Neurocomputing 126 (2014) 151–165

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Online fault detection of a mobile robot with a parallelized particle filter

Michał Zając*

University of Zielona Góra, ul. Podgórna 50, 65-246 Zielona Góra, Poland

ARTICLE INFO

Article history:
Received 2 January 2012
Received in revised form
16 September 2012
Accepted 28 November 2012
Available online 11 June 2013

Keywords:
Particle filter
Mobile robots
Fault detection
Parallel computing

ABSTRACT

Fault diagnosis is one of the most challenging problems, which have to be solved if one considers real-life applications of mobile robots. In this paper, we present a particle filtering-based approach combined with the negative log-likelihood test to address the fault detection task. The major disadvantage of the method is its high computational burden closely related to the number of particles used, which can be computationally too expensive to be processed online by the onboard computer of the robot. In order to address this problem, a solution, in which a part of computations are delegated to an external parallel computing environment such as a computer cluster, is presented. The proposed methods of parallelizing particle filters are aimed at improving their performance in terms of efficiency, estimation error and execution time, which are vital factors in an online setup. To depict the performance benefits of the presented methods, they are confronted with some other existing approaches in a series of experiments.

1. Introduction

1.1. Fault diagnosis of mobile robots

An acute need for uninterrupted, faultless and safe operation becomes obvious if we consider real-life applications of mobile robots. The problem is particularly significant for systems which operate in critical conditions with a high risk of fault occurrence [1,2]. This makes a reliable *fault detection* (FD) system an inherent part of a reliable mobile robotic system. *Particle filter* (PF), since its introduction by Gordon et al. [3], has proven to be a powerful method which can facilitate solving a large group of state estimation problems [4]. Its attractive properties, such as the ability of estimating states of highly nonlinear systems disturbed by a process noise, which can be described by virtually any probability distribution, make it also a powerful tool to use in *fault detection and isolation* (FDI) applications [5,6], particularly in the diagnosis of mobile robots [7–11].

A severe disadvantage of the particle filter is its computational complexity increasing with the number of particles N. Thus, to obtain satisfactory results, e.g., in the non-trivial case of a multi-dimensional problem, usually a high number of particles are required and it is hard to satisfy the time-constraints typical for real-time applications. However, the increase in the computational power of processors and the rapid development of multi-core/multiprocessor computing systems can effectively facilitate such solutions. The particle filter, due to its distributed nature, is an (almost) ideal candidate for being parallelized. Indeed, this form of enhancing its computational capabilities has been considered by the founders of the method already in 1993 [3], and has been developed by researches since then.

1.2. Related research

In general, we can divide the research related to this work into two groups. The first group consists of parallelization methods for particle filters utilizing diverse multi-core/multiprocessor environments such as CPU (*central processing unit*) clusters, GPUs (*graphics processing unit*), specialized FPGA/ASIC chips or hybrid architectures. However, parallelization of a generic particle filter in a naive way seems to be straightforward, due to its distributed nature, researchers put forward various techniques aimed particularly at optimizing execution time without degrading the overall performance of the algorithm.

The most commonly tackled problem is optimization of the global resampling procedure. A technique which eliminates the need of resampling by means of an appropriate particle exchange scheme is reported by Teuliére et al. [12]. Unfortunately, the work lacks a comparison with other (even naive) approaches, which makes any evaluation of the results provided by the author difficult. Bolić et al. [13–15] proposed a family of techniques for distributed resampling aimed at efficient hardware (ASIC) implementation of particle filters, namely *distributed resampling with proportional allocation* (DRPA) and *distributed resampling with non-proportional allocation* (DRNA). The former turns out to be a smart

* Tel.: +48 68 328 2422; fax: +48 68 328 4751.
E-mail address: M.Zajac@weit.uz.zgora.pl

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2012.11.049

implementation of systematic resampling, which does enable full parallelization but requires implementing rather complex particle routing mechanisms. The latter, by exploiting data parallelism, pipelining and decreasing global communication overhead, can achieve a high degree of parallelism. A thorough analysis of DRNA and comparison with other techniques is performed by Míguez [16]. The author also proposed also a method of local selection [17], which enables fine-grained parallelism, but was designed for the family of cost reference particle filters, and thus its applicability to a generic particle filter is not clear. An application of the DRNA-based particle filter to real-time identification of a complex mathematical model of physiology for clinical data interpretation, prediction and therapy optimization, together with a performance study of the implementation on a general-purpose CPU cluster was presented by Zenker [18]. Happe et al. [19] proposed a framework for multi-threaded implementation of the sampling importance resampling particle filter on a hybrid CPU/FPGA system, which can be relatively easy adapted to changing performance requirements and resource constraints when compared with other hardware implementations.

The ever increasing availability of multi-core GPUs results in big and cheap computing power being available on almost each desktop, which in turn can facilitate its applications in parallelization of particle filters, and can be an attractive alternative to custom hardware implementations. Hendeby et al. [20] present a complete GPU implementation of the PF, with a rasterizer unit employed to the stratified resampling procedure, and the solution turns out to outperform the CPU implementation, while achieving the same accuracy. Another GPU-based implementation, employing the CUDA architecture, was put forward by Chao et al. [21]. Their solution, the *finite-redraw importance-maximizing* (FRIM) prior editing, employs a combination of n-fold prior editing [3] with localized resampling to reduce the global communication overhead and achieve a high level of parallelism.

The second group of works related to this paper concerns research on PF-based FDI, either directly aimed at obtaining real-time diagnosis of mobile robots or employing PF state estimation methods which can be particularly interesting from the point of view of online mobile robot diagnosis. De Freitas et al. [7] proposed a Rao–Blackwellized particle filter-based solution for real-time fault diagnosis, where exploiting a one-step look-ahead strategy seems to provide superior performance in applications, when compared with a standard Kalman filter-based approach. In turn, an algorithm closely related to the Rao–Blackwellized particle filter for non-linear system diagnosis, the Gaussian particle filter, is presented by Hutter and Dearden [8]. In this method each particle samples a discrete mode and approximates the continuous variables by a multivariate Gaussian distribution at each time step by means of an unscented Kalman filter. Although the simulation results presented by the authors are promising, their paper lacks the results of tests on a real system. Another approach in several variants – a multiple particle filter for fault diagnosis of a mobile robot – was put forward and investigated by Cai and Duan et al. [22–24], and the method shows a good performance in real-robot experiments.

A number of interesting particle-based FDI techniques for mobile robots, complementary to the Gaussian particle filter, such as the variable-resolution particle filter, the risk-sensitive particle and the one-step look-ahead particle filter are introduced by Verma et al. [9]. The ideas presented by authors seem to work well in simulations, but extensive real-life tests are not provided. Another method is the Gaussian process proposals particle filter as put forward by Plagemann et al. [11]. A series of real robot experiments demonstrate that the developed system is able to track the state of the robot more reliably through collision events than an optimized version of the standard particle filter with uninformed proposals. A disadvantage of the system is that it deals only with binary failure variables. Li and Kadirkamanathan [25] formulate the FDI task in a multiple model environment where a particle filter is combined with the likelihood ratio test. This approach is further developed by the authors in their works [5,6] to account for abrupt and incipient faults in the detection and isolation phases. Unfortunately, the approach is investigated only on a simulation benchmark problem, and no results have been reported regarding a real system. A set of nonstandard applications of particle methods to change detection, system identification and control in nonlinear non-Gaussian state space can be found in the survey paper by Andrieu et al. [26].

1.3. Our results

In this paper, we introduce techniques which are intended to improve the performance of a parallelized particle filter, and then we verify their applicability in an online fault diagnosis system, since the topic has not been widely discussed in the literature yet. Although we illustrate our considerations with a fault detection problem, we presume that the applicability of the proposed methods is not limited only to this exemplary class of applications, and can be successfully applied to other state estimation problems. The contribution of this paper is fourfold. First, we propose a performance measure – the node divergence index – for evaluation of particle filter performance in a distributed computing environment. Second, we put forward a novel prior editing method – the finite-redraw low importance maximizing – which combines the most useful properties of the discussed existing approaches. Third, we present a parallelization scheme for particle filtering, which incorporates the proposed prior editing and local particle exchange steps into one filtering framework. Fourth, we provide a practical implementation and evaluation of a particle filter-based FD system for a mobile robot in a distributed environment, based on the prior research of the author concerning the FDI systems for mobile robots [27,10].

2. Material and methods

2.1. Particle filtering

Consider the class of dynamic systems whose dynamics can be described by a general discrete time nonlinear state space model [28]

$$\boldsymbol{x}_t = \boldsymbol{f}(\boldsymbol{x}_{t-1}, \boldsymbol{u}_t, \boldsymbol{s}_{t-1}),$$
$$\boldsymbol{y}_t = \boldsymbol{h}(\boldsymbol{x}_t, \boldsymbol{z}_t), \tag{1}$$

where $\boldsymbol{x}_{t-1}$ and $\boldsymbol{x}_t$ represent respectively the state vector at subsequent time instants $t-1$ and t with a known initial probability density function $p(\boldsymbol{x}_0)$, $\boldsymbol{u}_t$ is the control vector at time t, $\boldsymbol{f}(\cdot)$ and $\boldsymbol{h}(\cdot)$ are respectively nonlinear transition and measurement functions, and $\boldsymbol{s}_{t-1}$ and $\boldsymbol{z}_t$ represent respectively the process and measurement noise with known probability density functions.

Particle filters represent the system state as a set of samples from the state space. The density of samples in a given region represents the probability that the system state is just in that area of the state space. If we denote by $\boldsymbol{x}_{0:t} = \{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_t\}$ the set of all states up to discrete time t, by $\boldsymbol{y}_{1:t} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t\}$ the set of past measurements and by $\boldsymbol{u}_{1:t} = \{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_t\}$ the set of past control vectors, and if we assume that the system is Markovian (i.e., such that $p(\boldsymbol{x}_t | \boldsymbol{x}_{0:t-1}, \boldsymbol{z}_{1:t-1}, \boldsymbol{u}_{1:t}) = p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}, \boldsymbol{u}_t)$), then the posterior probability density function $p(\boldsymbol{x}_t | \boldsymbol{y}_{1:t}, \boldsymbol{u}_{1:t})$ at time t can be