Available online at www.sciencedirect.com

SCIENCE DIRECT

Robotics and Autonomous Systems 54 (2006) 184–191

Robotics and Autonomous Systems

www.elsevier.com/locate/robot

Scalable robot fault detection and identification

Vandi Verma[a,*], Reid Simmons[b]

[a] MS 269-1, NASA Ames Research Center, Moffett Field, CA 94043, United States
[b] Robotics Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, United States

Received 18 January 2005; accepted 13 September 2005

Abstract

Experience has shown that even carefully designed and tested robots may encounter anomalous situations. It is therefore important for robots to monitor their state so that anomalous situations may be detected in a timely manner. Robot fault diagnosis typically requires tracking a very large number of possible faults in complex non-linear dynamic systems with noisy sensors. Traditional methods either ignore the uncertainty or use linear approximations of non-linear system dynamics. Such approximations are often unrealistic, and as a result faults either go undetected or become confused with non-fault conditions.

Probability theory provides a natural representation for uncertainty, but an exact Bayesian solution for the diagnosis problem is intractable. Monte Carlo approximations have demonstrated considerable success in application domains such as computer vision and robot localization and mapping. But, classical Monte Carlo methods, such as particle filters, can suffer from substantial computational complexity. This is particularly true with the presence of rare, yet important events, such as many system faults.

This paper presents an algorithm that provides an approach for computationally tractable fault diagnosis. Taking advantage of structure in the domain it dynamically concentrates computation in the regions of state space that are currently most relevant without losing track of less likely states. Experiments with a dynamic simulation of a six-wheel rocker-bogie rover show a significant improvement in performance over the classical approach.

Keywords: Robot fault diagnosis; Fault protection; Robot fault detection

1. Introduction

In this paper a fault is defined as a deviation from expected behavior. Experience has shown that even carefully designed and tested robots may encounter faults [6]. One of the reasons for this is that components degrade over time. Another is that the developers of the robot rarely have complete knowledge of the environment in which it operates and hence may not have accounted for certain situations.

Fault Detection and Identification (FDI) for robots is a complex problem. This is because the space of possible faults is very large, robot sensors, actuators, and environment models are uncertain, and there is limited computation time and power.

The algorithm presented in this paper uses Monte Carlo methods to gain accuracy. Classical Monte Carlo methods for dynamic systems, such as particle filters, are capable of tracking complex non-linear systems with noisy measurements. The problem is that estimates from a particle filter tend to have a high variance for small sample sets. Using large sample sets is computationally expensive and defeats the purpose.

This paper presents an approach for improving the accuracy of fault monitoring with a computationally tractable set of samples in a particle filter. The combination of two algorithms described in this paper enables monitoring of a wider range and larger number of faults during robot operation than has hitherto been possible. It can handle noisy sensors, non-linear, non-Gaussian models of behavior, and is computationally efficient.

2. Robot fault detection, identification, and monitoring

A *fault* is defined as a deviation from the expected behavior of the system. A *failure* is a complete interruption of the system's ability to perform the required operation [12]. *Fault detection* is defined as the process of determining that a fault

* Corresponding author.
E-mail addresses: vandi@email.arc.nasa.gov (V. Verma), reids@ri.cmu.edu (R. Simmons).

0921-8890/$ - see front matter
doi:10.1016/j.robot.2005.09.028

has occurred. *Fault identification* is the process of determining exactly which exception or fault occurred. Fault detection and identification are typically passive, i.e., they do not alter control actions. *Fault monitoring* is the process of providing a distribution over fault and operational states when there is uncertainty in the domain. The approach presented in this paper performs fault monitoring.

The faults addressed here include mechanical component failures, such as broken motors and gears; faults due to environmental interactions, such as a wheel stuck against a rock; and sensor failures, such as broken encoders.

2.1. Motivation

In a number of application domains robots are required to operate without human intervention. It is essential for these robots to monitor their behavior so that faults may be addressed before they result in catastrophic failures. An example of this is the Dante II robot [4]. In 1994, Dante II was deployed in a remote Alaskan volcano to demonstrate remote robotic exploration. While ascending out of the crater, it encountered steep slope and cross-slope conditions that changed the system dynamics. Failure to identify this resulted in the robot falling on its side. Dante II was unable to self-right and had to be rescued by helicopter.

Another example is the Mars Polar Lander. It is hypothesized that a sensor spike made it turn off its landing thrusters before had actually landed. Since the engine turned off too soon, the spacecraft fell to the surface at about 50 miles per hour, and crash-landed [17].

A recent example is from the Mars Exploration Rover, Spirit. There is a lubricant leak in one of the wheels on Spirit. This fault was detected by the large team of engineers who painstakingly analyze rover telemetry every night. A fully autonomous rover would be required to detect this fault autonomously. The detection of this fault has allowed the team to modify the control algorithm and continue operation with five wheels.

Not only are robots venturing into areas inaccessible or dangerous for humans, but they are also increasingly becoming a part of day to day life. It is also important for these robots to detect faults in a timely manner, since failure to do so may result in expensive consequences, both monetary and in terms of consumer trust that may be hard to regain. If faults go undetected, autonomous robots in real-world environments may behave in an unpredictable or dangerous manner. On the other hand, detecting and recovering from faults can considerably improve the performance of the robots [5]. To maximize successful operation the emphasis needs to be on designing to minimize faults as much as possible, and to include algorithms to detect and recover from faults when they do occur. The focus of this paper is on the fault detection aspect of the problem.

2.2. Challenges

Identifying certain faults requires context sensitive interpretation of sensor data that can be obtained only by monitoring the dynamics of the system over time, which tend to differ according to operating conditions. For example, for a rover, an increase in the power required for locomotion on flat ground may be a cause for concern, but a similar increase on a slope might be perfectly acceptable. Sensors do not directly report these dynamics because they are noisy and limited, i.e., they do not have complete information about the state of the rover and the environment that it is operating in. Control actions do not provide complete information about state transitions either, since faults and environmental interactions induce involuntary transitions. In addition, there are a large number of components that can fail in various combinations at any instant in time and the computational resources are too limited to consider all possible combinations.

3. Classical particle filter for monitoring faults

Our formulation of the fault monitoring problem requires estimating the robot and environmental state, as it changes over time, from a sequence of sensor measurements that provide noisy, partial information about the state. The Bayesian approach to dynamic state estimation addresses this problem. Computing the exact Bayesian posterior analytically is intractable for the fault monitoring problem. Hence, we use a particle filter approximation in this paper. Particle filters are a Monte Carlo approximation method for dynamic state estimation. Particle filters have been extensively used for Bayesian state estimation in non-linear systems with noisy measurements [11,9,8]. They approximate the probability distribution with a set of samples or particles.

State estimation is the process of determining the state of a system from a sequence of data. Fault monitoring has a natural interpretation as a state estimation problem. Possible fault and operational modes of the systems are represented as explicit states. The sequence of measurements is then used to determine the state of the system.

The multivariate state at time t is denoted as s_t and measurements or observations as z_t. We use the discrete time, first-order Markov formulation of the dynamic state estimation problem, hence the state at time t is a sufficient statistic for the history of measurements. That is, $p(s_t \mid s_{0\ldots t-1}) = p(s_t \mid s_{t-1})$ and the observations depend only on the current state, i.e., $p(z_t \mid s_{0\ldots t}) = p(z_t \mid s_t)$. The posterior distribution at time t, $p(s_{1\ldots t} \mid z_{1\ldots t})$, includes all the available information up to time t and provides the optimal solution to the state estimation problem. We are interested in the filtering distribution, $p(s_t \mid z_{1:t})$, which is a marginal of this distribution. The recursive filter is defined as follows:

$$p(s_t \mid z_{1\ldots t}) = \eta_t \; p(z_t \mid s_t) \int p(s_t \mid s_{t-1}) \; p(s_{t-1} \mid z_{1\ldots t-1}) \, \mathrm{d}s_{t-1}. \quad (1)$$

This process is known as Bayesian filtering, optimal filtering, or stochastic filtering and may be characterized by three distributions: (1) a transition model $p(s_t \mid s_{t-1})$, (2) an observation model $p(z_t \mid s_t)$, and (3) an initial prior distribution, $\pi(s_0)$.