Robotica (2007) volume 25, pp. 479–492. © 2007 Cambridge University Press
doi:10.1017/S0263574707003359 Printed in the United Kingdom

A neural-network approach to high-precision docking of autonomous vehicles/platforms

Joseph Wong†, Goldie Nejat‡*, Robert Fenton† and Beno Benhabib†

†*Computer Integrated Manufacturing Laboratory, Department of Mechanical and Industrial Engineering University of Toronto, 5 King's College Road, Toronto, M5S 3G8, Ontario, Canada*
‡*Department of Mechanical Engineering, State University of New York at Stony Brook, Stony Brook, 11794–2300, New York, USA*

(Received in Final Form: January 1, 2007. First published online: February 13, 2007)

SUMMARY
In this paper, a Neural-Network- (NN) based guidance methodology is proposed for the high-precision docking of autonomous vehicles/platforms. The novelty of the overall online motion-planning methodology is its applicability to cases that do not allow for the direct proximity measurement of the vehicle's pose (position and orientation). In such instances, a guidance technique that utilizes Line-of-Sight- (LOS) based task-space sensory feedback is needed to minimize the detrimental impact of accumulated systematic motion errors. Herein, the proposed NN-based guidance methodology is implemented during the final stage of the vehicle's motion (i.e., docking). Systematic motion errors, which are accumulated after a long-range motion are reduced iteratively by executing corrective motion commands generated by the NN until the vehicle achieves its desired pose within random noise limits. The proposed guidance methodology was successfully tested via simulations for a 6-dof (degree-of-freedom) vehicle and via experiments for a 3-dof high-precision planar platform.

KEYWORDS: Online motion planning; Localization; Line-of-sight guidance; Neural networks.

1. Introduction

Navigation of autonomous vehicles/platforms (hereafter referred to as simply "*vehicles*") has been the subject of intensive research in the past two decades. Two-stage motion-execution strategies have been commonly proposed. Namely, after an initial *gross*, but fast motion of the autonomous vehicle toward its desired goal, the use of external sensory feedback has been advocated for *high-precision* localization. In this context, *docking* refers to the online autonomous guidance of the vehicle during the secondary high-precision localization stage for the purpose of achieving a desired pose (position and orientation) within required tolerances.[1] This *fine* corrective-movement stage is normally necessary in order to compensate for the systematic motion errors that might have accumulated during the initial gross motion of the vehicle. The pertinent literature is briefly reviewed below in the context of two important motion-planning issues (*i*) sensing resolution and accuracy, as well as (*ii*) ability to acquire and process sensory data in real-time for effective online guidance.

*Corresponding author. E-mail: Goldie.Nejat@stonybrook.edu

1.1. Sensing

As a particular example of the implementation of autonomous docking, the use of visual servoing was proposed by Lee *et al.* for the guidance of an Autonomous Underwater Vehicle (AUVs) toward a stationary underwater station for recharging the energy system by minimizing a cost function that compares the optical flow equation of the charge-coupled device (CCD) camera and the linearized equation of motion of the AUV.[2] Other common autonomous-docking implementations that have been discussed in the literature include: homing of Automated Guided Vehicles (AGVs) in office settings[3,4] or in manufacturing plants and warehouses for material transportation,[5] automobile parking,[6,7] docking of rapid-transit buses to a predefined platform,[8,9] docking of ships to a stationary pier/berth,[10,11] docking of spacecrafts to a space station/satellite for automated resupply, refuel, maintenance, and/or disposal of wastes,[12–14] aligning and merging of connections on self-configurable robotic modules,[15,16] and pregrasping/positioning of a robotic gripper/hand relative to a stationary or moving object.[17,18]

For autonomous ground vehicles, the most commonly used motion-sensing system has been odometry.[19–22] Odometry uses integration of incremental motion, thus allowing systematic errors to accumulate over time. In particular, unbound lateral positional errors can result from small errors in orientation. Inertial Navigation Systems were initially developed for aerospace applications. Accelerometers could be used to obtain linear displacement by integrating the acceleration information twice. Gyroscopes integrate the rate of rotation of a vehicle to determine orientation information and have been proposed to compensate for orientation errors. However, both accelerometers and gyroscopes are subject to temperature-dependent drift problems and the measurement errors are unbounded due to the nature of the incremental readings.[23,24] Thus, one can state that, in general, self-contained internal sensors do provide *good* short-term

https://doi.org/10.1017/S0263574707003359 Published online by Cambridge University Press

accuracy and permit *high*-sampling rate, however, they could be subject to relatively large systematic and nonsystematic errors in their measurements.

External, noncontact task-space sensors have been frequently proposed for vehicle guidance in order to reduce the detrimental impact of systematic motion errors and achieve high accuracies via closed-loop control. Computer-vision based systems have been widely adopted and provide comprehensive information about the vehicle's surroundings, when compared to other sensing technologies.[6,21,25–26] However, the negative influence of external parameters (e.g., lighting) and extraction of useful data may pose an operational bottleneck, and thus, limit accuracies for real-time and outdoor applications.[27] On the other hand, Differential Global Positioning Systems (DGPSs) work well in outdoor environments in real-time, but, they currently yield only at best centimeter-order resolution.[28] Furthermore, DGPSs may not operate effectively in indoor environments, tunnels, and urban areas with tall buildings. Underwater applications are also not suitable for DGPSs since radar waves do not propagate in water and the data-flow could be interrupted, it is insufficient for the operation of DGPSs unless equipped with an acoustic positioning system.[29,30]

Several types of proximity sensors have also been proposed for the direct task-space measurement of vehicle positions based on electromagnetic, acoustic, and electrooptical transducers.[31] Both inductive and capacitive electromagnetic sensors have limited measuring ranges[32,33] (several millimeters). Even though Acoustic sensors, on the other hand, have larger measurement ranges, they would be subjected to a wider angle of divergence.[33] Electrooptical sensors are the most suitable proximity sensors for autonomous vehicle navigation and can be configured to yield high accuracies over relatively large distances.[34]

1.2. Online motion planning

All above-mentioned sensing systems rely on the relative or absolute measurement of a target's (vehicle's and/or dock's) proximity. Frequently, however, a target's pose cannot be estimated accurately due to our *inability* to measure its orientation as precisely as its position. Thus, for applications where orientation accuracy is as vital as that of position, a *better* sensing system and a corresponding docking strategy are needed for autonomous online guidance (i.e., motion planning). In response to this need, in the past, we have proposed two such novel autonomous-vehicle guidance algorithms that utilize a Line-of-Sight (LOS) based task-space sensing system, in the latter stage of a two-stage motion-planning strategy, for indirect proximity estimation.[35–37]

The first type of the proposed guidance algorithms requires the use of an accurate analytical calibration model of the sensing system that employs high-precision external measurement devices (e.g., interferometers).[35] Furthermore, the calibration procedure can only be extended to vehicles with up to 5-Degree-of-Freedom (dof) mobility and periodic recalibration may be needed for accurate docking due to the sensitivity of guidance algorithms to errors in the calibration model. In general, due to the nature of the (indirect) measurements taken by our task-space sensing system, self-calibration is not feasible, hence, external measurement devices need to be utilized for parameter identification. However, it is difficult to develop such an accurate model for a system with a large number of dof due to the inability of current sensors to directly measure calibration parameters within the desired tolerances, in particular orientation angles. The task is further complicated for sensing-system models that are highly nonlinear, and when the vehicle has a nontrivial geometric shape. Hence, a model-independent docking algorithm would be needed.[36] In the past, we have developed gradient descent-based model-independent guidance methods (i.e., the second type of the proposed guidance algorithms),[37] however, these methods are slow to converge and can exhibit oscillatory behavior.

In this paper, we present a generic online motion-planning methodology that utilizes a Neural Network (NN) based sensing-system localization model. Both a Kalman Filter (KF) and a traditional controller require a priori information of the calibration model to either estimate the model via an optimal stochastic estimator or directly utilize the model (exact analytical equations) to determine control parameters. Furthermore, a KF requires direct measurements of the state in order to obtain a state-estimate, and cannot be utilized in situations where the actual pose of the vehicle is not known and cannot be directly measured by sensors. The proposed NN-based guidance algorithm successfully copes with the drawbacks of the above-mentioned *model-dependent* and *model-independent* (gradient descent type) methodologies. The main advantage of the proposed novel methodology is its ability to effectively guide an autonomous vehicle to its desired pose with as few corrective actions as possible without the need of any a priori calibration model information. Online attainable data are utilized to minimize the NN training set by enhancing system performance.

1.3. NNs

NNs have been in the past used in the motion planning of autonomous mobile vehicles. For example, sonar-ring sensors and a multilayer NN were used (i) to generate a path for a robotic vehicle while avoiding obstacles,[38] and (ii) to localize the pose of an autonomous vehicle by comparing a set of range readings with an *a priori* known environment map.[39] NNs have also been used to *learn* the systematic errors of a robot by performing offline calibration using a Coordinate-Measurement Machine (CMM). The CMM either directly measures the actual poses of the robot, or does so indirectly by constraining the end-effector into premeasured insertion holes.[40,41] Two-dimensional pose estimation using images from a CCD camera via NNs was also proposed.[42] Despite intensive efforts, most of the methods cannot compensate for unpredictable errors such as wheel slippage and floor irregularities in an autonomous vehicle application, particularly, in situations where the vehicle is required to make many turns in high-speed maneuver, unless some real-time sensory information is employed.

In the above context, the objective of this paper is to present a generic (sensing-system) *calibration-model-independent* LOS-based motion-guidance technique that utilizes background NN learning. The proposed NN model,

https://doi.org/10.1017/S0263574707003359 Published online by Cambridge University Press

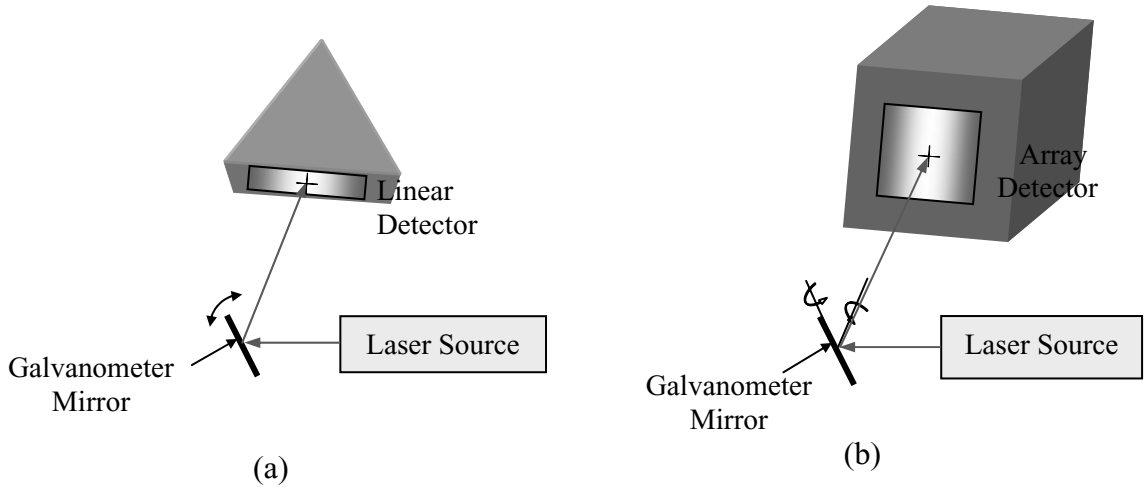


Fig. 1. (a) Planar and (b) spatial LOS-based sensing modules.

which is trained offline and modified online, has been shown to be very effective in generating corrective motion commands in order to *guide* the vehicle to its desired final pose within random noise. The overall system is applicable in situations where direct pose measurement of the vehicle is not available or difficult to accurately determine due to the limited capabilities of task-space sensors. In Section 2, we first present an overview of the task-space LOS sensing system, first suggested in,[1] and, then, briefly discuss the NN architecture used in our proposed guidance strategy. In Section 3, the generation of the NN sensing-system model and its implementation are explained in greater detail. The results of some of the motion-planning simulations conducted on a 6-dof vehicle are also given in this section. Experiments conducted on a 3-dof high-precision platform, utilizing our proposed approach are presented in Section 4. The conclusions of our work are discussed in Section 5.

2. System Overview

The proposed three-dimensional NN-based guidance methodology incorporates a multi-LOS sensing system for navigating the autonomous vehicle to its desired pose. Herein, we define a *vehicle* to be any mechanical platform capable of multi-dof motion, i.e., ground vehicles including mobile robots, spacecrafts and satellites, underwater vehicles, self-reconfigurable robots, etc. The NN model defines a relationship (approximate) between the LOS geometric information and the required task-space corrections necessary to guide the vehicle.

2.1. LOS task-space sensing system

Docking problems have typically been tackled by employing external task-space sensing systems to directly measure the vehicle's pose relative to a reference coordinate system. As noted in Section 1, however, a vast majority of task-space sensory systems to date either have only a limited working range or cannot estimate the vehicle's orientation as accurately as its position. In response to this drawback, the LOS concept utilized in our work for autonomous vehicle guidance via indirect proximity estimation employs a collection of (LOS-based) sensing modules, each comprising a laser source, a galvanometer mirror, and a detector (e.g., Position Sensitive Detector (PSD)). The detectors are mounted on the vehicle, while the laser beams, which define the desired LOS, are aligned using galvanometer mirrors to hit the center of their respective detectors when the vehicle is at its *desired* pose.

A multi-LOS system can be configured using several single-LOS modules. The minimum necessary number and the types of LOS in configuring such a system would depend on (i) the mobility requirement for the docking problem at hand and (ii) the motion range of the vehicle. For example, a planar single-LOS system would be sufficient to dock a vehicle at a point on a straight line, rail-type path in two-dimensional (2D) space, while docking in three-dimensional (3D) space (e.g., a satellite or a submarine) would require a system with three spatial LOSs, Fig. 1(a) and (b), respectively. A planar LOS requires one 1-dof galvanometer mirror to redirect the laser beam, whereas a spatial LOS can be achieved by two 1-dof galvanometer mirrors providing motion in two orthogonal axes.

The advantages of the proposed LOS sensing-system includes: (i) applicability to varying vehicle mobility requirements, (ii) accurate measurement even in the presence of poor lighting and weather due to optical characteristics of the laser, and (iii) small and lightweight attachments to the vehicle.

2.1.1. Proximity estimation. In a two-stage motion scheme, it is assumed that the autonomous vehicle is initially moved to its desired pose by using closed-loop feedback from the actuator-level sensors. This action is regarded as *long-range positioning*. External, task-space sensors are then used to improve the vehicle's accuracy (i.e., minimize systematic errors) via effective guidance during the second s*hort-range positioning* stage (i.e., docking).

In our sensing system, the mirrors are aligned such that the spatial LOSs would simultaneously hit the centers of their respective PSDs at the vehicle's desired pose. Inaccurate long-range positioning of the vehicle, due to systematic and random unpredictable motion errors would, however, result

https://doi.org/10.1017/S0263574707003359 Published online by Cambridge University Press

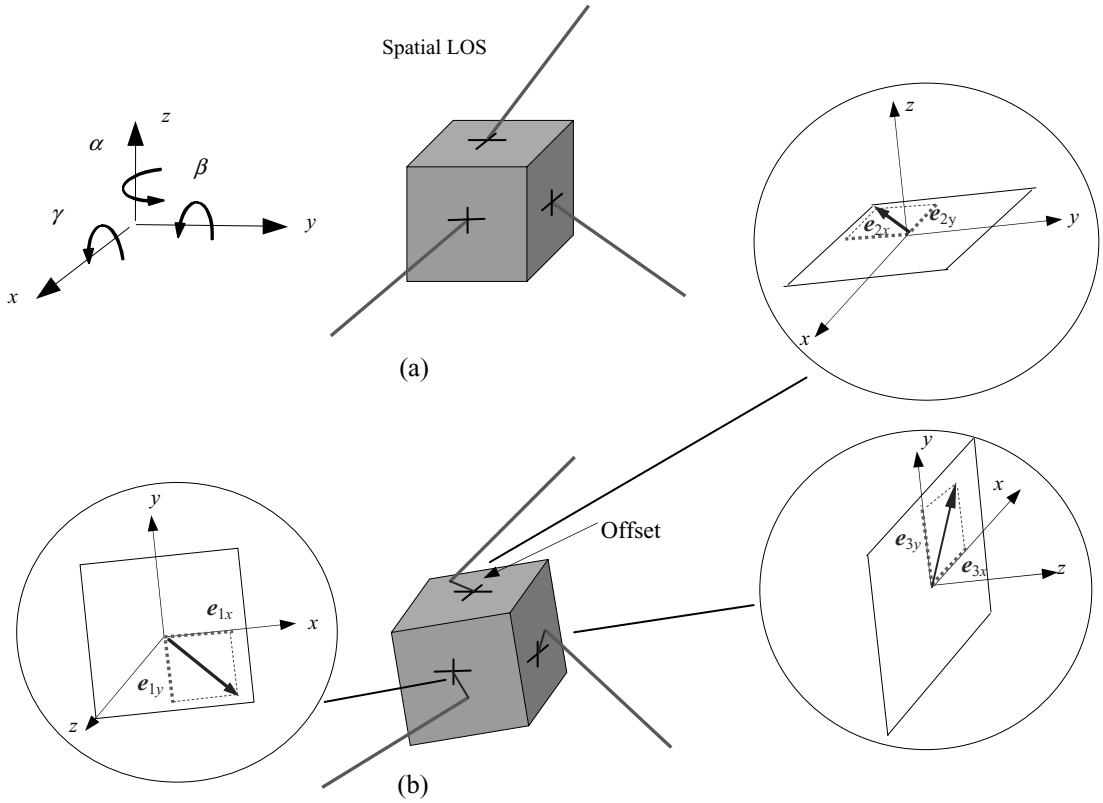


Fig. 2. (a) Desired vehicle pose and (b) actual vehicle pose.

in offsets on the PSDs. These offsets are defined in their own individual detector's coordinate frame and not in a global world frame of reference. The offsets are utilized herein as *indirect estimates of proximity* to guide the vehicle to its desired pose from the inaccurate actual pose, during the short-range positioning stage. In this paper, for maximum generality, we assume that the vehicle's shape is a cube with 6-dof mobility (x, y, z, γ, β, α). Three array type detectors are placed on at least three of its orthogonal faces, Fig. 2.

2.2. NN-based sensing-system model

For (*short-range*) guidance (i.e., docking), the use of NNs is proposed here to predict the necessary incremental corrective Point-to-Point (PTP) vehicle motion in an online manner based on the measured PSD offsets and the LOS angles that correspond to the desired pose. This is an iterative process that continues until the vehicle converges to its desired pose within the required tolerances in terms of PSD offsets. Successive iterations of incremental corrective motion result in the minimization of the PSD offsets, which indirectly reduces the difference between the vehicle's actual and desired poses.

The two categories of NN sensing-system models used in our work are: *global* and *local*. The *Global* NN (GNN) model is trained by collecting measurement data at a finite number of R possible docking poses of the vehicle within its workspace via a *teaching-by-demonstration* method. A data point, $r = 1$, R, is obtained as follows: the LOSs are aligned and pretaught as the desired angles, $(\boldsymbol{\phi}_{\text{desired}})_r$ ($3 \times 2 = 6$ angles), corresponding to the vehicle's desired pose. Then, the vehicle is commanded to move a small incremental motion, $(\Delta \boldsymbol{x})_r$ (6 displacement parameters), to emulate imperfect docking and the resultant PSDs offsets, $\boldsymbol{e}_r$ ($3 \times 2 = 6$ error parameters), are measured.

After long-range positioning of the vehicle, the estimation of the necessary corrective motion of the vehicle toward the desired pose is achieved via the task-space sensory feedback and the online implementation of the GNN model. During run-time, the GNN model is concurrently improved through online training by means of incremental adaptation of the NN parameters. In each step of the implementation of the GNN, the performed incremental corrective motions and the resultant changes in the PSDs readings are recorded, and subsequently used to update the GNN parameters.

In cases where the vehicle's docking operation is to be *repeated* at other times, improvement on the convergence rate can be achieved by using a *Local* NN (LNN) sensing-system model corresponding to this specific docking pose. LNNs can be trained using predicted data samples from the GNN and/or online data that are acquired in previous successful docking attempts. In subsequent docking attempts at this specific docking pose, only the LNN is used to predict the necessary guidance motions toward the desired pose.

2.2.1. NN architecture and learning. The NNs used in our work have standard multilayer feed-forward architectures, with one hidden layer of tangent sigmoidal neurons and an output layer of linear neurons. Conventional batch learning is used for offline modeling. A regularization approach is essential for the batch training of a NN, in order to determine a good balance between bias and variance. Gauss–Newton approximation to Bayesian Regularization (GNBR)

https://doi.org/10.1017/S0263574707003359 Published online by Cambridge University Press

algorithm does not require a validation set as in early stopping and its benefit is prevailing when scarcity of data exists, but yet, suffers from excessive computation overhead.[43] On the other hand, scaled conjugate gradient utilizes a step-size scaling mechanism that avoids time-consuming line search, and is generally faster than other standard second-order algorithms including the standard gradient-descent method.[44] In this paper, the scaled conjugate gradient with early stopping and a GNBR algorithm are employed in the offline modeling of the sensing system for the 6-dof simulation and 3-dof experiment of vehicle-docking problems, respectively.

Advanced optimization techniques, which can be used in batch-mode learning are not applicable for incremental-mode learning.[45] Hence, a standard gradient-descent algorithm is employed here to optimize the interconnection weights for online learning.

A multilayer NN with standard feed-forward architecture is well-known for its ability to perform continuous, complex, and nonlinear mappings in multidimensional spaces in any degree of accuracy. For dealing with high-precision localization problems, however, obtaining adequate offline learning samples and training a high-complexity NN model may become intractable due to time constraints. Our proposed approach minimizes the required number of training data for high-precision docking by utilizing (i) an iterative approach, (ii) online sensory data as learning samples, which are normally available as part of the docking process, and (iii) local sensing-system models for repetitive docking to desired poses.

3. Docking Via A NN Model

The generic 3D NN guidance methodology presented here incorporates multi-LOS sensing with corrective motion actions that the vehicle needs to undertake in order to achieve its desired pose within required tolerances (typically, governed by the system's random noise). As noted above, the proposed guidance methodology does not require the existence and the use of a sensing-system's analytical calibration model. In this section, the development and implementation of the GNN and LNN sensing-system models are described. We assume the vehicle's shape to be a cube and have the most generalized 6-dof mobility (x, y, z, γ, β, α), with array-type detectors on at least three of its orthogonal faces. In practice, after the long-range positioning stage of the vehicle, offsets would be registered on the detectors due to systematic and random motion errors, as shown in Fig. 2.

3.1. GNN sensing-system model

The objective of an offline trained GNN sensing-system model is to predict the vehicle's necessary corrective docking after long-range positioning, given the sensory feedback from PSD detectors and the desired pose information from the LOS angles. GNN is also expected to provide a *start-up model* that can be improved in run-time for enhancing accuracy and convergence rate.

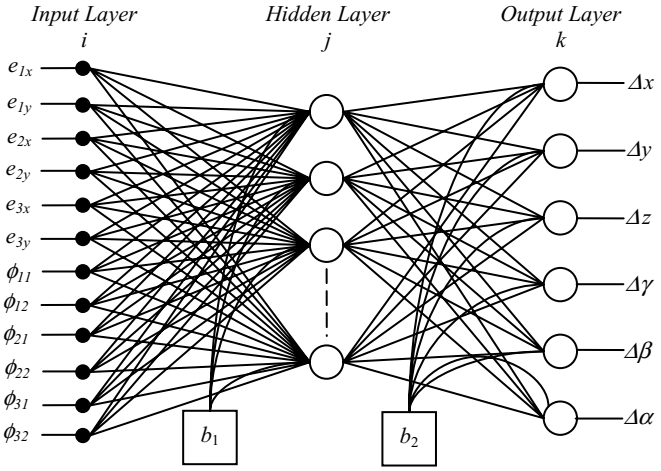


Fig. 3. GNN sensing-system model architecture.

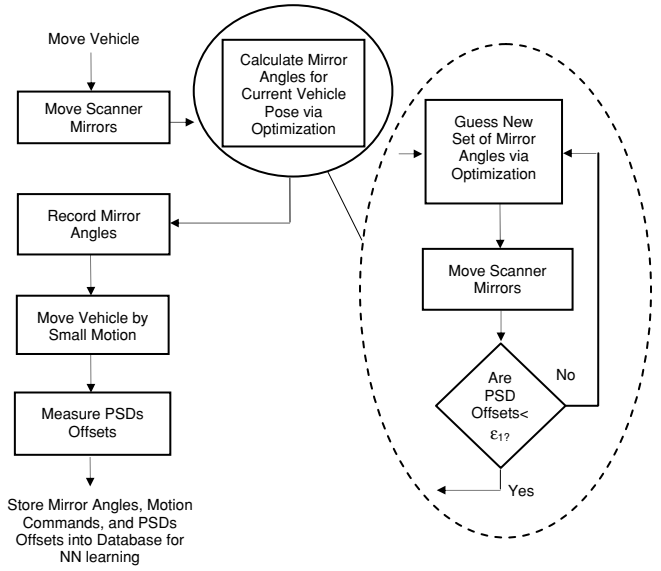


Fig. 4. Data-collection framework.

3.1.1. Development of the model. The GNN sensing-system model requires knowledge of the desired pose, $(\boldsymbol{x})_{\text{desired}}$, and the deviations between the vehicle's desired and actual poses at time of docking in order to estimate and output the necessary incremental corrective motion commands (Δx, Δy, Δz, $\Delta \gamma$, $\Delta \beta$, and $\Delta \alpha$) that would guide the vehicle toward its desired pose. The desired pose is specified by the LOS angles, ϕ_{mn} ($m = 1$ to 3 and $n = 1$ to 2), and the errors in docking are specified by the detectors' offsets, e_{ij} ($i = 1$ to 3 and $j = x, y$). The schematic of the GNN model is shown in Fig. 3.

Discrete measurement data are required for the training of the GNN to establish the relationship between the sensory information and the incremental motion commands, as described below and summarized in Fig. 4. Each data point contains the tuples $(e_{ij}, \phi_{mn}, \Delta x, \Delta y, \Delta z, \Delta \gamma, \Delta \beta, \Delta \alpha)_r$, obtained by the following procedure:

(i) Move the vehicle to a random (desired) docking pose within the workspace, and concurrently, align the individual scanners such that each LOS would hit as near

https://doi.org/10.1017/S0263574707003359 Published online by Cambridge University Press

its respective PSD's center as possible at the vehicle's desired pose.

(ii) Realign the LOSs, ϕ_{mn}, after the vehicle reaches its *actual* pose, using an active search method (e.g., the Flexible Polyhedron method in our case)[46] that determines the optimal scanner angles, which correspond to minimal offset values registered on the detectors. The search for each LOS's two scanner angles occurs independently and concurrently. These final orientations of the LOSs are related to the corresponding *actual* vehicle pose (within noise limits).

(iii) Move the vehicle by a small random incremental displacement (Δx, Δy, Δz, $\Delta\gamma$, $\Delta\beta$, $\Delta\alpha$) to emulate imperfect docking and measure the resultant PSD offsets, e_{ij}, on each of the detectors, while keeping the LOSs locked.

After the collection of a set of discrete measurement points, the data set is split into *training* and *testing* subsets (a *validation* subset is also required if early stopping is used). An additional set of constraining data, equivalent to the number of training data, is appended to the training subset to train the NN. The constraining set has identical LOS angles as the training set, but has null PSD offsets and incremental motion $\{\phi_{mn}(m=1 \text{ to } 3 \text{ and } n=1 \text{ to } 2),\ e_{pq}=0\ (p=1 \text{ to } 3 \text{ and } q=x,\ y),\ \Delta x=0,\ \Delta y=0,\ \Delta z=0,\ \Delta\gamma=0,\ \Delta\beta=0,\ \Delta\alpha=0\}_r$, $r=1$ to R, where R is the total number of training points. The constraining set implies that when the vehicle is at its desired pose, theoretically, the offset readings are also zero, and no corrective motion is needed. Thus, the purpose of the constraining set is to compel the GNN regression model to output a zero incremental corrective command when the LOSs hit the centers of their respective PSDs.

In order to lessen the effect of variability between values in the data, the inputs and outputs are normalized in the region of $[-1, 1]$. In our study, a growing scheme is used to determine the network size; namely, we start with building a model that is relatively small in size and add hidden neurons if the performance of the testing data is unsatisfactory. Given the probable docking workspace of the vehicle, the increase in measurement points for training can enhance (i) the desired accuracy of the docking vehicle and (ii) the rate of convergence to the desired accuracy.

3.1.2. Implementation. The proposed localization (docking) methodology is based on the estimation of the current pose of the vehicle by utilizing indirect proximity measurements after its long-range motion, in order to determine the necessary corrective motions via the use of a NN. The desired vehicle pose and its corresponding LOSs are predefined. Once the vehicle has reached its actual pose upon completion of the long-range positioning stage, the PSD offsets are measured and fed into a *temporary* GNN. The temporary GNN is initialized with the identical architecture and interconnection weights as the original GNN model. The temporary GNN, then, subsequently predicts the required corrective motions to minimize the PSDs offsets.

The motion commands and the resultant *changes* in the PSD offsets, due to the execution of these corrective actions, are used as the training sample of the temporary NN in an online manner: Each online learning sample contains the tuples (Δe_{ij}, ϕ_{mn}, Δx, Δy, Δz, $\Delta\gamma$, $\Delta\beta$, and $\Delta\alpha$). Namely, after each execution of the predicted corrective commands, a learning tuple would be generated and presented to the temporary GNN for the online incremental learning using a gradient-descent algorithm. Although the training tuples do not provide exact information about the necessary corrective motion commands, they do provide an adequate approximation, whose *goodness* progressively improves as the vehicle approaches its desired pose. Due to the nature of a connectionist learning system, the succession of local interconnection weight corrections may not improve its global performance. Hence, the temporary NN is only employed for the docking task at hand and discarded upon its completion.

Accordingly, the implementation stage includes the concurrent prediction of the corrective vehicle motions and online fine-tuning of the temporary GNN interconnection weights. The imperfections of the trained network necessitate the use of an iterative procedure to achieve a desired accuracy. Moreover, successive iterations cause the diminishing of the magnitude of the systematic errors of the vehicle due to the decrease of the vehicle's motion in the short-range motioning phrase. Docking is considered to be successful if all of the PSDs offsets converge within bounds of the random noise level of the system, ε_2, as shown in Fig. 5.

3.2. LNN sensing-system model

In situations where a vehicle is to repeatedly dock at a desired pose, a LNN model can be trained offline and utilized subsequently. The objective is to enhance the convergence rate by using acquired information from past docking *experiences*. Consequently, the LNN model would be used in lieu of the GNN model for this specific pose, when available.

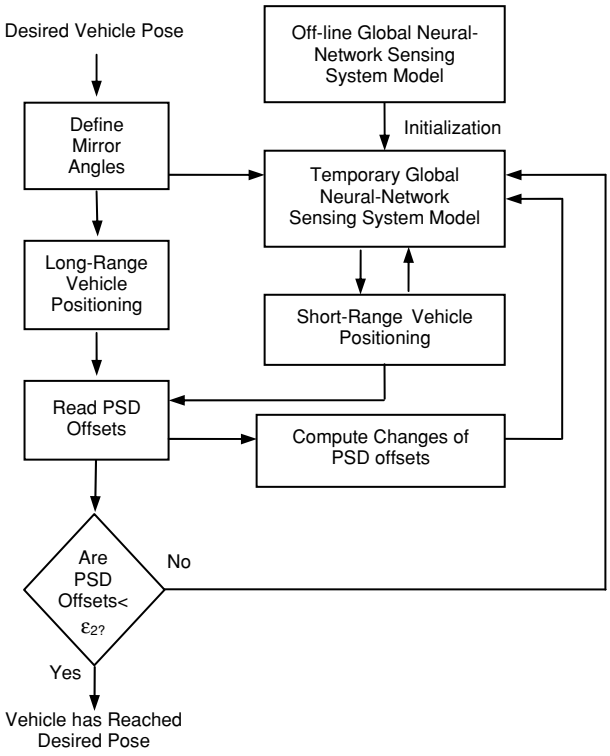


Fig. 5. Implementation of GNN.

https://doi.org/10.1017/S0263574707003359 Published online by Cambridge University Press

Given that the LOS angles, ϕ_{mn}, for the same desired pose are fixed, the inputs to the local model that vary are solely the PSDs offsets, e_{ij}. The outputs of the LNN model as that for GNN model, are the incremental corrective motion commands (Δx, Δy, Δz, $\Delta\gamma$, $\Delta\beta$, $\Delta\alpha$).

The learning tuples for a LNN (e_{ij}, Δx, Δy, Δz, $\Delta\gamma$, $\Delta\beta$, $\Delta\alpha$) can be obtained via previous docking experiences at that particular pose. Namely, upon the completion of a docking task (i.e., where the detectors' offsets are successfully minimized to the random-noise level), a set of data points whose size equals to the number of taken corrective actions, A, are used for the training of LNN:

$$\begin{aligned} &e_{ij}^{a}, \\ &\Delta x^{a} = \sum_{i=a}^{A} \Delta x^{i}, \\ &\Delta y^{a} = \sum_{i=a}^{A} \Delta y^{i}, \\ &\Delta z^{a} = \sum_{i=a}^{A} \Delta z^{i}, \qquad a = 1 \text{ to } A \\ &\Delta\gamma^{a} = \sum_{i=a}^{A} \Delta\gamma^{i}, \\ &\Delta\beta^{a} = \sum_{i=a}^{A} \Delta\beta^{i}, \\ &\Delta\alpha^{a} = \sum_{i=a}^{A} \Delta\alpha^{i}, \end{aligned} \tag{1}$$

The learning tuples acquired in previous docking experiences are stored in a database for the future development of a LNN. In practice, one may not be able to build a LNN on request during the operation of the system due to operational constraints, and consequently, construct the LNN after numerous successive docking experiences at that desired pose. A necessary condition for good generalization of the LNN is that the training data obtained from the prior docking experiences has to be a large and representative subset of the population in the input space such that the LNN would not only learn from frequently appearing data, but also place less emphasis on the rarely seen training data within the database. This can be a concern in our LNN data-collection method noted above: for example, when in the first trial, if the vehicle docks at (the desired) Pose a, arriving from Pose b, via the implementation of the GNN, but then, in the next ten trials, it arrives at Pose a from Pose c (instead of Pose b). Inevitably, the trained LNN would now place less emphasis on the learning sample from the first docking experience and over-train based on the data acquired for the next ten experiences. Namely, it would "waste" potentially valuable knowledge gained from the first experience.

As a practical consideration, the number of data points within the database cannot be unrestricted. Thus, a method for simplification of the database samples is necessary to limit the memory requirement on the database and the necessary training time of the LNN model, while maintaining a representative dataset for good generalization. Hence, a database-control technique based on the geometric distance of inputs of the LNN (i.e., PSDs offsets) is used herein by removing redundant learning samples. As a new learning data point is added into the database, the new PSDs offsets, $e_{ij(\text{new})}$, are compared to those that are already there in the database, $e_{ij(\text{old})}$: the older tuples are removed from storage if the geometric distance between the two is smaller than a predefined threshold, d, Eq. (2). It implies that the old points are removed when the difference of the new and old tuples are insignificant, allowing the database to save only sufficiently different data while maximizing the input spaces of the NN mapping at the same time:

Add $e_{ij(\text{new})}$ into database

if

$$\begin{aligned} &\left[\left(e_{1x(\text{new})} - e_{1x(\text{old})}\right)^2 + \left(e_{1y(\text{new})} - e_{1y(\text{old})}\right)^2 \right. \\ &\quad + \left(e_{2x(\text{new})} - e_{2x(\text{old})}\right)^2 + \left(e_{2y(\text{new})} - e_{2y(\text{old})}\right)^2 \\ &\quad \left. + \left(e_{3x(\text{new})} - e_{3x(\text{old})}\right)^2 + \left(e_{3y(\text{new})} - e_{3y(\text{old})}\right)^2\right]^{1/2} < d \end{aligned} \tag{2}$$

remove $e_{ij(\text{old})}$

end.

One must note, however, that the number of learning samples from previous dockings at a desired pose may not be sufficient for the training of the corresponding LNN. In such cases, it may be advantageous to combine information generated via the GNN with the online acquired samples for the training of the LNN. However, the differences in the architecture of the GNN and LNN models prevent the inheritance of GNN's structure parameters (i.e., interconnection weights) to the LNN. As a result, the inclusion of knowledge from the pretrained GNN can only be achieved through the generation of *pseudodata* by means of feeding the GNN with sets of pertinent input vectors to obtain the corresponding sets of output vectors. An input vector includes the LOS angles, ϕ_{mn}, corresponding to the particular desired pose for the LNN, and a random set of PSD offsets, e_{ij}; It yields an output vector, (Δx, Δy, Δz, $\Delta\gamma$, $\Delta\beta$, $\Delta\alpha$), via the GNN, as shown in Fig. 6. In order to avoid the generation of unreliable pseudodata, the PSD offsets are randomly chosen *within* the bounds of the PSD offsets data that were obtained for the offline learning of GNN model $e_{ij(\text{global})}$. The pseudodata are then combined with the acquired *real* learning tuples from previous docking experiences within the database to construct the specific LNN model.

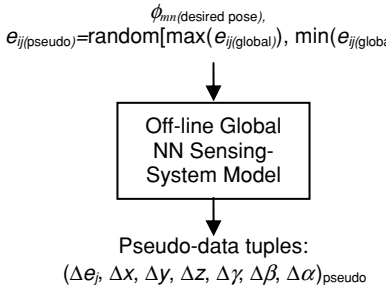


Fig. 6. Pseudodata generation.

https://doi.org/10.1017/S0263574707003359 Published online by Cambridge University Press

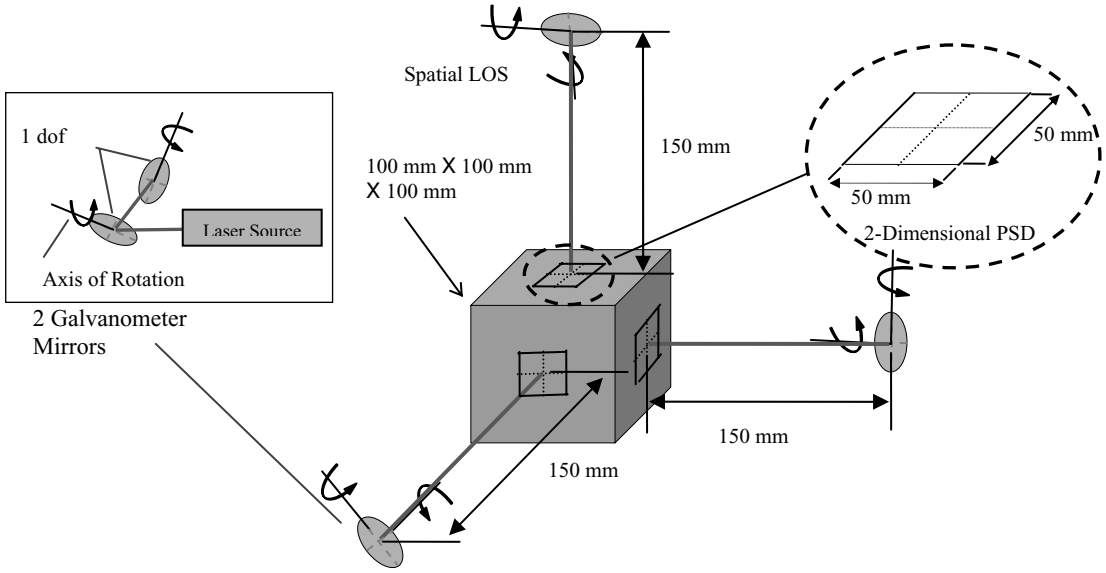


Fig. 7. Simulation environment.

3.3. Simulations

In order to illustrate the workings of the NN-based approach in minimizing the systematic errors and converging within the random noise of the autonomous system, a computer simulation environment was developed and numerous simulations were implemented for a vehicle with 6-dof mobility.

3.3.1. Setup and procedure.

3.3.1.1. The setup. The vehicle's shape is a $0.1 \times 0.1 \times 0.1\,\mathrm{m}^3$. The three spatial LOS sources are placed symmetrically around the perimeter of the (docking) workspace of the vehicle. Each source uses two 1-dof galvanometer mirrors to provide a spatial LOS. The three array-type detectors, $50 \times 50\,\mathrm{mm}^2$, are placed centrally on the three faces of the vehicle, respectively, as shown in Fig. 7.

3.3.1.2. The procedure. The aim of the simulation is to demonstrate that the proposed NN-based methodology has the ability to lessen the impact of the vehicle's systematic errors, by accurately calculating corrective vehicle motion commands, and allow the vehicle to converge to its desired pose within the predefined tolerances. A secondary goal is also to illustrate that the convergence rate is tangibly enhanced in cases of repetitive docking to a particular pose through the use of a LNN model and that this convergence is not a result of a statistical coincidence and is maintainable. Thus, during a simulation, the vehicle is commanded to (long-range) move to a desired pose from an initial random pose. Subsequently, corrective motions are applied iteratively via the proposed NN-based methodology.

3.3.1.3. Motion errors. The inaccuracy of the motion of the vehicle is represented twofold: systematic errors and random errors. Instead of modeling each source of systematic errors separately, their combined effect is represented herein as a function of the overall motion of the vehicle:

$$\textit{Systematic error} = (\textit{Inaccuracy/full_range}) \times \textit{Displacement_of_vehicle}, \quad (3)$$

where in our simulations, (*inaccuracy/full_range*) was chosen as 5.8 μm/mm for translation and 5.6 millidegree/degree for rotation. Unpredictable motion errors, as a result of the vehicle's interaction with its environment, were represented by a normally distributed random noise: $N(\mu = 0.0,\ \sigma = 0.5\ \mu\mathrm{m})$ for translation and $N(\mu = 0.0,\ \sigma = 0.262$ millidegree) for rotation.

3.3.1.4. Sensing-system errors. The galvanometer mirrors are assumed to have *perfect* accuracy. This assumption places the emphasis on testing the methodology itself rather than the physical components of the system. The PSDs readings were modeled to have random noise of $N(\mu = 0.0,\ \sigma = 1.0\ \mu\mathrm{m})$ due to imperfect Analog-to-Digital (A/D) signal conversions. The overall *noise level* of the autonomous system for convergence purposes was thus assumed to be approximately $\pm 12\ \mu$m in terms of PSD-offset readings.

3.3.1.5. The GNN. The GNN was trained offline with 150 tangent sigmoidal neurons in its hidden layer. Due to the excessive computation time and memory requirement of the GNBR algorithm on a complex problem presented herein, the scaled conjugate gradient algorithm with early stopping was used.[38] For the construction of the sensing-system model, 400 discrete data points were taken (300 data for the training subset, and 50 data for each of the evaluation and testing subsets, respectively). By incorporating 300 data from the constraining set, a total of 600 data points were used to train the network. The implemented network ceased training at 3,602 epochs. The training took about 648 s in MatLAB environment on a Pentium IV 2.0-GHz 512-Mb RAM system.

In the online incremental learning phase, each learning sample (i.e., combination of LOS angles, corrective motion

https://doi.org/10.1017/S0263574707003359 Published online by Cambridge University Press

commands, and the resultant differences in PSDs readings) was passed once at a learning rate of 0.01 and each pass required approximately 0.17 s to complete. In order to simulate the applicability of our method in high-speed localization applications, the implementation of the online temporary NN was delayed by three steps to allow sufficient learning time for the online temporary network and meanwhile without holding up the actual docking process (i.e., a temporary NN which has identical weights and architecture as the offline trained NN was used to predict the corrective motion command for the first three steps of docking, where the corrective commands in the fourth step of the iterative procedure were generated by the temporary NN that was updated with the learning sample obtained in first step, etc.).

3.3.1.6. The LNNs. Three LNNs were constructed for a repetitive docking task at the desired pose: $x = 100.0$ mm, $y = 90.0$ mm, $z = -100.0$ mm, $\gamma = -15.0^\circ$, $\beta = -15.0^\circ$, $\alpha = 15.0^\circ$. The first LNN (LNN1) was built using information obtained during the first (successful) docking attempt and six pseudodata points. The second and third models, LNN2 and LNN7, were built only with *real* data obtained from the *first two* and *first seven* (successful) GNN-only *docking attempts*, respectively. The geometric threshold of the database control, d, was set to 0.1 to reduce the number of learning samples for the LNN2 and LNN7 from 21 to 18 and 78 to 38, respectively. All three LNNs were trained with 10 hidden neurons, and took approximately 6, 16, and 11 s to train with GNBR algorithm for LNN1, LNN2, and LNN7, respectively.

3.3.2. Simulation results.

3.3.2.1. GNN-only. The results of 100 random simulation runs are shown in Fig. 8. The 100 desired poses were uniformly distributed within the workspace of the vehicle and were approached from different random poses. For comparison, the results for solely utilizing an offline trained GNN model (without online improvement) are also shown. In all 100 simulation runs, the vehicle converged to the desired pose within the random noise level within 11 steps when online sequential learning was employed. This was not the case for the GNN without online learning.

3.3.2.2. LNNs. Fig. 9 presented the working of the LNN model for simulated dockings to a particular desired pose: $x = 100.0$ mm, $y = 90.0$ mm, $z = -100.0$ mm, $\gamma = -15.0^\circ$, $\beta = -15.0^\circ$, $\alpha = 15.0^\circ$, by using (*i*) solely the GNN model with online incremental learning, (*ii*) a LNN model trained with only one previous docking experience and pseudodata, LNN1, (*iii*) a LNN model trained with two previous docking experiences, LNN2, and (*iv*) a LNN model trained with seven previous docking experiences, LNN7. The desired pose was approached from 100 random distinct poses within the workspace. (One must note that the 100 docking tests were carried out only after the LNNs were already trained using different *previous* docking experiences). It is clearly notable that the effectiveness of the system tangibly increases via the implementation of a LNN for repetitive docking tasks.

A specific convergence simulation test point is also detailed herein for the above desired pose. Due to systematic and random errors, the vehicle's actual pose after the initial long-range motion was noted to be $x = 98.5$ mm, $y = 88.5$ mm, $z = -101.5$ mm, $\gamma = -15.17^\circ$, $\beta = -15.17^\circ$, and $\alpha = 15.17^\circ$. The corresponding PSD_i ($i = 1$ to 3) offsets were "measured" as 2.518, 2.776, and 2.708 mm, respectively. The proposed docking methodology was invoked iteratively to guide the vehicle to its desired pose using LNN2, which was *achieved* in five steps. The PSD offsets for each iteration, given as the absolute distance from the PSD's center, are shown in Fig. 10(a), where Fig. 10(b) is the close-up view of each PSD. In order to illustrate that

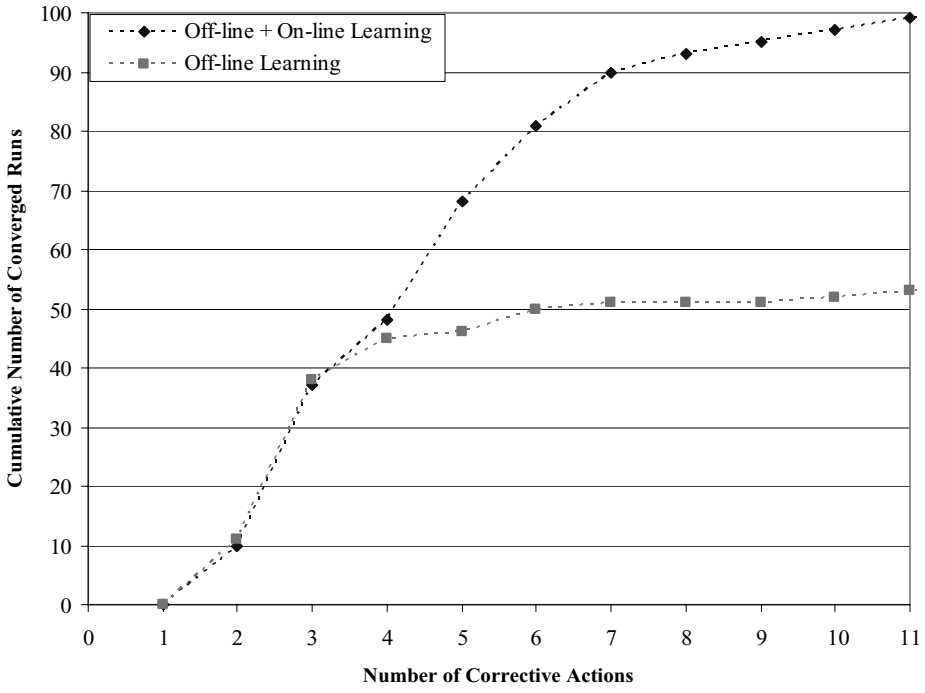


Fig. 8. Results for 100 random simulation runs.

https://doi.org/10.1017/S0263574707003359 Published online by Cambridge University Press

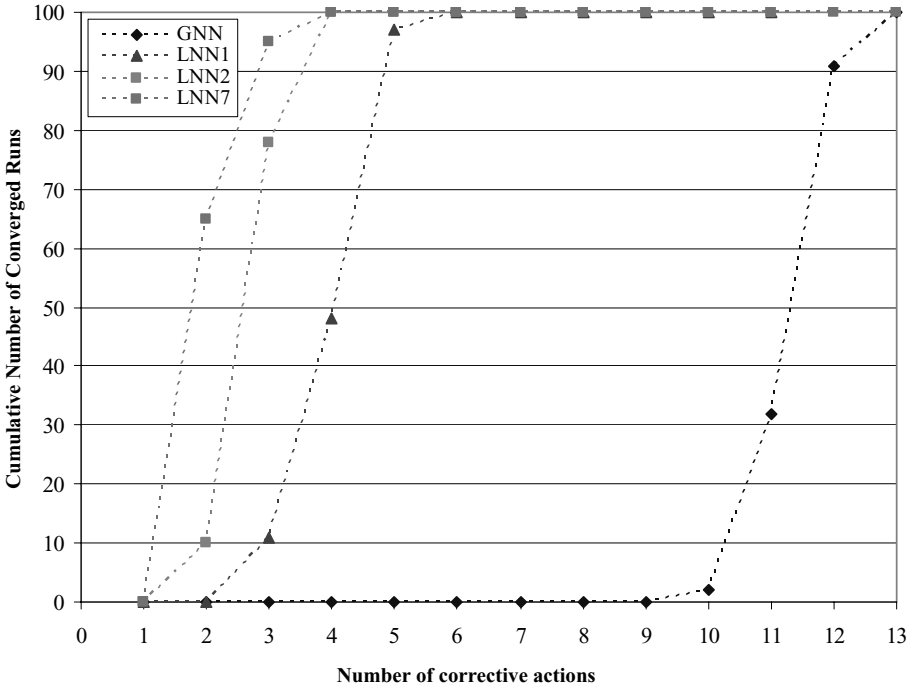


Fig. 9. Results for 100 simulation runs at a particular desired pose.

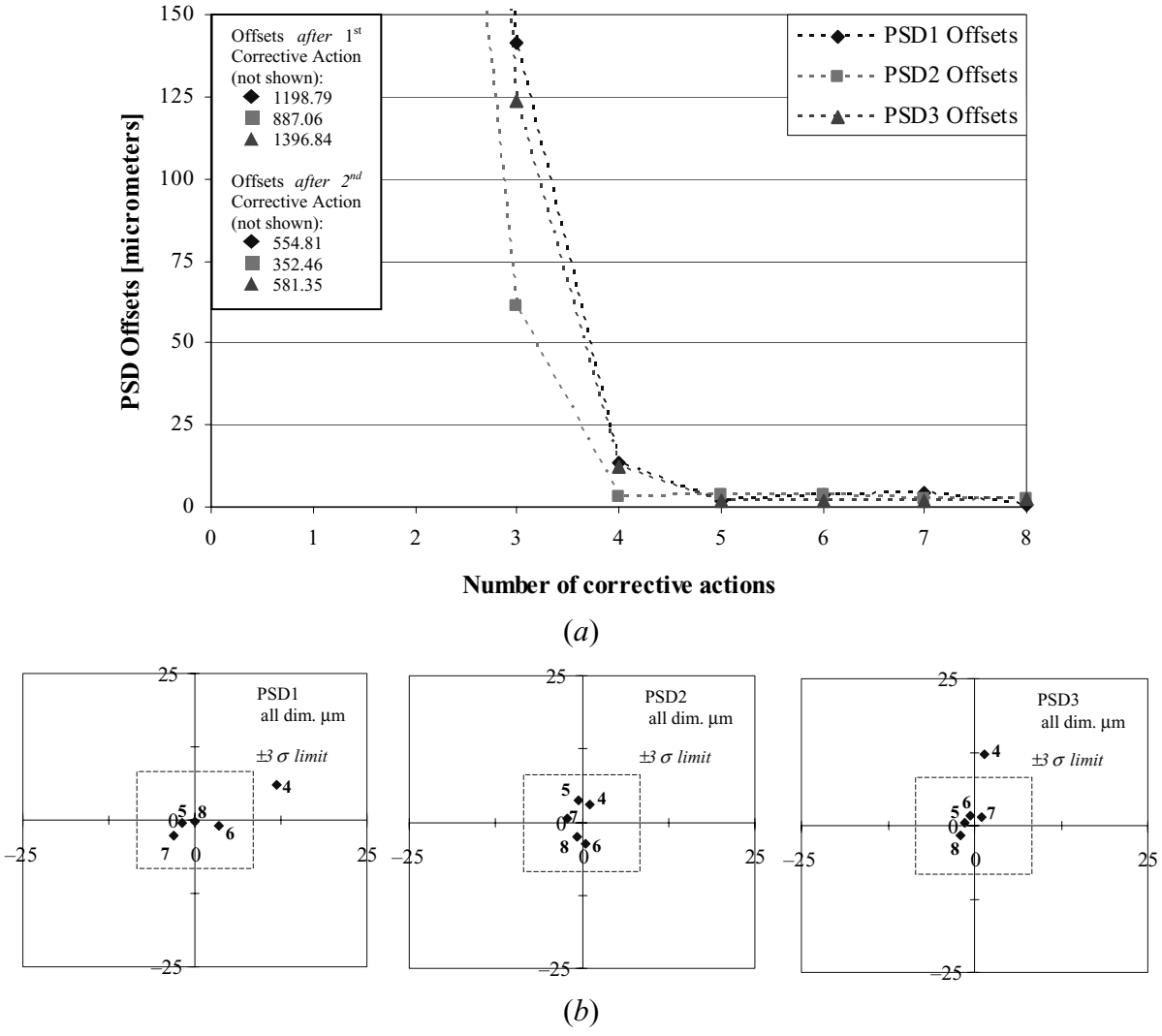


Fig. 10. Simulation results of PSD offsets: (a) Absolute distances and (b) *x* and *y* directions in close-up views of the PSDs.

https://doi.org/10.1017/S0263574707003359 Published online by Cambridge University Press

convergence is maintainable, the PSDs readings for several additional corrective actions are also shown (corrective actions #6 – #8).

3.3.3. Discussion. In our proposed methodology, the GNN is used for the docking of the autonomous vehicle at any possible desired pose within its workspace. If docking at a particular pose is known to be repeated, the GNN would be used to dock the vehicle at this pose prior to the construction and availability of a LNN model, though the convergence rate would be relatively slow in comparison to what a LNN could possibly yield. A LNN is trained offline for a particular pose using *pseudodata* generated from GNN and/or *real* data from previous successful dockings, and subsequently used in lieu of the GNN for faster docking.

The simulation results have illustrated that our NN-based docking methodology can successfully guide an autonomous vehicle to its desired pose within the bounds of the random noise level of the system and that the convergence rate can be tangibly improved via the use of LNN models. It is also important to note that whether the noise-to-signal ratio is large or not, the generic 6-dof vehicle has the ability to converge to its desired pose within its random noise limits regardless of the application at hand. In particular, the results presented here, show the effectiveness of the method to allow the vehicle to converge to its desired pose with *initial errors* 200 times greater than the random noise level.

Although one could argue that the convergence rate of the GNN can be enhanced by increasing the number of offline acquired learning samples and the complexity of the model, this may be intractable in practice for high-precision docking applications due to computational burden and/or economics. Thus, in response, the proposed two-level NN sensing-system strategy can be employed: use of a low-precision GNN model for the first docking task and use of a LNN model for subsequent repeated docking tasks at the same desired pose. The primary reason for having a GNN model in the first place is that, in some autonomous-vehicle applications, there may be a large number of different docking poses that may, furthermore, be unknown prior to the operation of the vehicle. These docking poses may also change throughout the operational lifetime of the vehicle.

3.4. Summary

The generic NN-based guidance approach proposed in this paper utilizes two models for the sensing system: (*a*) global and (*b*) local. These sensing-system models employ indirect-proximity information from the LOS sensing system to predict the vehicle's motion commands for the purpose of minimizing the offsets registered on the PSDs. Although the NN-based approach does necessitate some offline training, it does not require the existence and use of an analytical calibration model, and yet provides a fast rate of convergence even in the presence of large deviations between the vehicle's actual and desired poses. Through this two-level sensing-system framework and the implementation of online learning, the offline measurement time for a high-precision vehicle docking task has been greatly minimized.

The two-level NN sensing-system framework starts with the offline acquisition of discrete measurement points via a teaching-by-demonstration method for the training of the GNN model. A data point for the training of the GNN is obtained by the following procedure: (i) the vehicle is commanded to move to a possible desired docking pose and the corresponding LOSs angles are recorded, and (ii) after a small incremental motion of the vehicle, while keeping the LOSs locked, the registered offsets on each of the PSDs are measured. The offline trained GNN provides a start-up model for the docking of the vehicle at any pose within its workspace.

During the short-range motion-planning stage of the vehicle (i.e., docking task), a temporary GNN model, which initiates with identical network's parameters as the offline-trained GNN, is utilized to predict the necessary corrective motions after long-range positioning of the vehicle, based on estimation of the current pose of the vehicle by employing the PSD offsets. The temporary GNN is progressively refined via online incremental learning.

The LNN model is utilized to emphasize the docking at a particular pose. Due to its simplicity and robustness on the necessary NN modeling, it requires a less-complex training process while providing a faster convergence rate than the GNN. The LNN utilizes learning samples from previous successful docking attempts for offline learning and/or pseudodata. It is designed for the repeated docking at a particular desired pose and, when available, it can be used instead of the GNN.

4. Experiments

In order to illustrate the implementation of the proposed docking methodology, a number of experiments were conducted. These verified that the use of a LOS-based (indirect proximity-estimation) sensing system in conjunction with the NN-based docking strategy proposed in this paper can effectively minimize systematic motion errors of autonomous vehicles and allow them converge to a desired pose within the random noise limits of the system.

4.1. Setup and procedure

The proposed guidance method was implemented and tested in a controlled physical environment for the online docking of a planar high-precision 3-dof (x, y, ϕ) platform (i.e., a *simulated* autonomous vehicle), as shown in Fig. 11. One can note that when using such a precise platform, we can simulate any scenario we wish with errors that would be experienced by an autonomous vehicle (i.e., automobiles, mobile robots) in that specific application. The experimental setup included three single-LOS sensing modules placed symmetrically on the perimeter of the platform's workspace, a combination of beam splitters, flat mirrors, and a filter to split the beam from the laser source into three beams, and then, redirect them to their corresponding galvanometer mirrors. The systematic errors were manually set to values of 5.8 μm/mm for the linear stages and 6.7 millidegree/degree for the rotational stage. The objective of the experiments is to validate the proposed methodology's ability to reduce the impact of these systematic motion errors of the platform resulting from inaccurate long-range positioning by generating the

https://doi.org/10.1017/S0263574707003359 Published online by Cambridge University Press

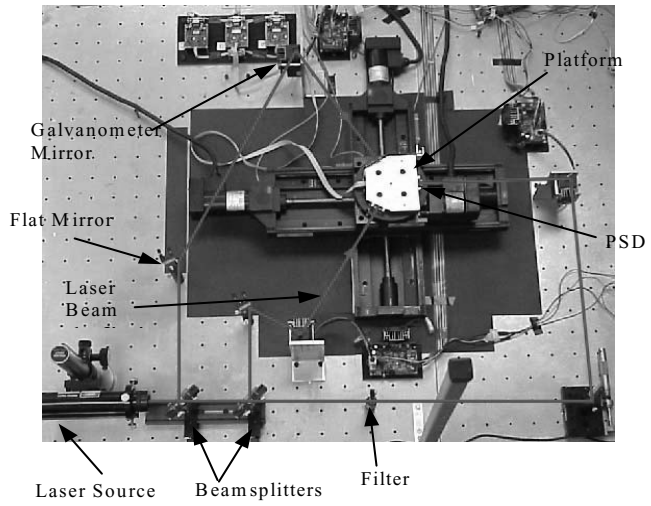


Fig. 11. Experimental setup.

necessary corrective motion commands using real-time LOS sensory information.

4.1.1. Data acquisition and training. As noted in Section 3, the generic NN-based guidance approach proposed in this paper utilizes two models for the sensing system: (*a*) a GNN and (*b*) a LNN: only the GNN requires offline training, which does not need to be extensive due to the online development of LNNs and their subsequent uses at repeated docking locations. The offline acquisition of discrete measurement points for the training of the GNN model is carried out via a teaching-by-demonstration method and the effort required for this task is directly proportional to the number of data points. A data point for the training of the GNN is obtained by the following procedure: (i) the vehicle is commanded to move to a possible desired docking pose and the corresponding LOSs angles are recorded and (ii) after a small incremental motion of the vehicle, while keeping the LOSs locked, the registered offsets on each of the PSDs are measured.

The offline acquisition of a data point for the GNN requires the (iterative) realignment of the three galvanometer scanners until the PSD offsets are zeroed (within their random noise level), which is the primary time-consuming task. Although a manual process could take a few minutes for each data point, the data collection period can be substantially shorter through the use of an automated data-acquisition system. Regardless of whether one uses a manual or an automated system, it is noted that (a) offline data collection is only performed once for the construction of the GNN and (b) the LNN does not require any offline acquisition of data.

In our work, the GNN was trained with the GNBR algorithm and 30 hidden neurons, with 25 and 10 measurement points for the training and testing subsets, respectively. With the inclusion of 25 data points from the constraining set, the network ceased training in 124 s at 737 epochs. For the online sequential learning, a delay factor of 2 was used. Each learning sample is passed once with the learning rate set to 0.0025 and takes around 0.10 s to complete. The LNN incorporated information from one previous docking experience (6 learning points) with four additional pseudodata and completed training in 8.7 s. The LNN2 utilized 12 learning points obtained from two previous experiences and ceased training in 14.5 s.

4.2. Experimental results

The guidance methodology was tested for several distinct poses within the platform's workspace. Results for one particular desired pose are presented here where: $x = -57.427$ mm, $y = -45.711$ mm, $\phi = -19.234°$, when using only the GNN, the LNN1, and the LNN2, Fig. 12 (a), (b), and (c), respectively. After long-range motion, the actual

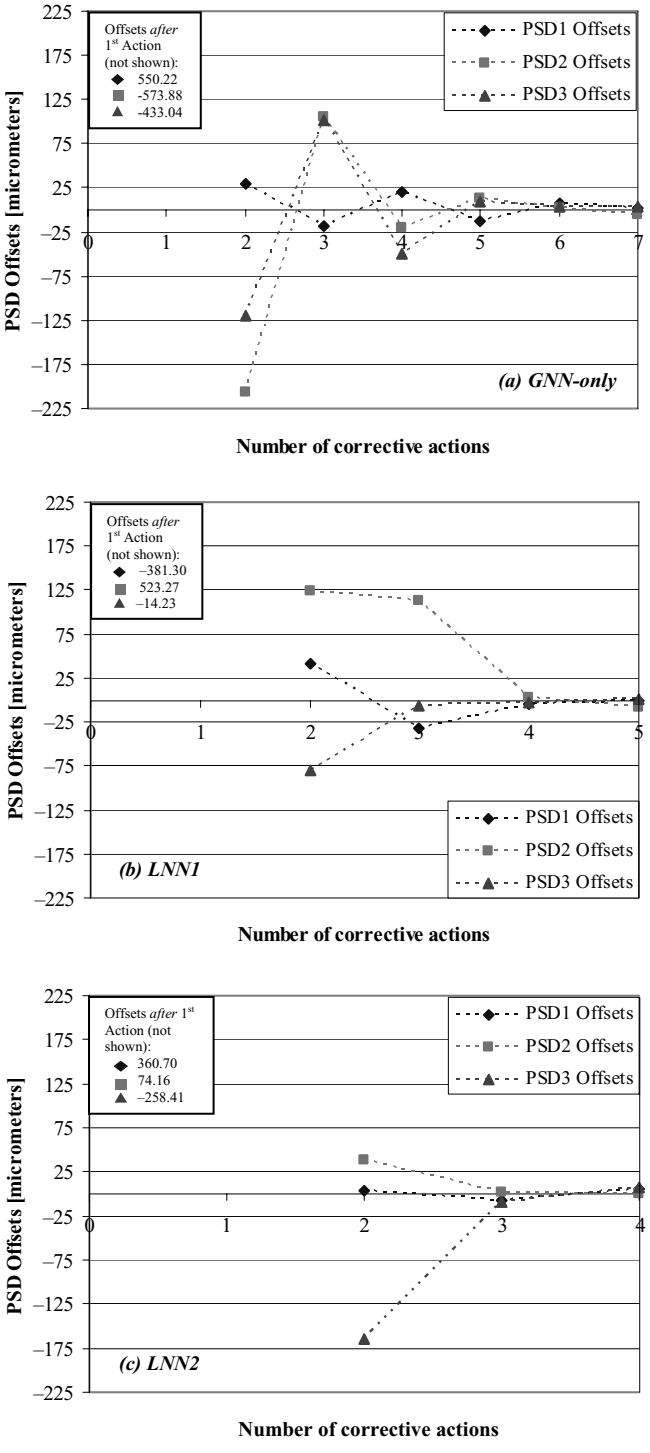


Fig. 12. Convergence graphs for experimental results.

https://doi.org/10.1017/S0263574707003359 Published online by Cambridge University Press

(uncorrected) pose, $x = -55.927$ mm, $y = -46.711$ mm, and $\phi = -19.034^\circ$, was deviated from the desired pose due to systematic and random motion errors resulted from the long-range positioning of the platform.

The initial PSD_i ($i = 1$ to 3) offsets were measured as 1.93, 1.22, and 0.47 mm, respectively. The proposed guidance methodology needed six, four, and three steps to guide the platform to the desired pose within the random noise limits of the PSDs (± 10 μm) using the GNN, LNN1, and LNN2 models, respectively. In order to ensure convergence, an extra correction action was performed for all three tests.

5. Conclusions

A generic NN-based task-space multi-LOS guidance method has been proposed in this paper for use in the high-precision docking of autonomous vehicles. The novel online motion-planning methodology successfully overcomes the primary inability of the most existing systems in obtaining accurate orientation measurements. LOS-based sensory data and PSD offsets are utilized by the (online) updated GNN and LNN to generate corrective motion commands for online docking during the final short-range positioning stage, namely, minimizing the systematic errors of the vehicle after long-range motion and allow it to converge within the random noise level of the overall system.

The proposed two-level sensing-system strategy minimizes the size of the required NN training set for high-precision docking tasks through the use of online attainable data, thus, significantly enhancing system performance. The guidance methodology does not require a sensing-system calibration model, and yet provides fast and stable convergence.

The effectiveness and robustness characteristics of our proposed method were verified successfully in both simulations and experiments. In particular, we can highlight here that whether the noise-to-signal ratio is large or not, or whether the vehicle has 6-dof mobility or only 3-dof, the algorithm guides the autonomous vehicle to its desired pose within its random noise limits, regardless of the application at hand.

Acknowledgments

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

References

1. G. Nejat and B. Benhabib, "A Guidance-Based Motion-Planning Methodology for the Docking of Autonomous Vehicles," *J. Robot. Syst.* **22**(12), 779–793 (2005).
2. P. M. Lee, B. H. Jeon and C. M. Lee, "A Docking and Control System for An Autonomous Underwater Vehicle," *Proceedings of the MTS/IEEE, Oceans Conference*, Biloxi, MI, (2002) pp. 1609–1614.
3. T. Gomi and K. Ide, "Vision-Based Navigation for an Office Messenger Robot," *Proceedings of the IEEE, Conference on Intelligent Robots and Systems*, Munich, Germany (1994) pp. 2015–2022.
4. K. O. Arras and S. J. Vestli, "Hybrid, High-Precision Localization for the Mail Distributing Mobile Robot System MoPS," *Proceedings of the IEEE, Conference on Robotics and Automation*, Leuven, Belgium (1998) pp. 3129–3134.
5. H. Roth and K. Schilling, "Navigation and Docking Manoeuvres of Mobile Robots in Industrial Environments," *Proceedings of the IEEE, Conference of Industrial Electronics Society*, Aachen, Germany (1998) pp. 2458–2462.
6. C. Vestri, S. Bougnoux, R. Bendahan, K. Fintzel, S. Wybo, F. Abad and T. Kakinami, "Evaluation of a Vision-Based Parking Assistance System," *Proceedings of the IEEE, Conference on Intelligent Transportation System*, Vienna, Austria (2005) pp. 56–60.
7. M. Wada, K. Yoon, H. Hashimot, S. Matsuda and H. Masuda, "Development of Advanced Parking Assistance System," *Proceedings of the IEEE, Conference on Intelligent Transportation Systems*, Tokyo, Japan (1999) pp. 716–721.
8. Transportation Research Board, "Transit IDEA Strategic Initiative on Bus Rapid Transit (BRT)," Online. http://gulliver.trb.org/publications/sp/BRTInitiative.pdf (March 7, 2006).
9. S. E. Shladover and M. A. Miller, "Evaluation of Lane-Assist Systems for Urban Transit Operations," *IEEE, Conference on Intelligent Transportation Systems*, Washington, DC (2004) pp. 749–754.
10. SmartDock Docking Aid system, [cited (2006 February 16)], Available HTTP: http://www.harbourmarine.com/products/smartdock-DAS.html.
11. D. Inaudi, G. Brunetti, A. Del Grosso and M. Fedolino, "Monitoring of Harbor Piers with Fiber Optic Displacement Sensors", *Proc. SPIE* **3995**, 175–179 (2000).
12. G. Ortega and J. M. Giron-Sierra, "Genetic Algorithms for Fuzzy Control of Automatic Docking With a Space Sation," *Proceedings of the IEEE, Conference on Evolutionary Computation*, Perth, WA (1995) pp. 157–161.
13. R. T. Howard, T. C. Bryan and M. L. Book, "An Advanced Sensor for Automated Docking," *Proceedings of the Conference on Digital Avionics Systems*, Daytona Beach, FL (2001) pp. 8B6 /1–8B6/7.
14. C. J. J. Paredis and P. K. Khosia, "Agent-Based Design of Fault Tolerant Manipulators for Satellite Docking," *Proceedings of the IEEE, International Conference on Robotics and Automation*, Albuquerque, NM (1997) pp. 3473–3480.
15. W. M. Shen, P. Will and B. Khoshnevis, "Self-Assembly in Space via Self-Reconfigurable Robots," *Proceedings of the IEEE, Conference on Robotics and Automation*, Taipei, Taiwan (2003) pp. 2516–2521.
16. M. Ruberstein, K. Payne, P. Will and W. M. Shen, "Docking Among Independent and Autonomous CONRO Self-Reconfigurable Robots," *Proceedings of the IEEE, Conference on Robotics and Automation*, Barcelona, Spain (2004) pp. 2877–2882.
17. A. Nilsson and P. Holmberg, "Combining a Stable 2-D Vision Camera and an Ultrasonic Range Detector for 3-D Position Estimation," *Proceedings of the IEEE, Transaction on Instrumentation and Measurement* **43**(2), 272–276 (2004).
18. D. J. Balek and R. B. Kelley, "Using Gripper Mounted Infrared Proximity Sensors for Robot Feedback Control," *Proceedings of the IEEE, Conference on Robotics and Automation* **2**, 282–287 (1985).
19. M. Wada, K. S. Yoon and H. Hashimoto, "Nonlinear Filter Road Vehicle Model Development," *Proceedings of the IEEE, International Conference on Intelligent Transportation Systems*, Oakland, CA (2001) pp. 734–739.
20. N. Doh, H. Choset, and W. K. Chung, "Accurate Relative Localization Using Odometry," *Proceedings of the IEEE, International Conference on Robotics and Automation*, Taipei, Taiwan (2003) pp. 1606–1612.
21. F. Chenavier and J. Crowley, "Position Estimation for a Mobile Robot Using Vision and Odometry," *Proceedings of the IEEE, International Conference on Robotics and Automation*, Nice, France (1992) pp. 2588–2593.
22. J. Borenstein and L. Feng, "Correction of Systematic Odometry Errors in Mobile Robots," *Proceedings of the IEEE,*

https://doi.org/10.1017/S0263574707003359 Published online by Cambridge University Press

International Conference on Intelligent Robots and Systems, Pittsburgh, PA (1995) pp. 569–574.
23. Y. Akcayir and Y. Ozkazanc, "Gyroscope Drift Estimation Analysis in Land Navigation Systems," *Proceedings of the IEEE, Conference on Control Applications*, Istanbul, Turkey (2003) pp. 1488–1491.
24. M. D. Cecco, "Self-Calibration of AGV Inertial-Odometric Navigation Using Absolute-Reference Measurements," *Proceedings of the Instrumentation and Measurement Technology Conference*, Padova, Italy (2002) pp. 1513–1518.
25. P. Roessler, S. A. Stoeter, P. E. Rybski, M. Gini and N. Pananikolopoulos, "Visual Servoing of A Miniature Robot Toward A Marked Target," *Proceedings of the IEEE, Conference on Digital Signal Processing*, Santorini, Greece (2002) pp. 1015–1018.
26. C. J. Taylor and J. P. Ostrowski, "Robust Visual Servoing Based on Relative Orientation," *Proceedings of the IEEE, Computer Vision and Pattern Recognition Conference*, Fort Collins, CO (1999) pp. 574–580.
27. J. Borenstein, H. Everett, L. Feng and D. Wehe, "Mobile Robot Positioning: Sensors and Techniques," *J. Robot. Syst.* **14**(4), 231–249 (1997).
28. R. Willgoss, V. Rosenfeld and J. Billingsley, "High Precision GPS Guidance of Mobile Robots," *Proceedings of the Australian Conference on Robotics and Automation*, Brisbane, Australia (2003) pp. 1–6.
29. N. H. Kussat, C. D. Chadwell and Richard Zimmerman, "Absolute Positioning of an Autonomous Underwater Vehicle Using GPS and Acoustic Measurements," *IEEE J. Ocean. Eng.* **30**, 153–164 (2005).
30. A. Alcocer, P. Oliveira and A. Pascoal, "Underwater Acoustic Positioning Systems Based on Buoys with GPS," *Proceedings of the IEEE, European Conference on Underwater Acoustics*, Carvoeiro, Portugal (2006) pp. 1–8.
31. R. E. Saad, A. Bonen, K. C. Smith and B. Benhabib, "Proximity Sensing for Robotics," **In**: *The Measurement, Instrumentation, and Sensors Handbook* (J.G. Webster, ed.), (CRC Press, 1999) pp. 1–16.
32. E. I. Konukseven and B. Kaftanoglu, "Robot End-effector Based Sensor Integration for Tracking Moving Parts," *Proceedings of the International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies*, Brighton, U.K. (2000) pp. 628–634.
33. P. P. L Regtien, "Accurate Optical Proximity Detector," *Proceedings of the IEEE, Instrumentation and Measurement Technology Conference*, San Jose, CA (1990) pp. 141–143.
34. A. Bonen, R. E. Saad, K. C. Smith and B. Benhabib, "A Novel Electrooptical Proximity Sensor for Robotics: Calibration and Active Sensing," *Proceedings of the IEEE Transactions on Robotics and Automation* **13**, 377–386 (1997).
35. G. Nejat and B. Benhabib, "High-Precision Task-Space Sensing and Guidance for Autonomous Robot Localization," *Proceedings of the IEEE, International Conference on Robotics and Automation*, Taipei, Taiwan (2003) pp. 1527–1532.
36. G. Nejat, A. Membre and B. Benhabib "Active Task-Space Sensing and Localization of Autonomous Vehicle," *Proceedings of the IEEE, International Conference on Robotics and Automation*, Barcelona, Spain (2005) pp. 3781–3786.
37. G. Nejat and B. Benhabib, "Docking of Autonomous Vehicles: A Comparison of Model-Independent Guidance Methods," *Proceedings of the Virtual International Conference on Intelligent Productions and Machines* (2005) pp. 1–6.
38. I. K. Jung, K. B. Hong, S. K. Hong, and S. C. Hong, "Path Planning of Mobile Robot Using Neural Network," *Proceedings of the IEEE, International Conference on Fuzzy Systems*, New Orleans, LA (1996) pp. 2208–2214.
39. I. K. Sethi and G. Yu, "A Neural Network Approach to Robot Localization Using Ultrasonic Sensors," *Proceedings of the IEEE, International Symposium on Intelligent Control*, Philadelphia, PA (1990) pp. 513–517.
40. W. L. Xu, K. H. Wurst, T. Wantanabe and S. Q. Yang, "Calibrating a Modular Robotic Joint Using Neural Network Approach," *Proceedings of the IEEE, World Congress on Computational Intelligence*, Orlando, FL (1994) pp. 2720–2725.
41. X. L. Zhong and J. M. Lewis, "A New Method for Autonomous Robot Calibration," *Proceedings of the IEEE, International Conference on Robotics and Automation*, Nagoya, Japan (1995) pp. 1790–1795.
42. C. S. Langley and G. M. T. D'Eleuterio, "Neural Network-Based Pose Estimation for Fixtureless Assembly," *Proceedings of the IEEE, International Symposium on Computational Intelligence in Robotics and Automation*, Banff, Canada (2001) pp. 248–253.
43. F. D. Foresee and M. T. Hagan, "Gauss-Newton Approximation to Bayesian Regularization," *Proceedings of the IEEE, International Joint Conference on Neural Networks*, Houston, TX (1997) pp. 1930–1935.
44. M. F. Moller, "A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning," *Neural Netw.* **6**, 525–533 (1993).
45. D. Saad, *On-line Learning in Neural Networks*, (Cambridge University Press, Cambridge, UK (1998).
46. J. A. Nelder and R. Mead, "A Simplex Method for Function Minimization," *Comput. J.* **7**, pp. 308–313 (1964).

https://doi.org/10.1017/S0263574707003359 Published online by Cambridge University Press