To appear in Journal of Engineering Optimization, published by Taylor and Francis. Accepted and forwarded to publisher October 2001.

PARAMETER IDENTIFICATION OF AUTONOMOUS VEHICLES USING MULTI-OBJECTIVE OPTIMISATION

Tomonari Furukawa and Gamini Dissanayake

Australian Centre for Field Robotics, J04
School of Aerospace, Mechanical and Mechatronic Engineering
University of Sydney, NSW 2006 Australia
Ph: +61-2-9351-8163
FAX: +61-2-9351-7474
E-mail: tomonari@acfr.usyd.edu.au

ABSTRACT

In order to properly operate an autonomous vehicle navigation system, it is important that the vehicle and sensor models of an autonomous vehicle are defined by an accurate parameter set. This paper presents a technique for identifying parameters of an autonomous vehicle using multi-objective optimisation, which enables the identification process without introducing additional parameters. A multi-objective optimisation method has been further proposed to solve the optimisation problem defined for the identification efficiently and promisingly. Results of numerical examples first show that the proposed optimisation method can work well for various multi-objective optimisation problems. Then, the proposed identification technique has been applied to the actual parameter identification of the autonomous vehicle developed by the authors, and an appropriate parameter set has been obtained.

Keywords: Autonomous vehicle navigation system, parameter identification, multi-objective optimisation

1 INTRODUCTION

The past few years have seen an increasing interest on the development of autonomous vehicles for outdoor applications [1-3]. A suitable vehicle navigation system that provides the knowledge of vehicle position and trajectory and subsequently controls the vehicle along a desired path is an essential requirement for successful deployment of autonomous vehicles. Basic navigation system of an autonomous vehicle consists of sensors that extract information from the vehicle drive train, such as wheel and steering encoders, or an inertial measurement unit that measures the angular velocities and the accelerations of the vehicle in three orthogonal axes. These sensors that measure the vehicle state internally are usually known as dead-reckoning sensors. Information from dead-reckoning sensors is used together with a kinematic model to predict the motion of the vehicle. Errors that inevitably accumulate due to the integration present in the prediction step can be reset using absolute sensors that directly measure the vehicle location with respect to its environment in an external sense, for example, using lasers that observe beacons present in the environment [4] or the Global Positioning System (GPS) in the case of outdoor vehicles. As absolute information is usually not available at high enough rates to be useful for control purposes, it is important that the dead-reckoning sensors provide accurate information in between such updates.

One of the main sources of error in the location predicted by the dead reckoning is due to the incorrect kinematic parameters and the calibration factors used in the kinematic model. Significant work has been done in identifying the sources of dead-reckoning error and devising benchmarks for measuring such errors in indoor robots accordingly [5]. In most of the previous work, the encoders

attached to the wheels and steering are calibrated using only specific manoeuvres such as moving along straight lines and circular paths [6] and correlating the distance travelled as measured by the encoders and an external measuring device, typically a tape measure, whereas the kinematic parameters are simply based on the original design. These parameters associated with the kinematic model and sensors are thus measured or computed when the vehicle is designed or commissioned. With the notable exception in the on-line estimation of the tyre radius [1], which can change significantly as the load on the vehicle is changed, the parameters are then used as constants within the navigation algorithms. As the characteristics of electro-mechanical systems, however, change with time gradually, the ability to periodically recompute these parameters to check their validity can be of significant benefit.

The goal of the paper is to develop a technique for the computation of the kinematic parameters and calibration constants using data gathered during the normal operation of an autonomous vehicle. This would make it possible to check whether the parameters used in the navigation algorithms are accurate and make any necessary changes without resorting to specific test manoeuvres or modifications to sensor configurations. Such a parameter identification problem is formulated as a minimisation of the difference between the locations of the vehicle computed using the kinematic equations and using an absolute position-sensing device. Basically, the information from the dead reckoning sensors are used in the prediction while a Global Positioning System (GPS) receiver is used as an absolute sensor, and the criteria to be minimised are the position and orientation of the vehicle, which are used to describe its location.

Any of the conventional optimisation strategies [7-10] can be utilised in the search for the set of parameters by formulating an objective function that minimises a weighted sum of the residuals in position and orientation. However, the solution obtained is obviously dependent on the weighting

factors used, and, as the magnitudes of the residues are measured in different units, it is not possible to determine the appropriate weighting factors except through trial and error.

The objective of this paper is to describe a novel strategy for solving optimisation problems that have multiple objectives and demonstrate the effectiveness of this technique by using it to solve the parameter identification problem for autonomous vehicles. In order to solve this class of problems efficiently and promisingly, multi-objective continuous evolutionary algorithms (MCEAs), extended from continuous evolutionary algorithms (CEAs) [11,12], have been further proposed. The use of the multi-objective optimisation technique allows the parameter set to be found without introducing any additional weighting factors [13-15]. As the multi-objective method also finds the solution space rather than a single solution, parametric studies can be done while searching for a solution.

This paper is organised as follows. Section 2 provides the background material on autonomous vehicles and describes the experimental set up used for obtaining the data. The parameter estimation problem for autonomous vehicles is formulated as a multi-objective optimisation problem in section 3. Section 4 describes the multi-objective optimisation method proposed to solve this class of problems efficiently and promisingly. Numerical examples in section 5 first demonstrate the applicability of the proposed optimisation method using example problems, and then the solution to the problem of parameter identification in autonomous vehicles is provided in section 6. Conclusions are summarised in section 7.

2 AUTONOMOUS VEHICLES

2.1 Experimental Setup

Figure 1 shows a vehicle used as a test bed for research into the navigation of autonomous vehicles. This is a rear wheel driven vehicle that is steered using an Ackerman type steering linkage driving the front wheels. Four sensors are mounted on the vehicle. As dead-reckoning sensors, an encoder, fitted to the rear left wheel, gives a measure of the vehicles speed, and a linear variable differential transformer (LVDT) on the steering rack provides a measurement proportional to the steering angle. The encoder and the LVDT are read at a rate of 20 Hz. Carrier Phase Differential GPS unit with a rated accuracy of 0.02 m in position and 0.02 m/s in velocity when at least six satellites are in view is used to measure the absolute position of the vehicle at a sample rate of 4 Hz. An inertial measurement unit comprising of three orthogonal gyroscopes and three accelerometers are also mounted on the vehicle. In the work described in this paper, only one of these gyroscopes is used to measure the angular velocity of the vehicle about a vertical axis. The inertial measurement unit provides information at a sample rate of 125 Hz.

2.2 Vehicle models

The kinematic model of a vehicle moving in the horizontal plane is shown in Fig. 2. Location of the vehicle is given by state variables $[x, y, \phi]$, where x and y are the coordinates of the centre of the rear-axle and ϕ is the orientation of the vehicle body as shown. The inputs that are used to control the vehicle are the velocity at the centre of the rear axle v and the average steering angle γ. The equations of motion for this vehicle at any time instant k is given by:

$$\dot{x}(k) = v(k) \cdot \cos\phi(k) \tag{1a}$$

$$\dot{y}(k) = v(k) \cdot \sin\phi(k) \tag{1b}$$

$$\dot{\phi}(k) = \frac{v(k)}{l} \tan \gamma(k) \tag{1c}$$

where l is the vehicle wheel base.

2.3 Sensor models

2.3.1 Steering encoder

The steering encoder measures the displacement of the steering rack $\gamma_{ENC}*(k)$, which is linearly proportional to the steering angle. The steering angle $\gamma(k)$ can be expressed as

$$\gamma(k) = c_1 \cdot \gamma_{ENC}*(k) + c_2, \tag{2}$$

where c_1 and c_2 are the gain and the offset of the encoder.

2.3.2 Wheel encoder

The wheel encoder provides the angular position of the left rear wheel of the vehicle. Difference between successive position measurements can be used to determine the velocity of the left rear wheel. The velocity of the vehicle $v(k)$ is related to the velocity measured by the encoder $v_{ENC}*(k)$ through the following kinematic transformation.

$$v(k) = c_3 \cdot v_{ENC}(k)* + \dot{\phi}(k) \cdot b, \tag{3}$$

where c_3 is the gain of the encoder. Note that the substitution of equation (3) into equation (1c) introduces another velocity term $v(k)$. Assembling the velocity terms, resultant velocity $v(k)$ is described as

$$v(k) = \frac{l \cdot c_3 \cdot v_{ENC}*(k)}{l - b \cdot \tan \gamma(k)}. \tag{4}$$

2.3.3 GPS

The GPS sensor to mounted on the vehicle directly provides the absolute position $[\, x_{GPS}\,*(k)\,,\, y_{GPS}\,*(k)\,]$ at which the sensor is mounted. The vehicle location is related to the measurement obtained from the GPS sensor through the following equations.

$$x*(k) = x_{GPS}\,*(k) - r\cdot\cos\big(\phi*(k)+\theta\big), \tag{5a}$$

$$y*(k) = y_{GPS}\,*(k) + r\cdot\sin\big(\phi*(k)+\theta\big). \tag{5b}$$

where r and θ are the location of the GPS unit, in polar coordinates, with respect to the local coordinate frame on the vehicle as shown in Fig. 3.

2.3.4 Inertial measurement unit

The rate of change of orientation of the vehicle $\dot{\phi}*(k)$ is related to the reading of the gyroscope $\dot{\phi}_{INS}\,*(k)$ by

$$\dot{\phi}*(k) = \dot{\phi}_{INS}\,*(k) + \dot{\phi}_{OFF}\,*, \tag{6}$$

where the initial offset $\dot{\phi}_{OFF}\,*$ is the average value of the gyroscope measurements obtained *a priori* when the vehicle is stationary:

$$\dot{\phi}_{OFF}\,* = \frac{\sum_{i=1}^{s}\dot{\phi}_{INS}\,*(i)}{s}\,. \tag{7}$$

2.4 Simulation and measurement of the vehicle state

The sensors described in the last section can be used to both simulate and measure the state of the vehicle at any instant. In the simulation, the state of the vehicle $[\, x(k+1)\,,\, y(k+1)\,,\, \phi(k+1)\,]$ at

time instant $k+1$ is iteratively computed from the state of the vehicle $[x(k), y(k), \phi(k)]$ as shown in Fig. 4, so that data to be prepared *a priori* are control inputs $[v(k), \gamma(k)]$ with respect to all time (k =1,2,..., k_f) and initial state of the vehicle $[x(0), y(0), \phi(0)]$. With the navigation data $[\gamma_{ENC}*(k), v_{ENC}*(k)]$ from the sensors, the control inputs are derived from equations (2) and (3) at any time instant k, and the initial position of the vehicle can also be obtained using equation (5) from measurements by setting $[x(0), y(0), \phi(0)]$=$[x*(0), y*(0), \phi(0)*]$. The state of the vehicle at any time instant can be conclusively computed as far as initial orientation $\phi(0)*$ is specified.

The location of the vehicle $[x*(k), y*(k), \phi*(k)]$ can also be computed from sensory measurments at all times. Since the data obtainable from the gyroscope is the rate of change of the vehicle orientation $\dot{\phi}*(k)$, the orientation of the vehicle $\phi*(k)$ can be computed by iteratively deriving $\phi*(k+1)$ from $\phi*(k)$, given its initial state $\phi(0)*$. As shown in equation (5), the position is also governed by $\phi(0)*$ in addition to the measurements from GPS. The location of the vehicle at all times is thus determined by obtaining measurements from the GPS and the gyroscope by using the equation.

2.5 Prediction of the vehicle state

Figure 5 shows a pictorial view of the process used for estimating the location of an autonomous vehicle. Information obtained from internal sensors is used together with a kinematic model of the vehicle to obtain an estimate of the vehicle state. Due to noise present in the sensors as well as the inaccuracies of the vehicle model, error in this estimate gradually increases. Thus, information from external sensors is periodically measured, and errors accumulated during each period are incorporated into the state estimate using a Kalman filter based estimator to correct errors

and obtain a more accurate state estimate. Note that this information may not be available for extended periods of time depending on the environment in which the vehicle operates. GPS signals are often prone to blackout near buildings and other structure that obstruct or reflect radio signals. In such situations, the vehicle navigation purely relies on the estimates obtained from the internal sensors and the vehicle model. Therefore, the availability of an accurate vehicle model with accurate kinematic parameters is extremely valuable for the proper functioning of an autonomous vehicle navigation system.

3 PARAMETER IDENTIFICATION OF AUTONOMOUS VEHICLES

3.1 Problem formulation

By observing the autonomous vehicles in the last section, the parameter identification problem of concern can be characterised as follows:

- Parameters to be identified are $\mathbf{x}^T = [c_1, c_2, c_3, l, b, \phi(0)^*, r, \theta] \in R^n$.
- Errors in position and orientation of the vehicle must be minimised to identify the parameters.
- For the Kalman filter based estimator, the predictor model is only needed to be accurate over each short time period between the receit of external sensor readings.

Identifying appropriate parameters depends upon the formulation of objective function. So as to handle more than one criterion, a single-objective function is most commonly generated by summing them each with a weighting factor as this enables conventional calculus-based optimisation methods to derive a solution. The significant problem of this formulation is however

that the solution depends upon the weighting factors chosen as the derivation of a single solution does not yield further discussion.

The only way for finding solutions that do not depend upon the weighting factors is to remove them from the formulation, and we hereby propose a multi-objective formulation.

$$\mathbf{f}(\mathbf{x})^T = \left[f_{position}(\mathbf{x}), f_{orientation}(\mathbf{x}) \right] \rightarrow \min_{\mathbf{x}}, \tag{8}$$

where, to be accurate over each short time period, objective functions $\mathbf{f}(\mathbf{x}) : R^n \rightarrow R^2$ are given by

$$\begin{aligned} & f_{position}(c_1, c_2, c_3, l, b, \phi(0)^*, r, \theta) \\ & = \sum_{i=1}^{n_p} \sum_{j=1}^{k_f'} \left(x(i \cdot k_f' + j) - x^*(i \cdot k_f' + j) \right)^2 + \left(y(i \cdot k_f' + j) - y^*(i \cdot k_f' + j) \right)^2, \end{aligned} \tag{9a}$$

$$f_{orentation}(c_1, c_2, c_3, l, b, \phi(0)^*) = \sum_{i=1}^{n_p} \sum_{j=1}^{k_f'} \left\| \phi(i \cdot k_f' + j) - \phi^*(i \cdot k_f' + j) \right\|^2, \tag{9b}$$

and

$$x(i \cdot k_f') = x^*(i \cdot k_f'), \quad y(i \cdot k_f') = y^*(i \cdot k_f'), \quad i = 1, ..., n_p, \tag{10a}$$

$$\dot{\phi}(i \cdot k_f) = \dot{\phi}^*(i \cdot k_f'), \quad i = 1, ..., n_p. \tag{10b}$$

where k_f' is the number of iterations for each period which is used for further autonomous navigation, and n_p is the number of partitions in the vehicle operation. The total number of iterations is given by $k_f = k_f' n_p$. The question first and foremost is what can be the solutions of this multi-objective optimisation problem, and the next subsection will present the Pareto-optimality accordingly.

3.2 Problem solution

While the single-objective optimisation tries to look for a single solution, multi-objective

optimisation needs to derive a solution space, and this space is the natural solution to be obtained for the weight-independent parameter identification formulated as a multi-objective optimisation problem. Any point in the space satisfies Pareto-optimality [16], which was introduced in the field of economics a century ago, so the goal of the multi-objective optimisation problem is converted to finding a set of Pareto-optimal solutions as equivalently as possible to the solution space.

Consider a problem where we have m objective functions, $f_k : R^n \to R$, $k = 1,...,m$:

$$\mathbf{f}(\mathbf{x})^T = \left[f_1(\mathbf{x}),...,f_m(\mathbf{x})\right] \to \min_{\mathbf{x}} . \tag{11}$$

A decision vector $\mathbf{x}_u \in R^n$ is said to be Pareto-optimal if and only if there is no vector $\mathbf{x}_v \in R^n$ for which $\mathbf{v} = \mathbf{f}(\mathbf{x}_v) = (v_1,...,v_n)$ dominates $\mathbf{u} = \mathbf{f}(\mathbf{x}_u) = (u_1,...,u_n)$, i.e., there is no vector $\mathbf{x}_v$ such that

$$v_i \le u_i, \forall i \in \{1,...,n\} \wedge v_i < u_i, \exists i \in \{1,...,n\} . \tag{12}$$

The great advantage of the proposed formulation is that we have multiple solutions without any weighting factor, unlike the weighted scalar optimisation that results in only a single solution with a specified weighting factor. The final solution is thus selected by analysing Pareto-optimal solutions through various parametric studies, i.e., investigating the distribution of solutions both in function and parameter spaces, and comparing solutions.

The optimisation method to find well-distributed Pareto-optimal solutions in an efficient and robust way will be presented in Section 4, and the next subsection deals with how to select a final solution from the Pareto-optimal set.

3.3 Determination of a solution

Figure 6 shows the proposed process of the selection of a single solution from a parameter identification problem where objective functions f_1 and f_2 are minimised to identify three

parameters $[x_1, x_2, x_3]$. First, the user possibly picks up a characteristic Pareto-optimal solution in function space by considering the distribution of the Pareto-optimal solutions and the importance of each objective function or by using any technique. The user may use a function space additionally defined to see other decision criteria such as L-curve [17,18].

If the function measure is of no significance, the user can choose a solution by viewing it in the solution space of each two-dimensional parameter space and using his knowledge on parameter space. The number of graphs with three parameters is thus three and that with n parameters will be $\frac{1}{2}n(n-1)$. If the number of parameters is considerably large, the visual decision-making is no longer possible, and the final solution must be selected automatically. In this case, one prominent way is to select the solution residing in the centre of solution space since this solution is robust. The authors here propose a technique where the closest solution to the centre-of-gravity is chosen as the solution. Let the Pareto-optimal solutions finally obtained be $\mathbf{x}^*_i$, $i = 1,...,r$. If each solution is evaluated in a scalar manner, i.e., $\varphi(\mathbf{x}^*_i)$, the centre-of-gravity is in general given by $\mathbf{x}^* = \sum_{i=1}^{r} \mathbf{x}^*_i \, \varphi(\mathbf{x}^*_i) \Big/ \sum_{i=1}^{r} \varphi(\mathbf{x}^*_i)$. As the Pareto-optimal solutions must be evaluated equally, we can consider all the Pareto-optimal solutions possess the same scalar value, i.e., $\varphi(\mathbf{x}^*_1) = \varphi(\mathbf{x}^*_2) = ... = \varphi(\mathbf{x}^*_r)$. No matter what the value is, the centre-of-gravity results in the form:

$$\mathbf{x}^* = \frac{\sum_{i=1}^{r} \mathbf{x}^*_i}{r}, \tag{13}$$

The effectiveness of the centre-of-gravity method cannot be proved theoretically, but it is highly acceptable, as it has been commonly used in fuzzy logic to find a solution from the solution space

described by fuzzy sets. If one still wants to choose it with human knowledge, the selection may be achieved by computational decision-making strategies such as expert systems and fuzzy logic. We shall not discuss this further as it is out of scope of the paper. The appropriateness of the solution chosen is investigated by comparing it with other solutions. If the solution is not the expected one, the user selects a different Pareto-optimal solution, and the same process is repeated until the desired solution is obtained.

In order to show another advantage of the use of a multi-objective optimisation method, Figures 7 and 8 compare the process for deriving the solution of a multi-objective optimisation problem. It is clearly seen that the optimisation process in the weighted scalar optimisation is within the loop and thus has to be repeated with a different set of weighting factors whereas the multi-objective optimisation requires only one execution. This means that multi-objective optimisation is much superior to the weighted scalar optimisation in efficiency, provided that one execution in terms of the multi-objective optimisation results in a similar Pareto-optimal set after a number of weighted scalar optimisations each with different weighting factors.

4 MULTI-OBJECTIVE OPTIMISATION

4.1 Fundamentals

As a method to find a well-distributed Pareto-optimal set robustly and efficiently, MCEA proposed here is basically represented by the following four characteristics; the method:

- Searches with multiple points such that it can find multiple Pareto-optimal solutions,
- Adopts probabilistic direct search algorithms based on evolutionary computation for

robustness,

- Implements the continuous representation of the points, continuous search formulation and continuous evaluation for efficiency,
- Stores all Pareto-optimal solutions historically generated to grasp the whole solution space.

Figure 9 shows the fundamental structure of the proposed method. First, a population of individuals, each represented by a continuous vector, is initially (generation t=0) generated at random, i.e.,

$$P^t = \{\mathbf{x}_1^t, ..., \mathbf{x}_\lambda^t\} \in (R^n)^\lambda, \tag{14}$$

where λ represent the population size of parental individuals [11]. Each vector thus represents a search point, which corresponds to the phenomenological representation of individual.

4.2 Reproduction

The definition of the recombination and mutation becomes the probabilistic distribution of the phenomenological measures accordingly. In the recombination, parental individuals breed offspring individuals by combining part of the information from the parental individuals, thereby creating new points inheriting some information from the old points. The recombination operation is then defined as

$$\begin{cases} \mathbf{x}'_\alpha = (1-\mu)\mathbf{x}_\alpha + \mu\mathbf{x}_\beta \\ \mathbf{x}'_\beta = \mu\mathbf{x}_\alpha + (1-\mu)\mathbf{x}_\beta \end{cases}, \tag{15}$$

where parameter μ may be defined by the normal distribution with mean 0 and standard deviation σ:

$$\mu = N(0, \sigma^2) \tag{16}$$

or simply a uniform distribution:

$$\mu = \text{rand}(\mu_{\min}, \mu_{\max}) . \tag{17}$$

The mutation can also be achieved simply by implementing

$$\mathbf{x}'' = \text{rand}(\mathbf{x}_{\min}, \mathbf{x}_{\max}) . \tag{18}$$

with a small possibility [19]. Note that the mutation may not be not necessary for parameter μ with normal distribution since it can allow individuals to alter largely with a small probability, when the coefficient μ is large.

4.3 Evaluation, Ranking and selection

As the Pareto-optimal set satisfying equation (12) is to be found as solutions, the ranking process of individuals is composed of an elimination rule. In the rule, the calculation of objective function at all the points $f(\mathbf{x}_i)$, $i=1,...,\lambda$, is first conducted, and the Pareto-optimal set is ranked No. 1. The points with rank No. 1 are then eliminated, and Pareto-optimal set in the population is ranked No. 2. All the subsequent ranks are generated stepwise in the same fashion until all the points are ranked [20]. The points in rank No. k, $G(k)$, are defined as

$$G(k) = \left\{ \mathbf{x}_i \mid \text{rank}(\mathbf{x}_i) = k, \forall i \in \{1,...,\lambda\} \right\} \tag{19}$$

for further convenience, and the ranking process of the individuals is summarised in Fig. 10. The ranked points are illustrated in Fig. 11.

The evaluation of fitness of each individual starts with finding the best and worst value of each objective function among the population:

$$f_{\text{best}j} = \min \left\{ f_j(\mathbf{x}_i) \mid \forall i \in \{1,...,\lambda\} \right\}, \tag{20}$$

$$f_{\text{worst}j} = \max\left\{ f_j(\mathbf{x}_i) \mid \forall i \in \{1,...,\lambda\} \right\}. \tag{21}$$

If we temporarily define the fitness as

$$\Phi'_j(\mathbf{x}_i) = \frac{f_{\text{worst}j} - f_j(\mathbf{x}_i)}{f_{\text{worst}j} - f_{\text{best}j}}, \tag{22}$$

we can get the normalised conditions:

$$0 \le \Phi'_j(\mathbf{x}_i) \le 1, \tag{23}$$

and this allows us to treat the fitness of each function with the same scale. The fitness of points with the same rank has to be the same, and the true fitness of each objective function is thus defined as:

$$\Phi_j(\mathbf{x}_i) \equiv \Phi_j^{G(k)}(\mathbf{x}_i) = \max\left\{ \Phi'_j(\mathbf{x}_i) \mid \mathbf{x}_i \in G(k) \right\}. \tag{24}$$

The fitness of each individual can be conclusively calculated as:

$$\Phi(\mathbf{x}_i) = \sum_{j=1}^{m} \Phi_j(\mathbf{x}_i), \tag{25}$$

which has the range

$$0 \le \Phi(\mathbf{x}_i) \le m. \tag{26}$$

The selection operator favourably selects individuals of higher fitness to produce more often than those of lower fitness. As $\Phi(\mathbf{x}_i) \ge 0$ is satisfied by this equation, the proportional selection [20], which is reported to be faster in convergence than the other popular selection of the ranking selection [21], can be directly used in the proposed algorithm. In this selection, the reproduction probabilities of individuals are given by their relative fitness:

$$P_s(\mathbf{x}_i) = \frac{\Phi(\mathbf{x}_i)}{\sum_{j=1}^{\lambda} \Phi(\mathbf{x}_j)}. \tag{27}$$

These evolutionary operations form one generation of the evolutionary process, which corresponds

to one iteration in the algorithm, and the iteration is repeated until a given terminal criterion is satisfied.

4.4 Historical storage of Pareto-optimal sets

So as to grasp the configuration of the whole solution space, the resultant Pareto-optimal solutions are stored outside the loop of the evolutionary operations. The whole Pareto-optimal solutions obtained in the first generation are saved in this storage. From the second generation, the newly created Pareto-optimal solutions in the loop are compared to the stored Pareto-optimal solutions, and the new set of Pareto-optimal solutions is saved in the storage. This strategy allows the Pareto-optimal solutions created in the past to be kept as solutions and yield a good chance to increase the number of solutions, thus making the solution space easier to see. The storage of the solution independent of the current population also may contribute to the good distribution of the resultant solutions.

4.5 Comparison with other methods

The proposed method has been characterised by the multi-objective formulation, continuous evolutionary search formulation and the historical storage for the robust and efficient search of well-distributed Pareto-optimal solutions. Some conclusions in the superiority of the proposed method can be easily deduced from the past research, but numerical investigations are necessary for the others.

In the search algorithms, two major EAs, genetic algorithms (GAs) [22] and evolution strategies (ESs) [23], originally uses binary search with proportional/ranking selection and continuous search

with ranking selection, respectively. In the previous reports [24-26], those with the binary points and the ranking selection search more robustly than those with continuous points and the proportional selection at the expense of fast convergence, and vice versa. CEAs proposed by the authors [11], incorporating continuous representation of points and proportional selection, therefore demonstrated its convergence approximately ten times faster than that of GAs and ESs [12]. MCEAs, taking over them from CEAs should also be faster than the multi-objective versions of GAs and EAs without loss of generality, and this shall not be further mentioned in the following numerical examples.

The evaluation of fitness for multi-objective optimisation is a completely new approach proposed in the paper. The ability of the approach to find appropriate Pareto-optimal solutions thus needs to be demonstrated and will be presented in the next section. The effectiveness of historical storage in the increase of the number of solutions is very likely but not certain so will also be investigated to make sure that the proposed method is better than other multi-objective optimisation methods in performance.

5 APPROPRIATENESS OF PROPOSED TECHNIQUE

5.1 Parameter identification with a simple numerical problem

In order to confirm its appropriateness for finding Pareto-optimal solutions and the increase of solutions over generations visually, MCEA was first used to identify parameters by minimising simple objective functions where the exact set of solutions is known. Let the function be given by

$$f_1(\mathbf{x}) = \|\mathbf{x}\|^2 = \sum_{i=1}^{n} x_i^2 , \tag{28}$$

$$f_2(\mathbf{x}) = \|\mathbf{x} - \mathbf{z}\|^2 = \sum_{i=1}^{n} (x_i - z_i)^2 , \tag{29}$$

where n =2, $\mathbf{z}^T$ =[0.2,0.4] $\in R^2$, and the set of parameter $\mathbf{x} \in R^2$ is subject to bound $\mathbf{x}_{\min} \le \mathbf{x} \le \mathbf{x}_{\max}$ with $\mathbf{x}_{\min}^T$ =[-5,-5] and $\mathbf{x}_{\max}^T$ =[5,5]. The exact Pareto-optimal solution for this problem can be determined analytically and is given by

$$X = \{\mathbf{x} \mid \mathbf{x} = r\mathbf{z}, r \in [0,1]\}, \tag{30}$$

and we can thus investigate the performance of MCEA with the exact solution. Values of major parameters for MCEA used to solve the problem are listed in Table 1.

Figure 12 shows the computed Pareto-optimal set in $f_1 - f_2$ space at 50th, 500th and 2500th generations respectively together with the exact solution. The figure first implies that the good approximate solutions have been already obtained after 50 generations. It is then easily seen that computed solutions at larger generations are closer to the exact line, and this indicates that the proposed method converges appropriately to the exact solution. In addition, the number of computed solutions increases with respect to the number of generations as shown in Fig. 13, and the increase of solutions contributes to the understanding of the shape of the solution space more precisely. Figure 14 shows the resultant Pareto-optimal solutions at 50, 500 and 2500th generations in $x_1 - x_2$. One can easily see that the solutions are settling down to the exact solution with the increase of the number of generations also in parameter space.

5.2 Parameter identification with a complex numerical problem

With the understanding of the appropriate and efficient performance of MCEA for identification

with simple objective functions, the identification with more complex objective functions consisting of a number of parameters, which is a more realistic parameter identification problem, has been investigated. In this example, the number of parameters to be identified is five (n=5) where the parameter space has a bound with $\mathbf{x}_{\min}^T$=[-5,-5,-5,-5,-5] and $\mathbf{x}_{\max}^T$=[5,5,5,5,5], and the first objective function has an additional term to equation (28):

$$f_1(\mathbf{x}) = \|\mathbf{x}\|^2 + 50 - \sum_{i=1}^{5} 10\cos(\omega x_i). \tag{31}$$

The cosine term clearly makes the function multimodal with a number of local minima and the function was used as a good example for a multimodal continuous function [27]. The second objective function is equation (29) with $\mathbf{z}^T$=[0.3,0.4,0.5,0.6,0.7], and MCEA parameters listed in Table 1 was again used to solve the problem.

Figure 15 shows the resultant Pareto-optimal solutions in function space at 50th, 500th and 2500th generations. It is again seen that the distribution of the Pareto-optimal solutions is becoming smoother and better as the number of generations increases, indicating that MCEA is searching appropriately. The coarse distribution of solutions with small f_1 is yielded by its complexity. The number of solutions also increased over generations; 18 at 50th generation, 37 at 500th generation and 53 at 2500th generation. The Pareto-optimal solutions at 50th, 500th and 2500th generations in parameter spaces $x_1 - x_4$ and $x_2 - x_4$ are shown in Figs. 16(a) and (b). In parameter space $x_1 - x_4$, three groups of solution are seen at 50th generation. After the number of groups is reduced to two at 500th generation, a new solution is found near [1,1] at 2500th generation. Meanwhile, three groups of solution are seen at 50th generation in parameter space $x_2 - x_4$ where one group consists of only one solution, and they result in two of them at 2500th generation. The

existence of some groups is due to the multimodality of objective function f_1, and finding them demonstrates that MCEA is robust.

6 APPLICATION TO PARAMETER IDENTIFICATION OF AUTONOMOUS VEHICLE

With the understanding that MCEA can be used promisingly and efficiently for multi-objective optimisation problem with numerical examples in the last section, the proposed technique was applied for identifying a parameter set of the autonomous vehicle developed by the authors. The vehicle tracked the path in a flat parking area for 100 seconds as shown in Fig. 17(a) as GPS sensor data, and the gyroscope sensor, steering encoder and velocity encoder readings are shown in Fig. 17(b)-(d). Note that x and y coordinate data are collected independently with respect to time in GPS sensor data. The information from these sensors was sub-sampled at 4 Hz to obtain a synchronous sequence of data for the use in the optimisation algorithm. Table 2 lists the parameters used for simulation and optimisation to execute identification, and the search space of parameters to be identified is listed in Table 3. The search space was chosen to include the original calibration data of each parameter at the centre of the space, and the range was determined based on its reliability.

Figure 18 shows the Pareto-optimal solutions in function space after 100 generations. It is first easily found that the orientation is much smaller than the position in objective function value but that the solutions are well distributed showing a smooth convex-shaped curve in such a different scale. Next, Pareto-optimal solutions in parameter space are depicted in Fig. 19. Although the

parameter scales are also different from each other, the solutions in each graph show a characteristic distribution, from upper-left to lower-right for Fig. 19(a)-(c) and from lower-left to upper-right for Fig. 19(d). A final solution can be determined after checking solution spaces.

Due to the constant distribution of solutions along a smooth convex curve in function space, a characteristic point cannot be chosen easily with this result, so that the final solution may be found in the parameter space in sanction with the proposed procedure. Because of the high dimensionality of the parameter space, we chose the final solution by the centre-of-gravity method, and the solution is listed in Table 4. The solutions each having the minimum position and orientation errors are also shown in the table for comparison. Since rather monotonic distributions of solutions in parameter space are obtained for this example, the solution chosen by the centre-of-gravity method is well within the solution space.

The simulation result using the parameter set chosen, used to calculate the objective function values, is shown in Fig. 20 with the GPS data denoted as 'experiment'. The simulated path shows some accumulated errors, but it is well along the GPS data, indicating clearly indicates that an appropriate parameter set is identified. The errors may be caused by the slip of the vehicle and other inaccuracies of the model rather than its parameters themselves. Since there is no way to investigate the errors with the current vehicle set-up, we shall not further discuss in this paper.

To investigate the appropriateness of this solution to the other Pareto-optimal solutions, the above three sets of solutions were used for simulation without correcting the path at every partition. In order to see how robust the solution is, the simulation was conducted not only for the first 100 seconds during which GPS data were used to find the solution but also for the next 100 seconds. The simulation results with the three solutions are depicted in Figs. 21-23 respectively. The solution with the minimum position error and the solution chosen correlate well with GPS data in

comparison to the solution with the minimum orientation error. The orientation accuracy must also be investigated to find the most appropriate solution, and in order to see the results in more detail, the error values are also listed in Table 5. It is first seen that the worst solutions in position error and orientation error in both the first and second 100 seconds are the solutions with the minimum orientation error and with the minimum position error, respectively. Particularly, the orientation error by the minimum position error solution and the position error by the minimum orientation error solution, both in the second 100 seconds, are significantly large compared to the others. This is certainly caused by the fact that the other objective function is ignored; to get the minimum position error solution, for example, the objective function describing the orientation errors is taken into account. Meanwhile, the solution chosen is not worst in any criterion, and it is even better than the solution with the minimum position error in the position error of the second 100 seconds. This characteristic remained even with different numerical examples. As the accurate orientation of the vehicle at each iteration can contribute to its accurate positioning, this may have increased the accuracy of the solution chosen in position.

7 CONCLUSIONS

A parameter identification technique for autonomous vehicles and, further, a multi-objective optimisation method of MCEA, which can search solutions efficiently and promisingly for this class of problems, have been proposed. The use of multi-objective method allows the whole solution set of the problem rather than a single solution to be derived by one optimisation. The user can therefore select a single solution later by investigating the whole solution set in function and

parameter spaces.

The Pareto-optimality of solutions derived by MCEA was confirmed with the minimisation of explicit objective functions. In these examples, the searching capability of MCEA and the superiority of the proposed multi-objective technique to the conventional weighted scalar optimisation have been demonstrated. The proposed technique was then applied to parameter identification of an autonomous vehicle, and a solution was chosen from the Pareto-optimal solutions derived by MCEA. The solution was compared to the other Pareto-optimal solutions, and its appropriateness in accuracy has been demonstrated. Conclusively, the overall effectiveness of the proposed technique for parameter identification of an autonomous vehicle has been confirmed, and an appropriate parameter set has been identified for the autonomous vehicle developed by the authors.

For further studies, the parameter set will be applied to estimate the location of the autonomous vehicle. In the parameter identification technique, developing various criteria that investigate the appropriateness of a solution is important, as the solution of inverse problems is naturally unknown. This is another big step, and the author is currently working on it as the advantage of the multi-objective approach has been confirmed in this paper [28,29].

REFERENCES

[1] Durrant-Whyte, H.F. (1996). An Autonomous Guided Vehicle for Cargo Handling Applications. *International Journal of Robotics Research*, **15**(5), 407-440.

[2] Madhavan, R. Dissanayake, G. and Durrant-Whyte, H.F., Roberts, J.M., Corke P.I. and

Cunningham, J. (1999). Issues in Autonomous Navigation of Underground Vehicles. *Mineral Resources Engineering*, **8**(3), 313-323.

[3] Pilarski, T., Happold, M., Pangels, H., Ollis, M., Fitzpatrick, K. and Stentz, A. (1999). The Demeter System for Automated Harvesting. *Proceedings of the 8th International Topical Meeting on Robotics and Remote Systems*, April.

[4] Scheding, S., Dissanayake, G., Nebot, E. and Durrant-Whyte, H. (1999). Autonomous Navigation of an Underground Mining Vehicle. *IEEE Transactions on Robotics and Automation*, **15**(1), 85-95.

[5] Borenstein, J. and Feng, L. (1996). Measurement and Correction of Systematic Odometry Errors in Mobile Robots. *IEEE Transactions on Robotics and Automation*, **12(**6), 869-880.

[6] Singh, S. and Shin, D.H. (1990). Vehicle and Path Models for Autonomous Navigation. *Vision and Navigation: The CMU Navlab*, Thorpe, C. E. (ed.), Kluwer Press.

[7] Bard, Y. (1974). Nonlinear Parameter Estimation. Academic Press, New York.

[8] Dixon, L.C.W. (1972). Nonlinear Optimisation. The English Universities Press Limited, London.

[9] Press, W.H., Flannery, B.P., Teukolsky, S.A. and Vetterling, W.T. (1988). Numerical Recipes in C. Cambridge University Press, Cambridge.

[10] Nemhauser, G.L., Rinnooy Kan, A.H.G. and Todd, M.J. (1989). Handbooks in Operations Management Science Vol. 1 Optimization. Elsevier Science Publishers B.V., Amsterdam.

[11] Furukawa, T. and Dissanayake, G. (1993). Genetic Algorithms Using Real-valued Strings," *Proceedings of the 71st JSME Annual Meeting*, **930**(71), 509-510.

[12] Furukawa, T. and Yagawa, G. (1997). Inelastic Constitutive Parameter Identification Using an Evolutionary Algorithm with Constitutive Individuals. *International Journal for Numerical Methods in Engineering*, **40**, 1071-1090.

[13] Fonseca, C.M. and Fleming, P.J. (1993). Genetic Algorithms for Multi-objective Optimisation: Formulation, Discussion and Generalisation. Forrest, S. (ed.), *Proceedings of the Fifth International Conference on Genetic Algorithms*, Morgan Kaufmann, San Mateo, CA, 416-423.

[14] Fonseca, C.M. and Fleming, P.J. (1995). An Overview of Evolutionary Algorithms in Multiobjective Optimization. *International Journal of Evolutionary Computation*, **3**(1), 1-16.

[15] Furukawa, T. (2001). Parameter Identification with Weightless Regularization. *International Journal for Numerical Methods in Engineering*, **52**, 219-238.

[16] Coello, C.A. (1999). A Comprehensive Survey of Evolutionary-based Multi-objective Optimization Techniques. *International Journal of Knowledge and Information Systems*, **1**(3), 269-308.

[17] Kitagawa, T. (1988). A Numerical Method to Estimate the Optimal Regularization Parameter. *Journal of Information Processing*, **11**, 263-270.

[18] Kitagawa, T. (1992). A Comparison between Two Classes of the Method for the Optimal Regularization. Kubo, S. (ed.) *Inverse Problems*, Atlanta Technology Publications, Atlanta, 25-35.

[19] Kowaltczyk, T., Furukawa, T., Yoshimura, S. and Yagawa, G. (1998). An Extensible Evolutionary Algorithm Approach for Inverse Problems. Tanaka, M and Dulikravich, G.S. (eds.) *Inverse Problems in Engineering Mechanics*, Elsevier Science Publishers B.V., Amsterdam, 541-550.

[20] Goldberg, D. (1989). Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading, MA.

[21] Baker, J.E. (1985). Adaptive Selection Methods for Genetic Algorithms. Grefenstette, J.J. (ed.) *Proceedings of the First International Conference on Genetic Algorithms and Their*

Applications, 101-111.

[22] Holland, J.H. (1975). Adaptation in Natural and Artificial Systems. The University of Michigan Press, Michigan.

[23] Schwefel, H.-P. (1981). Numerical Optimization of Computer Models. Wiley, New York.

[24] Hoffmeister, F. and Back, T. (1992). Genetic Algorithms and Evolution Strategies: Similarities and Differences. Technical Report, University of Dortmund, Germany, Sys-1/92.

[25] Baek, T. and Schwefel, H.-P. (1993). An Overview of Evolutionary Algorithms for Parameter Optimization. *International Journal of Evolutionary Computation*, **1**(1), 1993, 1-23.

[26] Furukawa, T. and Yagawa, G. (1995). Genetic Algorithms for Real Search Space and Their Use for Nonlinear Inverse Problems. *Transactions of the Japan Society of Mechanical Engineers*, **61**(586), 1409-1415.

[27] De Jong, K. (1975). An Analysis of the Behaviour of a Class of Genetic Adaptive Systems. Ph.D Thesis, University of Michigan.

[28] Jeong, M.J., Yoshimura, S., Furukawa, T., Yagawa, G. and Kim, Y.J. (2000). A Data Interpretation for Multi-objective Multi-solution Problems Using Clustering Algorithm," *Proceedings of Computational Engineering Conference*, **5**, 231-234.

[29] Furukawa, T. and Yagawa, G. (1998). Implicit Constitutive Modelling for Viscoplasticity Using Neural Networks. *International Journal for Numerical Methods in Engineering*, **43**, 195-219.

LIST OF TABLES

LIST OF FIGURES

Table 1 Optimisation parameters for MCEA

Parameter	Value
No. of generations	2500
Population	10
Mutation rate	0.02

Table 2 Parameters for autonomous vehicle parameter identification

Parameter	Value
No. of generations	100
Population	50
Mutation rate	0.10
Time step	0.05
No. of partitions	20

Table 3 Search space of parameters to be identified

Parameter	l	b	c_3	c_1	c_2	ϕ_0	r	θ
Minimum	3.10	0.85	4.90×10^{-4}	4.50×10^{-4}	-0.925	1.96	3.65	0.160
Maximum	3.20	1.05	5.00×10^{-4}	4.60×10^{-4}	-0.900	1.98	3.67	0.190

Table 4 Solutions identified

Parameter	l	b	c_3	c_1	c_2	ϕ_0	r	θ
Chosen	3.152	0.9441	4.955×10^{-4}	4.536×10^{-4}	-0.9149	1.967	3.675	0.1801
Minimum orientation errors	3.160	0.8831	4.946×10^{-4}	4.549×10^{-4}	-0.9154	1.963	3.650	0.1770
Minimum position errors	3.145	0.9641	4.963×10^{-4}	4.513×10^{-4}	0.9203	1.973	3.695	0.1807

Table 5 Position and orientation errors

Solution	Position error		Orientation error	
	1st 100 sec	2nd 100 sec	1st 100 sec	2nd 100 sec
Chosen	710.66	23,361	1.5656	316.515
Minimum position error	706.165	37,992	2.5941	2,673.7
Minimum orientation error	719.83	253,920	1.26297	8.5044

Fig. 1 Vehicle and sensors used

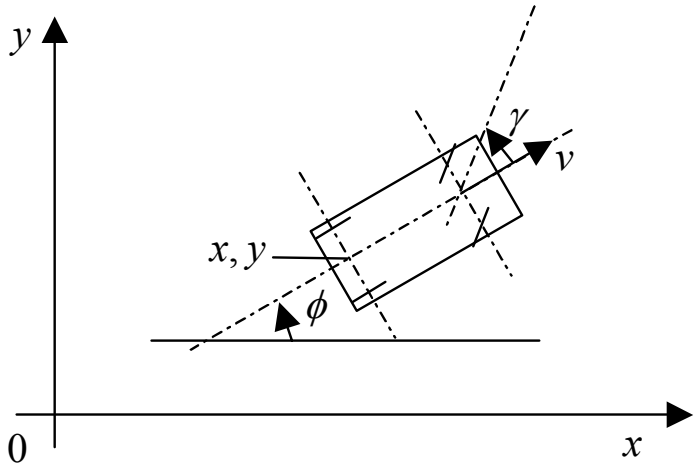


Fig. 2 State and control of the vehicle

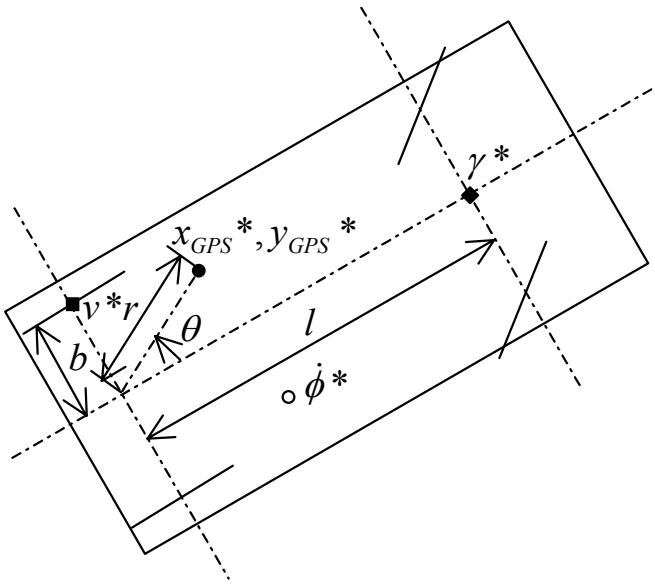


Fig. 3 Location and notation of sensors mounted on the vehicle

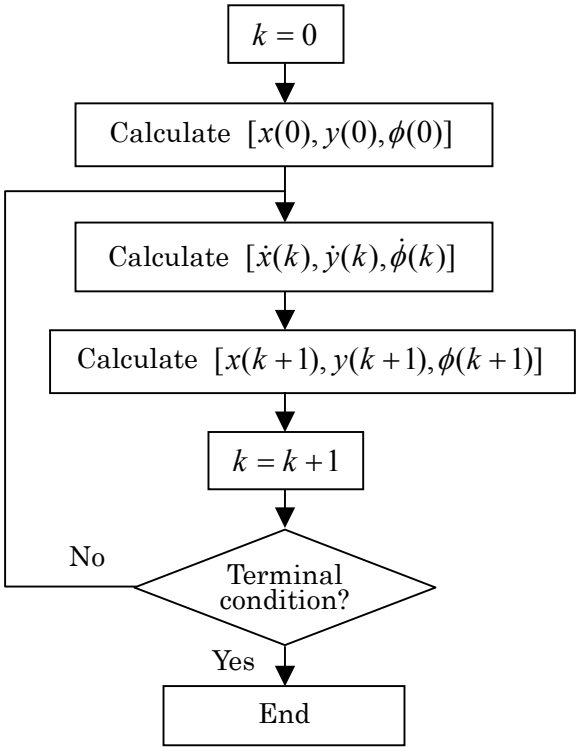


Fig. 4 Flowchart of simulation

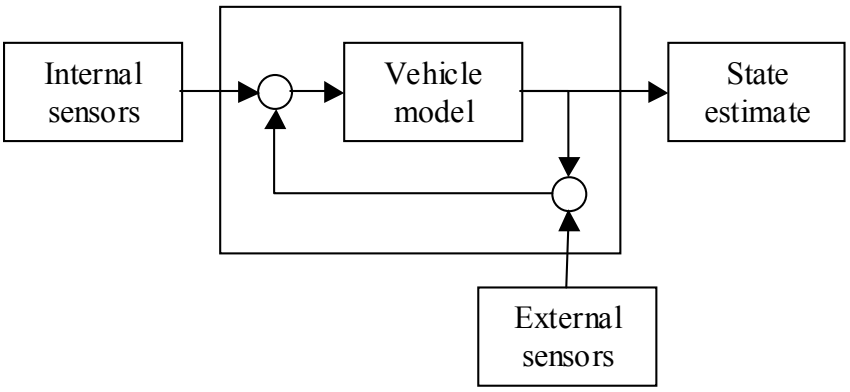


Fig. 5 Navigation system of an autonomous vehicle

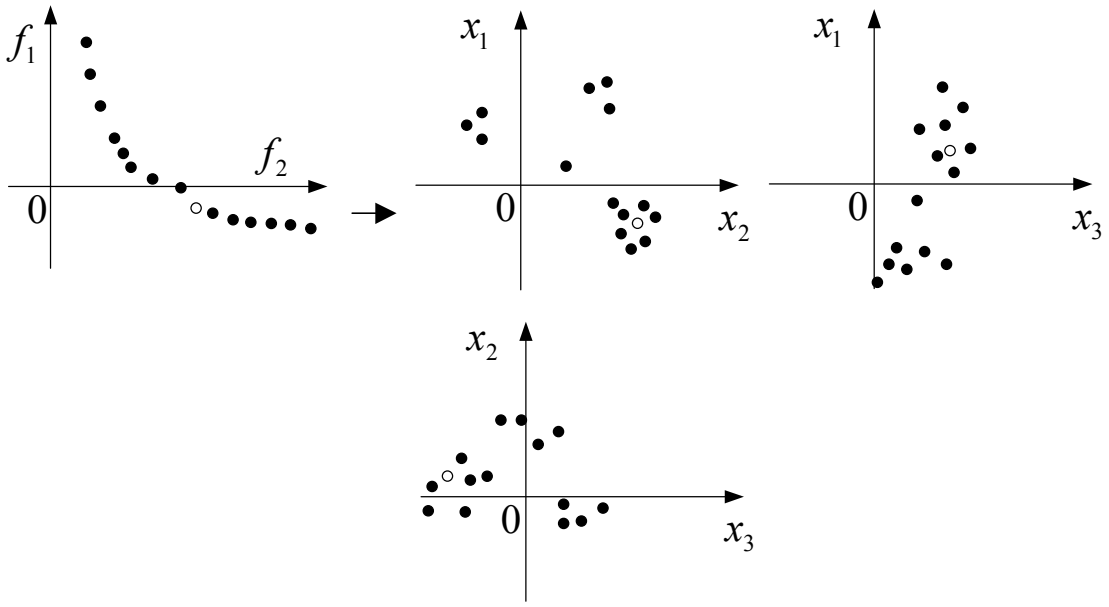


Fig. 6 Process of deriving a single solution

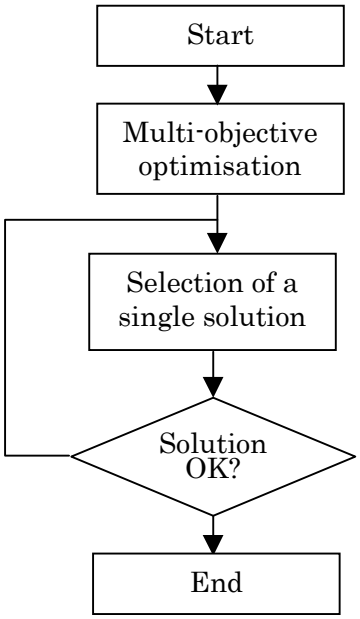


Fig. 7 Deriving a solution in multi-objective optimisation

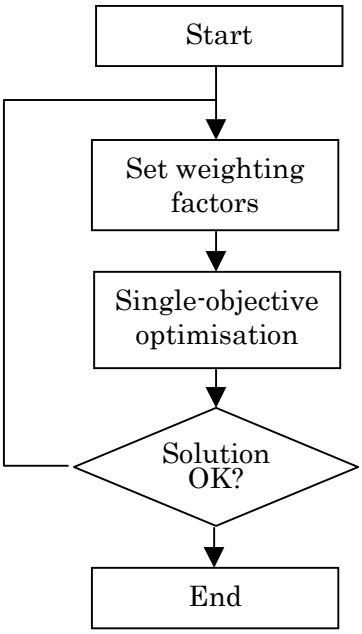


Fig. 8 Deriving a solution in weighted scalar optimisation

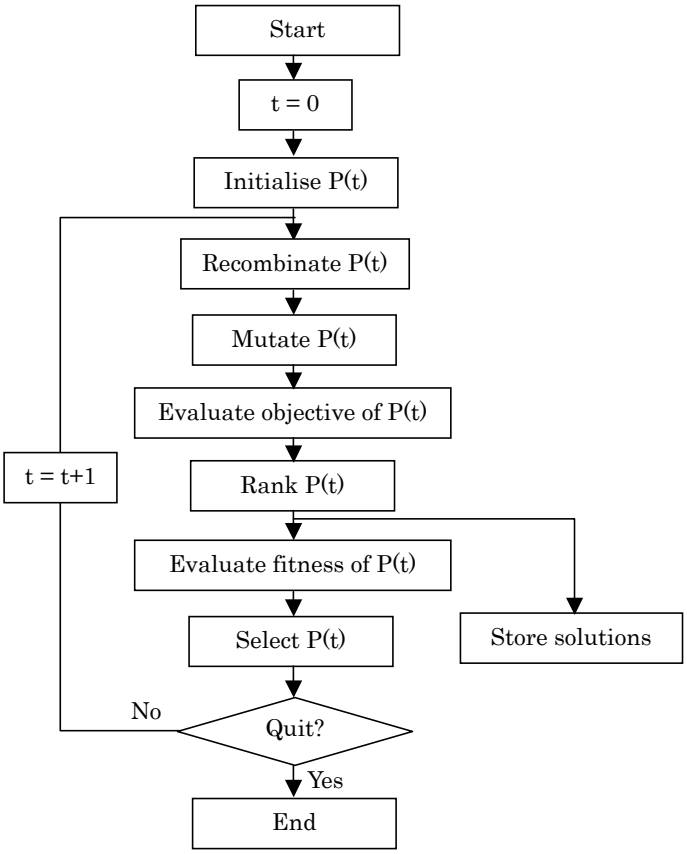


Fig. 9 Fundamental MCEAs

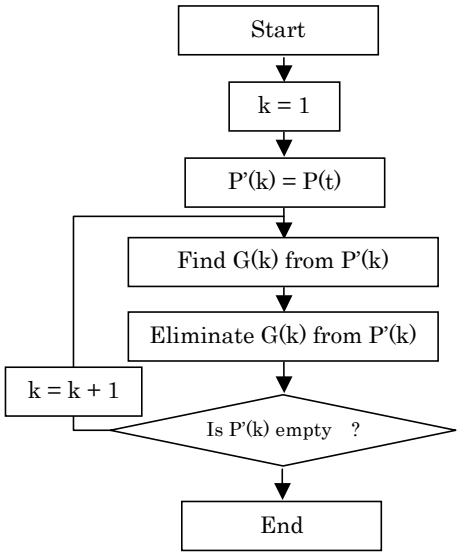


Fig. 10 Ranking process of individuals

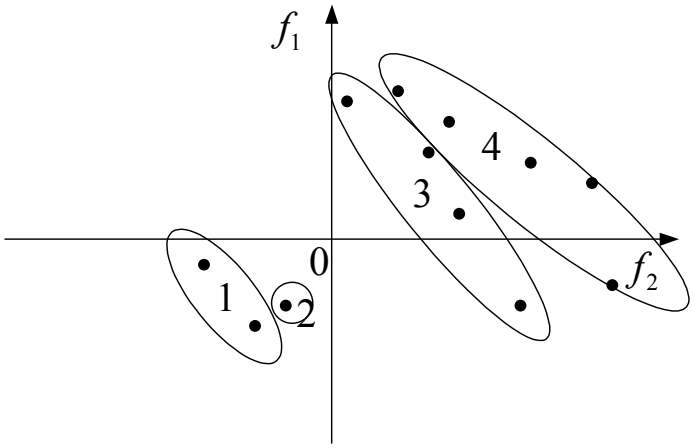
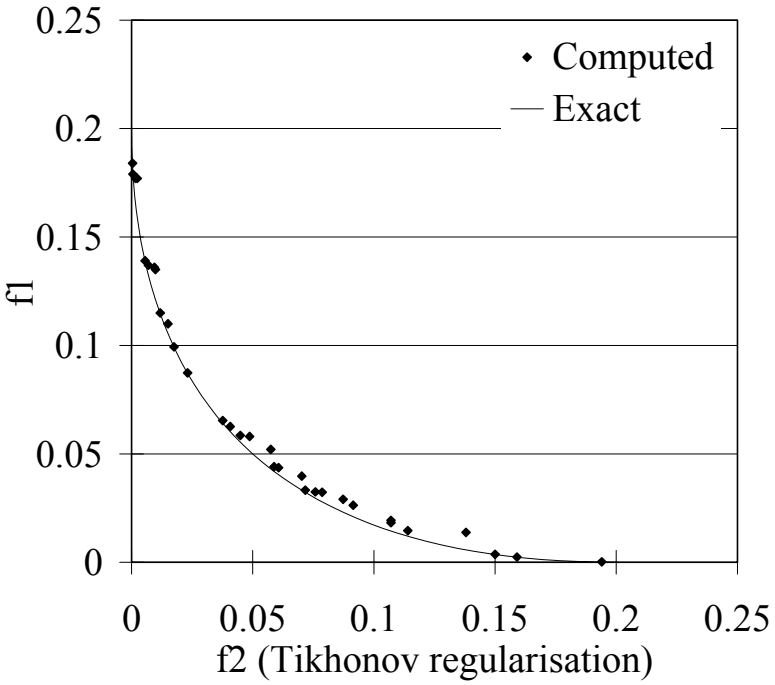
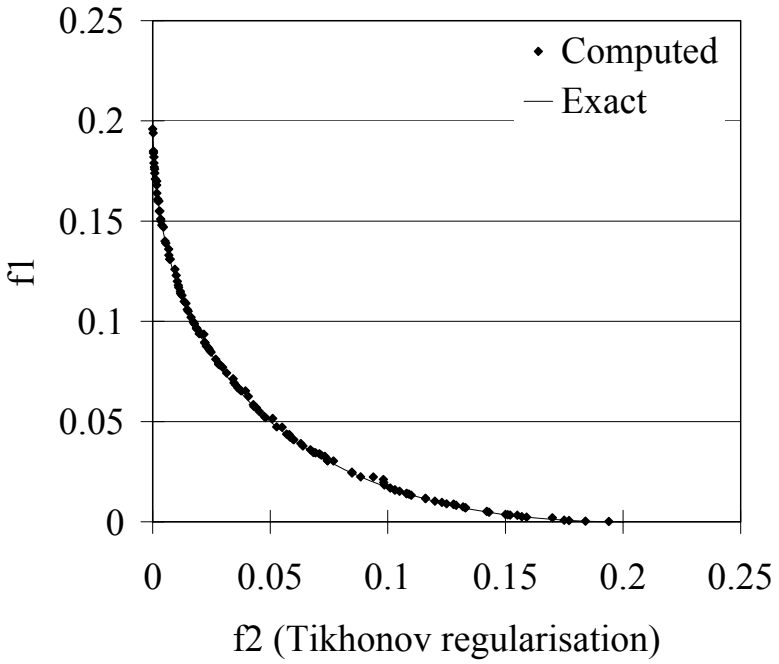


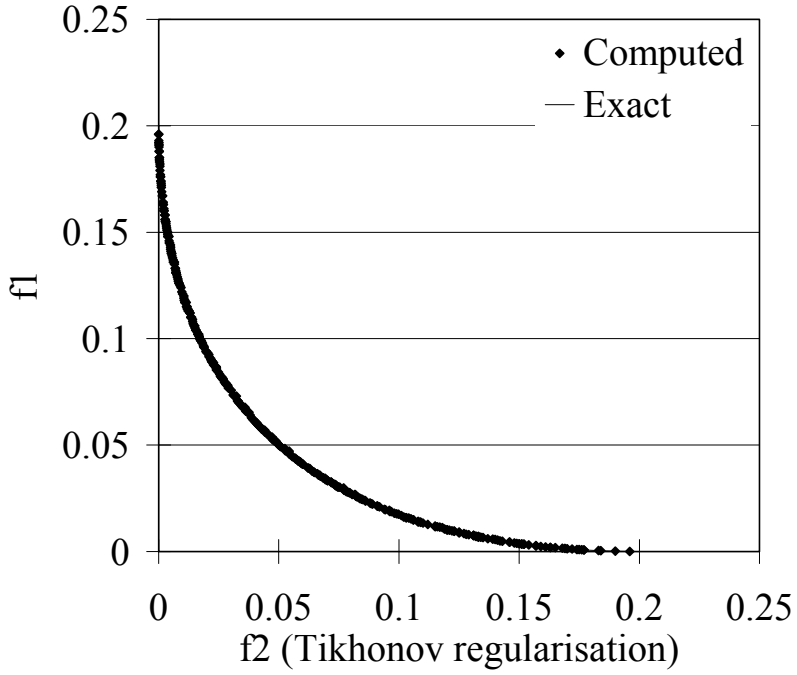
Fig. 11 Ranks of individuals.



(a) 50 generations



(b) 500 generations



(c) 2500 generations

Fig. 12 Pareto-optimal solutions of Example I in function space

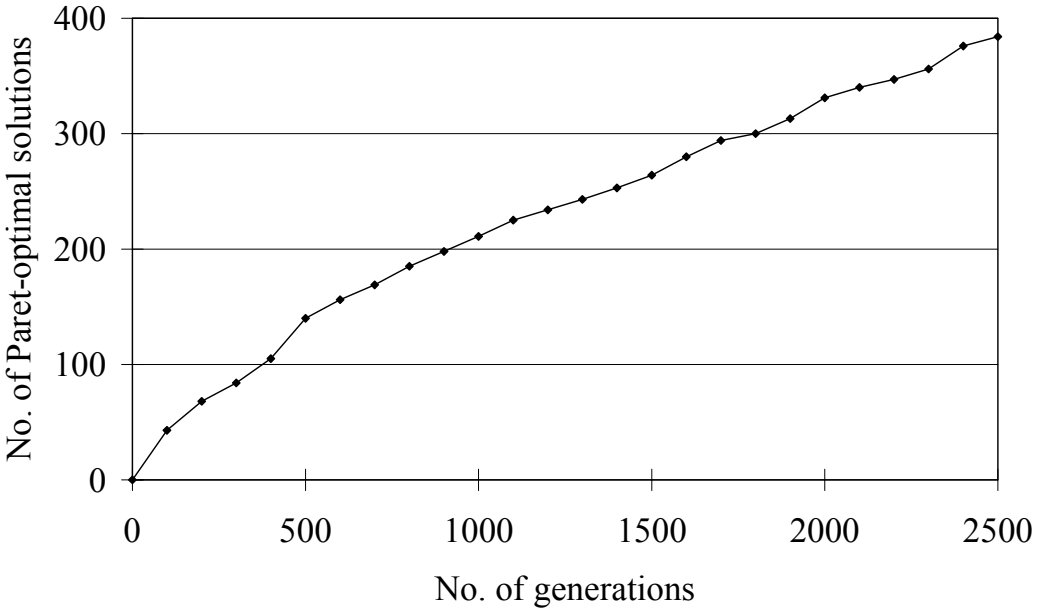
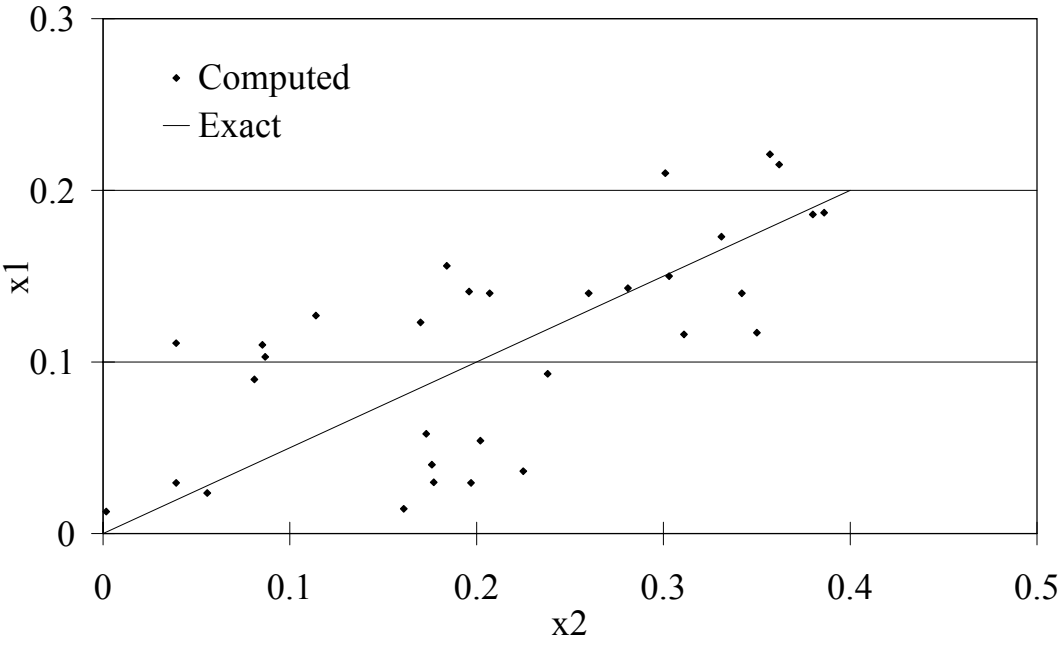
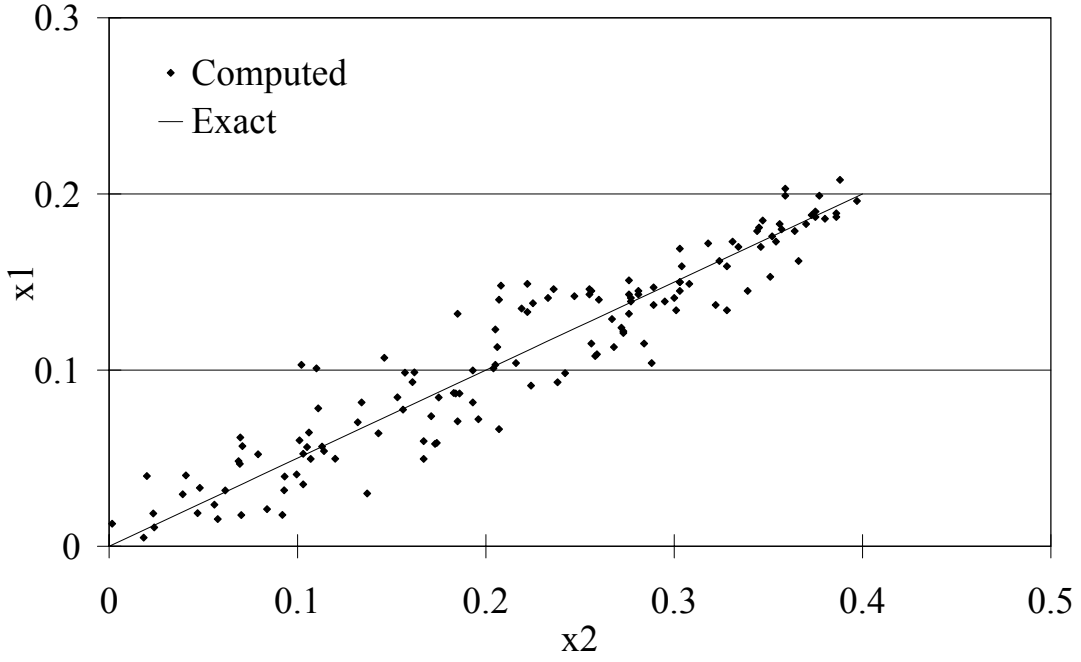


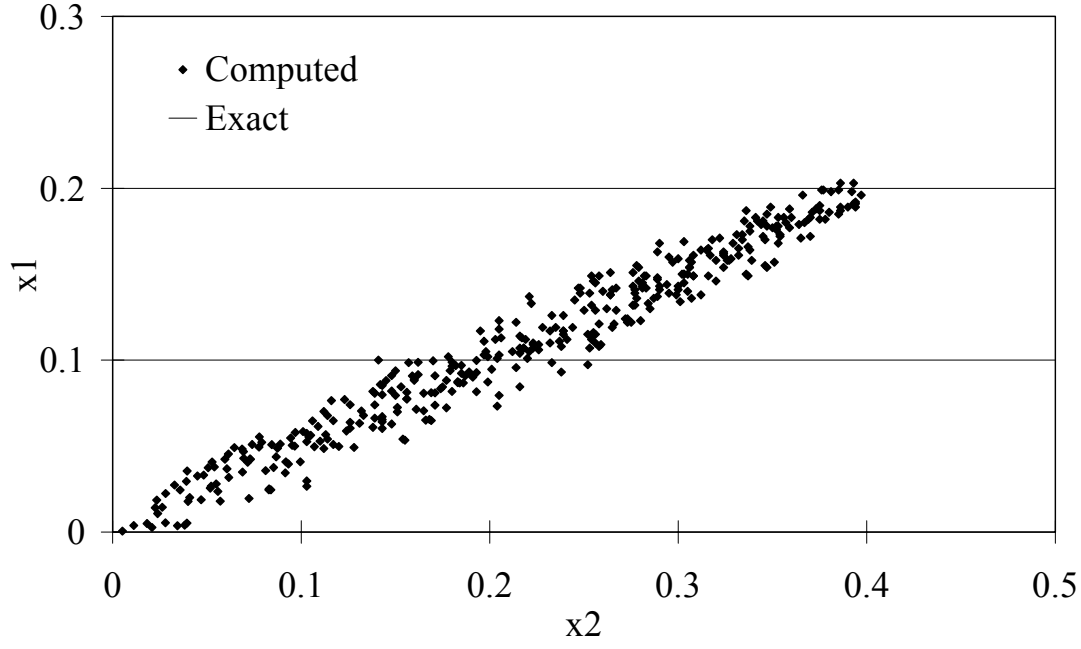
Fig. 13 Pareto-optimal solutions with respect to generations in Example I



(a) 50 generations



(b) 500 generations



(c) 2500 generations

Fig. 14 Pareto-optimal solutions of Example I in parameter space

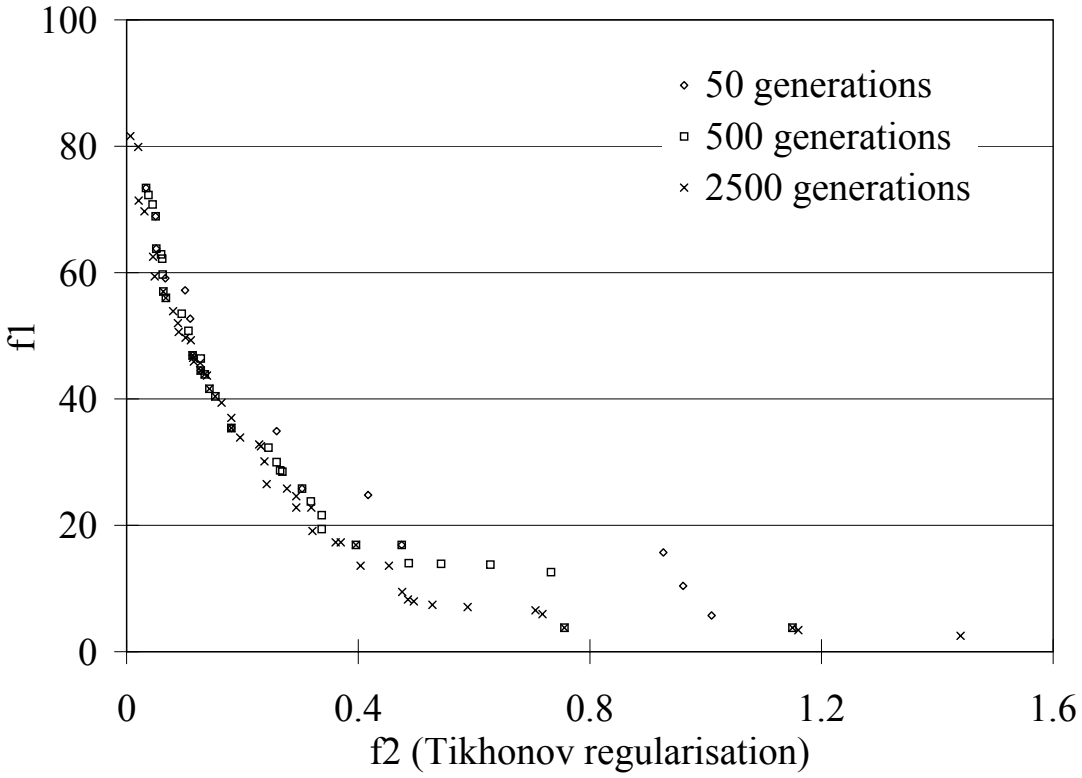
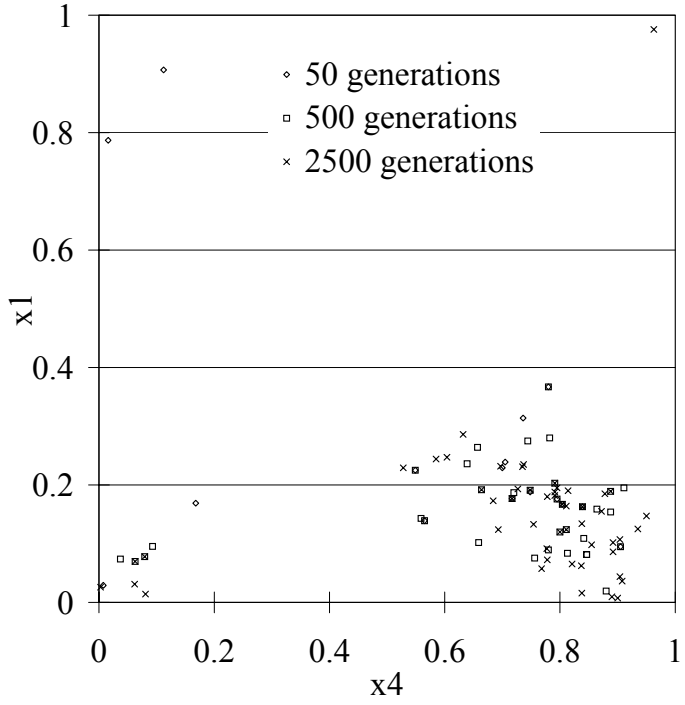
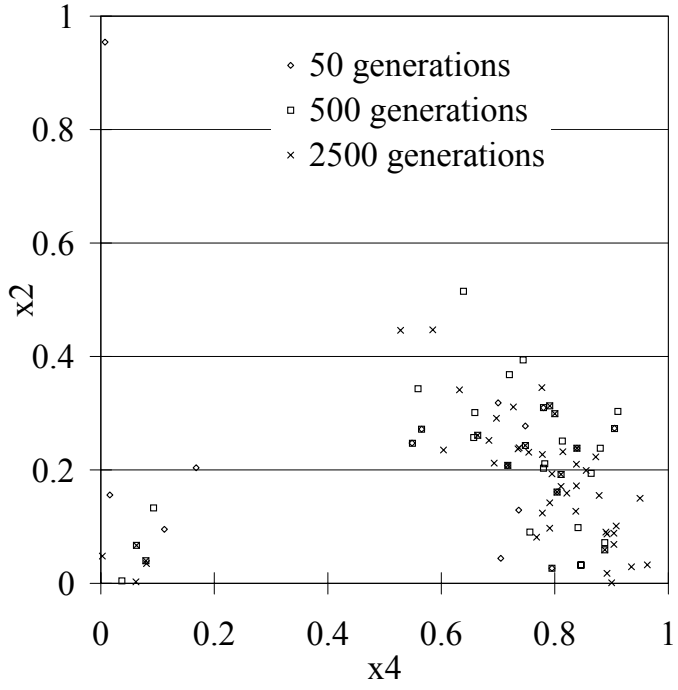


Fig. 15 Pareto-optimal solutions of Example II in function space

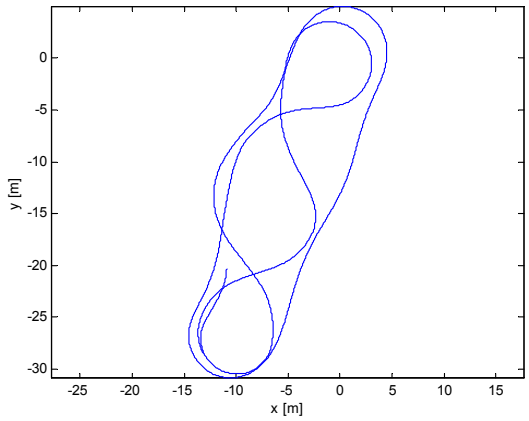


(a) x_1 - x_4 space

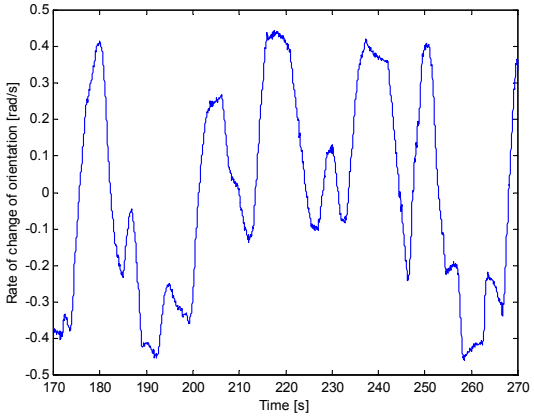


(b) x_2 - x_4 space

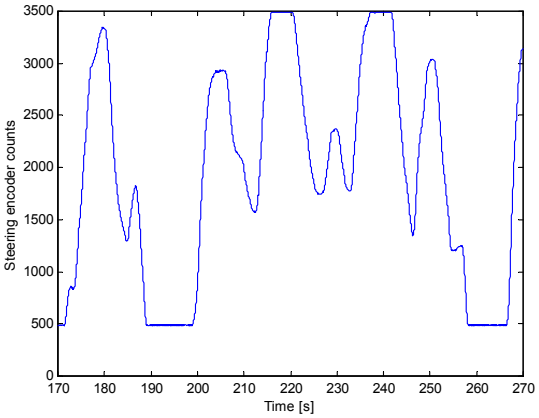
Fig. 16 Pareto-optimal solutions of Example II in parameter space



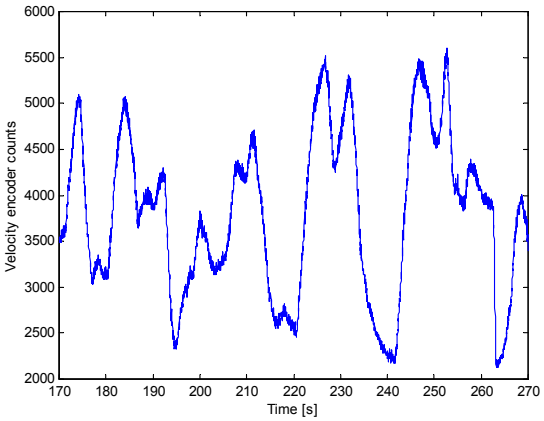
(a) GPS sensor data



(b) Gyro sensor data



(c) Steering encoder reading



(d) Velocity encoder reading

Fig. 17 Measurements of autonomous vehicle

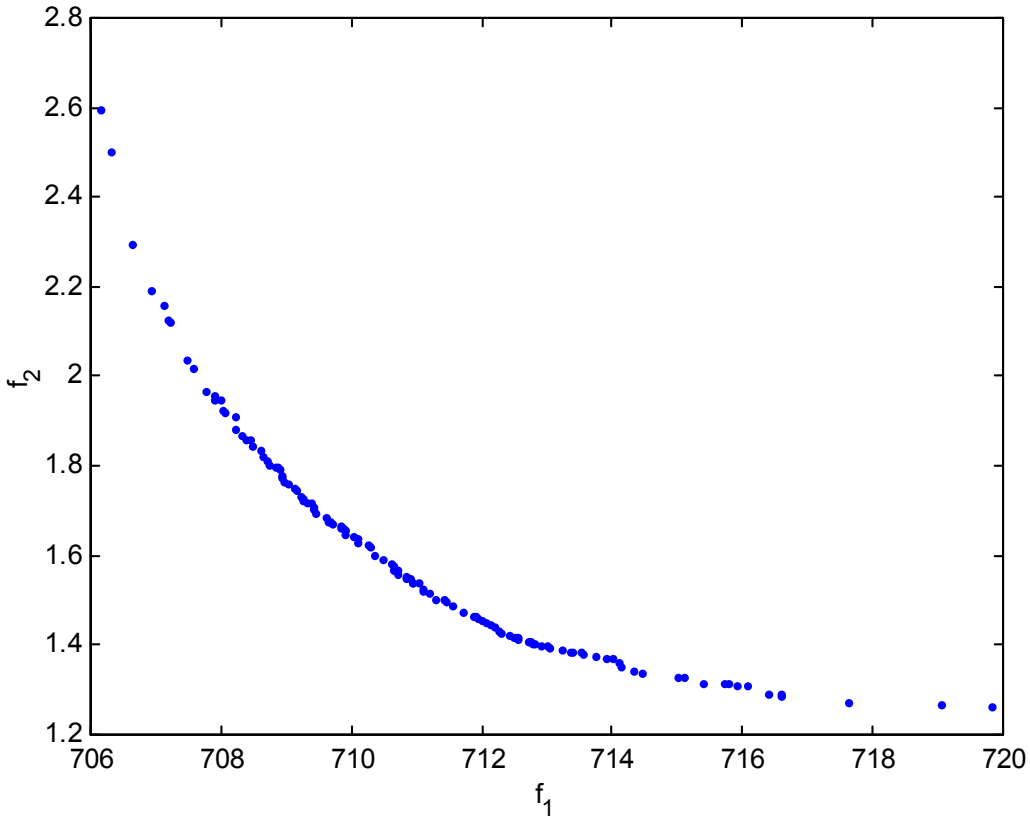
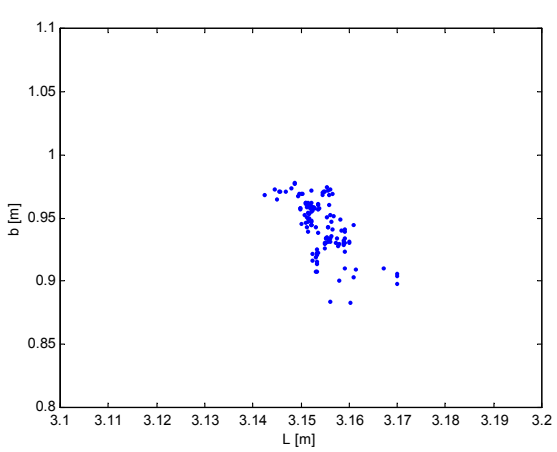
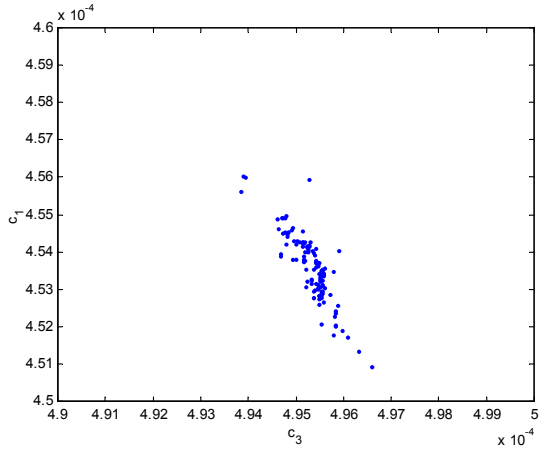


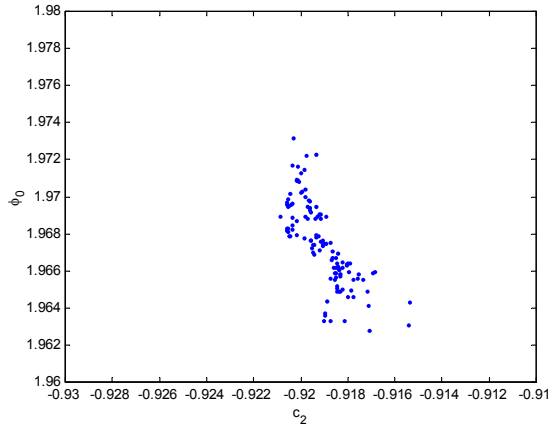
Fig. 18 Pareto-optimal solutions of identification in function space



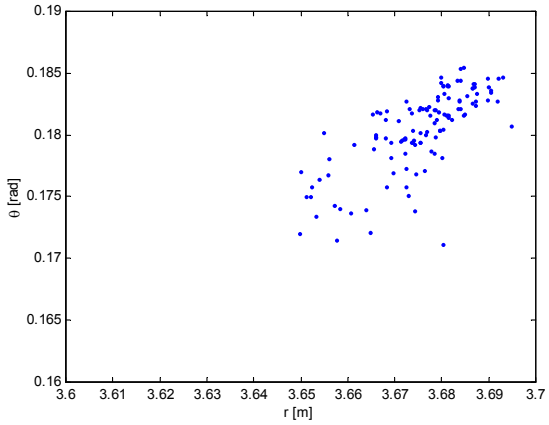
(a) L - b space



(b) c_3 - c_1 space



(c) c_2 - ϕ_0 space



(d) r - θ space

Fig. 19 Pareto-optimal solutions of identification in parameter space

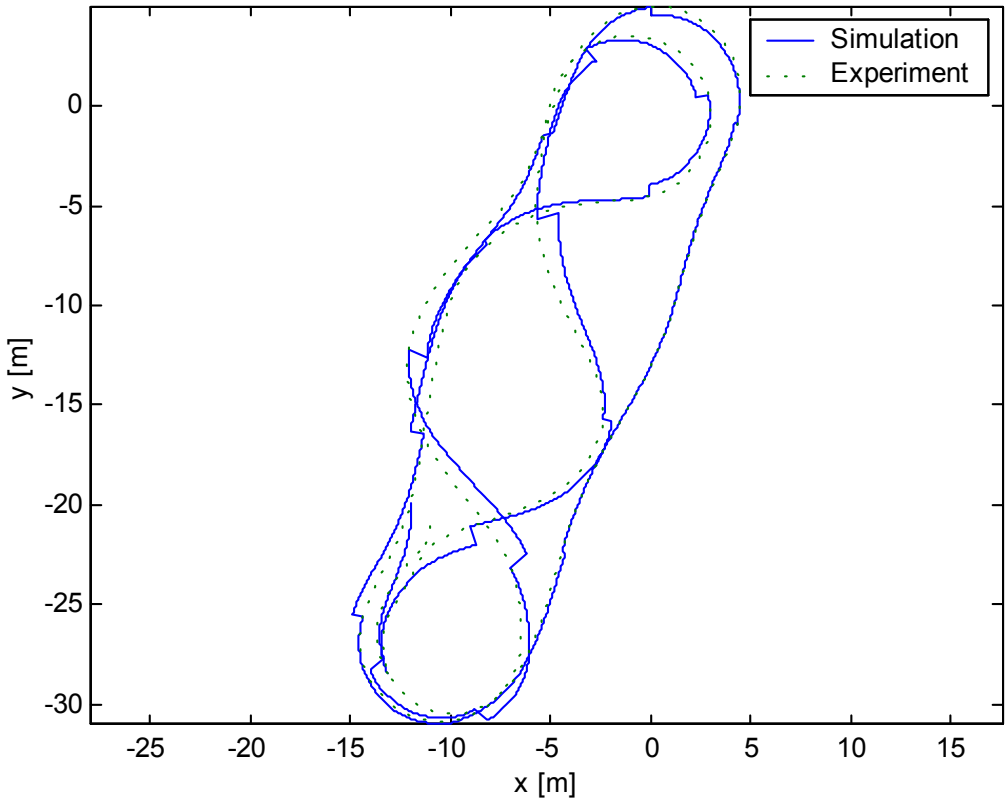
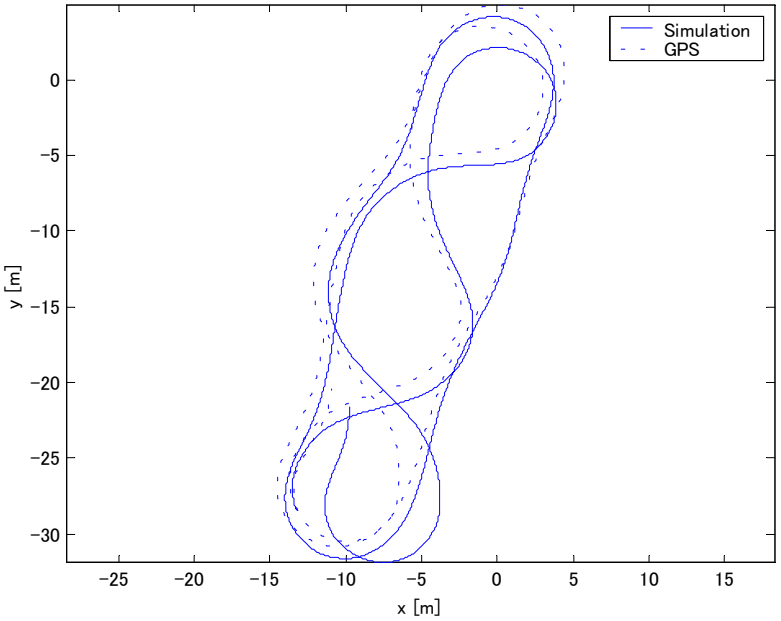
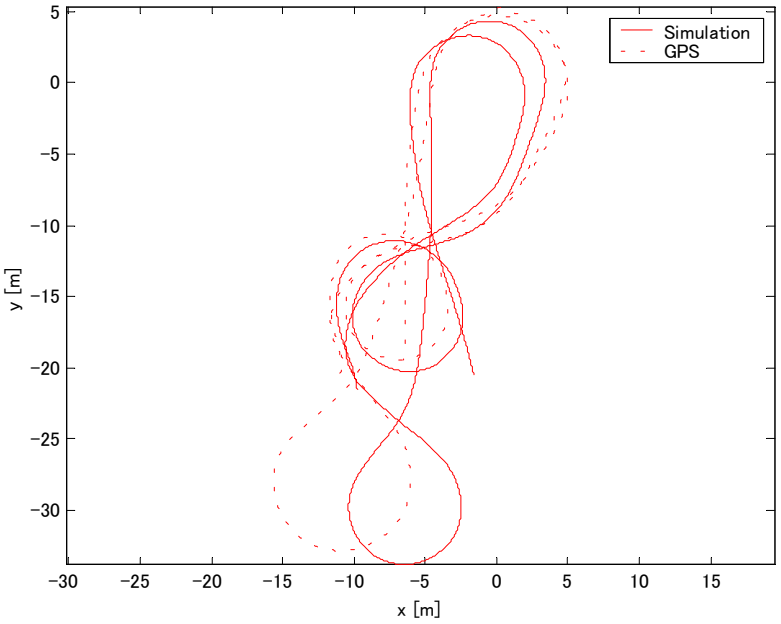


Fig. 20 Simulation results with parameters chosen

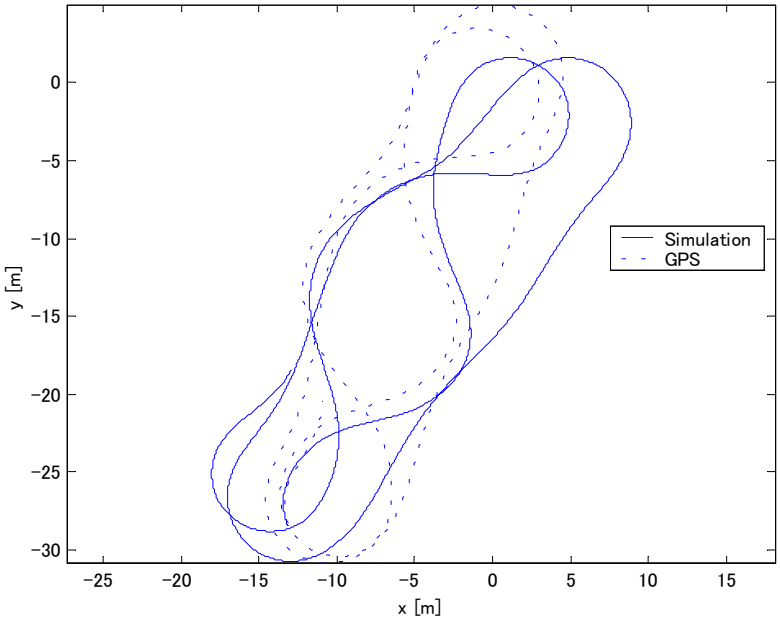


(a) 1st 100 seconds

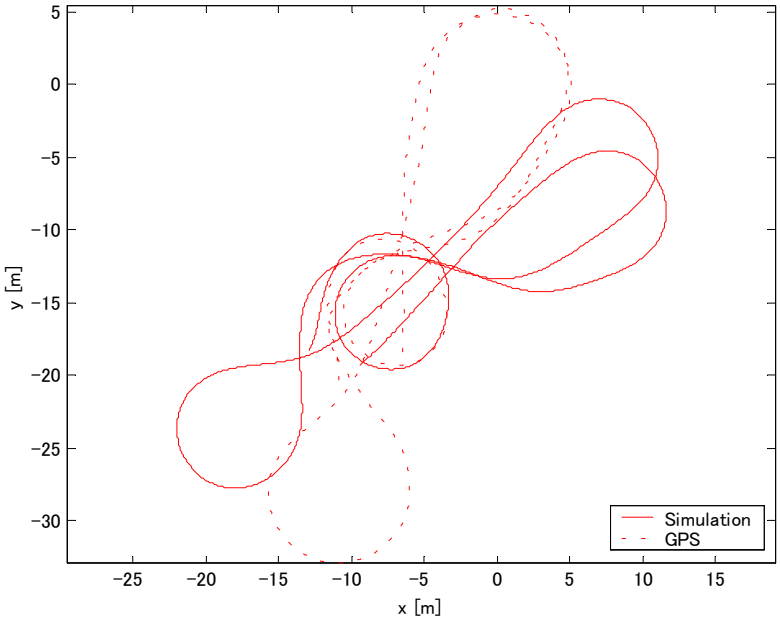


(b) 2nd 100 seconds

Fig. 21 Non-partitioned simulation results with parameters chosen

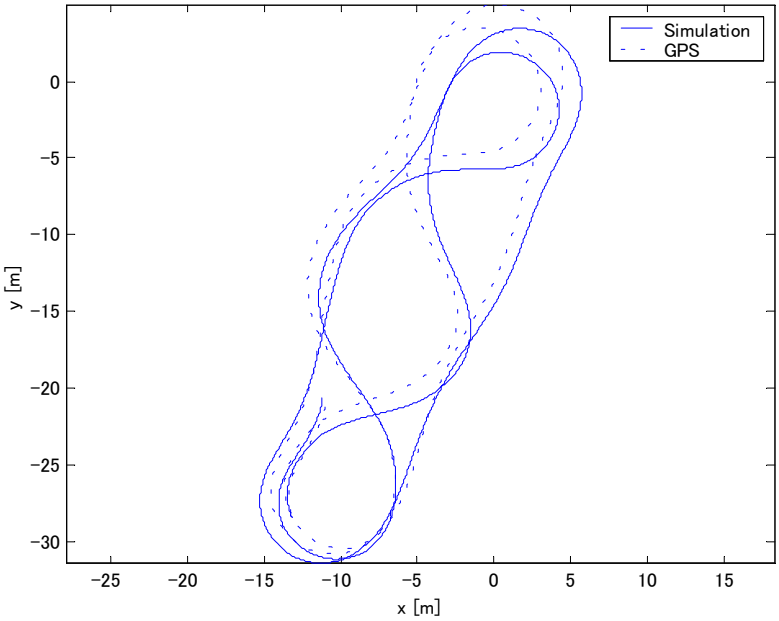


(a) 1st 100 seconds

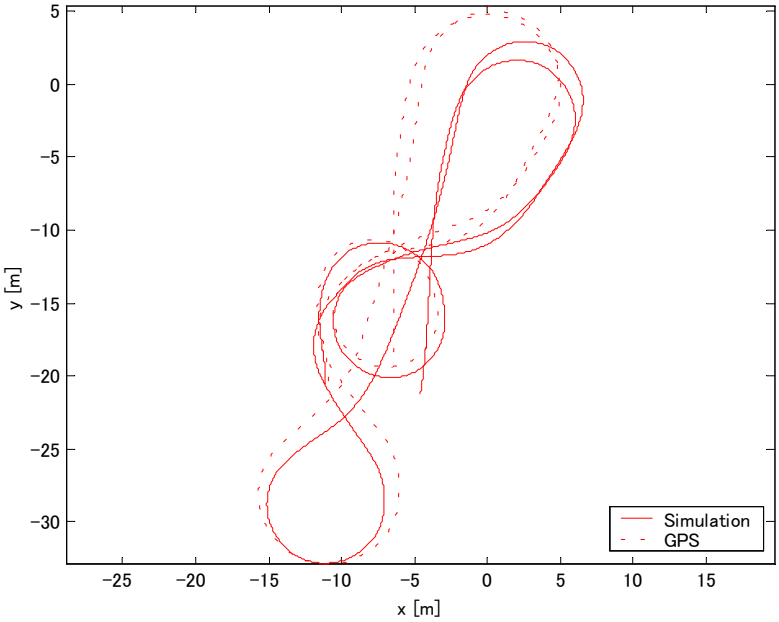


(b) 2nd 100 seconds

Fig. 22 Non-partitioned simulation results with minimum position error parameters



(a) 1st 100 seconds



(b) 2nd 100 seconds

Fig. 23 Non-partitioned simulation results with minimum orientation error parameters