Robotics and Autonomous Systems 88 (2017) 11–23

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

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

A fault tolerant architecture for data fusion: A real application of Kalman filters for mobile robot localization

Kaci Bader *, Benjamin Lussier, Walter Schön

Sorbonne Universités, Université de Technologie de Compiègne, CNRS, UMR 7253, Heudiasyc, CS 60 319, 60203 Compiégne, France

HIGHLIGHTS

- We propose a fault tolerance architecture in multisensor data fusion.
- We use the main classical Duplication/Comparison method for fault tolerance.
- We detail an application on vehicle localization for this architecture.
- We detail the error detection, system recovery services for our approach.
- We present our experimental study validating our approach using real data and fault injection.

ARTICLE INFO

Article history:
Received 11 December 2015
Accepted 14 November 2016
Available online 22 November 2016

Keywords:
Data fusion
Multi-sensor perception
Dependability
Fault tolerance

ABSTRACT

Multisensor perception has an important role in robotics and autonomous systems, providing inputs for critical functions including obstacle detection and localization. It is starting to appear in critical applications such as drones and ADASs (Advanced Driver Assistance Systems). However, this kind of complex system is difficult to validate comprehensively. In this paper we look at multisensor perception systems in relation to an alternative dependability method, namely fault tolerance. We propose an approach for tolerating faults in multisensor data fusion that is based on the more traditional method of duplication–comparison, and that offers detection and recovery services. We detail an example implementation using Kalman filter data fusion for mobile robot localization. We demonstrate its effectiveness in this case study using real data and fault injection.

1. Introduction

Perception is a fundamental input to any robotic system. However, data perceived by such systems are often complex and subject to significant uncertainties and inaccuracies.

To overcome these problems the multisensor approach takes data from multiple, complementary sensors, and uses their redundancy to filter noise, eliminate some aberrant data, increase the precision of perception, and extract complex knowledge about the environment. But increasing the number of sensors and the underlying data fusion algorithms increases the risks of hardware and software faults. Moreover, the validation of this approach encounters two major problems:

- First, fusion algorithms are part of the declarative programming paradigm, which consists in describing a problem and the system with a view to obtaining a solution. This paradigm is often used in artificial intelligence applications such as planning, but is harder to understand and validate than imperative programming (which is the description of successive steps to execute). Nowadays, critical system standards generally preclude declarative programming approaches. For example, the EN 50128 [1] railway standard states that artificial intelligence software is not recommended in critical applications, whereas procedural programming (that corresponds to the imperative paradigm) is highly recommended. The behavior of fusion algorithms is hard to predict, making them difficult to validate by formal approaches, such as formal model and proof checking.
- Second, the open environment which complex robotic systems need to be able to handle generates a near-infinite execution context. In validation, the execution context refers to the different possible situations that the system may be faced with. In an open environment, this context is deemed near infinite because obstacles may appear at any moment in many different ways, lighting and wind conditions may

* Corresponding author.
E-mail addresses: kaci.bader@hds.utc.fr (K. Bader), benjamin.lussier@hds.utc.fr (B. Lussier), walter.schon@hds.utc.fr (W. Schön).
URL: https://www.hds.utc.fr/~baderkac (K. Bader).

http://dx.doi.org/10.1016/j.robot.2016.11.015
0921-8890/

vary, etc. Consequently, the validation of automobile systems requires many thousands of hours spent driving on roads, with no guarantee that all possible situations have been encountered, which makes testing a long, difficult and costly operation.

An alternative to this validation is the development of fault tolerance mechanisms: since it is difficult to remove all faults in the system, we instead seek to limit their impact on system function. In this paper we focus on the issues of fault tolerance in multisensor perception systems, which are fundamental inputs to any autonomous robotic system. In our approach we propose an architecture based on duplication–comparison for detecting and diagnosing faults in a data fusion mechanisms. We illustrate this architecture with an example application for mobile robot localization using Kalman filter data fusion, and we detail its fault tolerance services such as fault detection and system recovery that make it suitable for ensuring the reliability of multisensor perception systems.

This paper is organized as follows: after this introduction, Section 2 summarizes concepts and related works on fault tolerance in data fusion. Section 3 describes our proposed architecture, and details its fault tolerance services such as fault detection and system recovery. Section 4 describes an intelligent vehicle localization application that implements our duplication–comparison approach, and Section 5 presents our experimental study, validating our approach using real data and fault injection. Finally the paper ends with conclusions and prospects for future works.

2. Concepts and related work

This paper studies two different domains, each using specific terminologies and concepts. Dependability centers on the notion of fault, as a potential cause of system failures, and offers various means, including fault tolerance, to deal with it. Data fusion involves merging different sensor outputs to provide better perception of the system's environment. Both fault tolerance and data fusion use redundancy, but the former tries to detect and tolerate internal faults, while the latter focuses on the vagaries of an open, shifting environment. This section introduces the concepts of both these domains, and presents a state of the art regarding fault tolerance mechanisms in data fusion.

2.1. Dependability

A system's dependability is its ability to deliver a service that can justifiably be trusted [2,3]. This notion encompasses three different concepts: (a) *its attributes,* i.e. the expected properties of the system, (b) *its threats,* i.e. unacceptable behaviors of the system that are causes or consequences of a lack of dependability, (c) *its means,* i.e. methods that allow a system to dependably perform its function (that is by placing a justified confidence in the service it delivers). For more information on general concepts in dependability, the reader may refer to [2] and [3].

- **The attributes** of dependability are properties that a system must satisfy. Six main attributes are defined: *Availability*, *Reliability*, *Maintainability*, *Safety*, *Confidentiality*, and *Integrity*. In this work we focus particularly on two attributes of dependability: *safety* that is the absence of catastrophic consequences on the user(s), the system, and the environment and *reliability* that ensure continuity of correct service. A system is reliable if it delivers continuously and correctly its service for a specified period. Note that trying to achieve both of these attributes may be contradictory. Indeed, for an autonomous vehicle in a dangerous situation, safety may require the vehicle to stop and assess the situation, while reliability would call for the service performed not to be interrupted.
- **Threats** are undesirable behaviors of the system. There are three types: *faults*, *errors* and *failures*. These *threats* are linked by a causal relationship: the fault is the adjudged or hypothesized cause of an error, while the error is likely to result in a failure.
- **Means** are designed to counter the threats described above. They are classified into four types: *Fault prevention, Fault removal, Fault forecasting,* and *Fault tolerance,* that is, how to allow a system to properly fulfill its function in the presence of faults.

2.1.1. Fault tolerance

One of the four means of dependability, *fault tolerance* aims to ensure proper delivery of a system's services despite the presence of faults. Fault tolerance is implemented principally via *error detection* and *system recovery*.

- **Error detection** is a prerequisite for the implementation of fault tolerant solutions. It aims to detect the erroneous state of the system before errors are propagated and cause system failures. There are three main methods of error detection: *Duplication–comparison:* consists in comparing results from at least two redundant units that are independent of the faults to tolerate and provide the same service; *Temporal watchdog:* consists in checking a temporal error in a system by controlling its response time, which should not exceed a maximum value (timeout); *Likelihood checks:* seeks to detect errors by checking against aberrant values in the system state.
- **System recovery** allows an error-free state to be substituted in place of an erroneous state. This substitution can be made in three ways: *Recovery* restores the system to a correct state that was encountered before the error occurred. This correct state must previously have been saved by the system; *Pursuit* seeks a new state from which the system can function properly (possibly in a degraded mode); *Error compensation* considers that the erroneous state contains enough redundancy to allow it to be transformed into a correct state.

2.2. Data fusion and Kalman filters

Information fusion consists in combining information from multiple sources to improve decision making [4]. Data fusion systems are widely used in various fields, especially robotics, for different applications, such as navigation, obstacle detection, object tracking, etc.

Kalman filtering in data fusion involves estimating the unknown state of the system, and systematically correcting this estimation through observation. This is achieved by performing sets of sequential calculations to provide a best estimate of the system's state variables, with at each step a correction proportional to the error between the current prediction and the outputs from sensors. It is a method that has been widely applied in many robotic applications [5–8] (such as autonomous navigation, target tracking and localization). In localization applications, the data fusion approach most often used is Kalman filtering. In Section 4 we present the main concepts of the Kalman filter data fusion algorithm, which we use in our application to locate a mobile robot in its environment.

2.3. Fault tolerance in data fusion

To our knowledge, few studies exist on fault tolerance in data fusion. The approaches we have found in the literature mainly use fault tolerance by duplication–comparison. We have separated these approaches into two different classes, namely duplication