.C. Ramirez and J.A. Marshall, "Can natural selection encode Bayesian priors?" *Journal of Theoretical Biology*, pp.57-66, Aug. 2017.
- [2] Kessaci, Yacine and Melab, Nouredine and Talbi, ElGhazali, "A pareto-based meta-heuristic for scheduling HPC applications on a geographically distributed cloud federation," *Cluster Computing*, vol. 16, pp. 451-468, Sep. 2013.
- [3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *Evolutionary Computation, IEEE Transactions on*, vol. 6, pp. 182-197, Apr. 2002.

5. 主な発表論文等

(研究代表者は下線)

[学会発表](計4件)

- (1) 松田拓己, 大下裕一, 村田正幸, "無線ネットワークを経由した移動ロボットの遠隔操作におけるベイズ推定を用いた環境同定手法," *電子情報通信学会技術研究報告(IN2018-16)*, pp. 21-26, Aug. 2018.
- (2) 秋下耀介, 大下裕一, 村田正幸, "パレート最適制御に基づく進化的ネットワーク省電力化手法," *電子情報通信学会技術研究報告(IN2017-113)*, pp. 141-146, Mar. 2018.
- (3) Yosuke Akishita, Yuichi Ohsita, and Masayuki Murata, "Network power saving based on Pareto optimal control with evolutionary approach," in *Proceedings of International Conference on Computing, Networking and Communications*, Jan. 2017.
- (4) 秋下耀介, 大下裕一, 村田正幸, "パレート最適制御によるネットワーク省電力化手法," *電子情報通信学会技術研究報告(IN2016-35)*, pp. 73-78, Jul. 2016.

6. 研究組織

研究代表者

大下 裕一 (OHSITA YUICHI)

大阪大学・大学院情報科学研究科・准教授

研究者番号：80432425

科研費による研究は、研究者の自覚と責任において実施するものです。そのため、研究の実施や研究成果の公表等については、国の要請等に基づくものではなく、その研究成果に関する見解や責任は、研究者個人に帰属されます。