

Research Article

Feasibility Investigation of Obstacle-Avoiding Sensors Unit without Image Processing

Yasuhisa Omura, Hiroshi Masuda, and Yoshio Mimura

Graduate School of Science and Engineering, Kansai University, 3-3-35 Yamate-cho, Suita, Osaka 564-8680, Japan

Correspondence should be addressed to Yasuhisa Omura; omuray@kansai-u.ac.jp

Received 11 February 2013; Accepted 9 June 2013

Academic Editor: Aiguo Song

Copyright © 2013 Yasuhisa Omura et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Feasibility of a simple method to detect step height, slope angle, and trench width using four infrared-light-source PSD range sensors is examined, and the reproducibility and accuracy of characteristic parameter detection are also examined. Detection error of upward slope angle is within 2.5 degrees, while it is shown that the detection error of downward slope angle exceeding 20 degrees is very large. In order to reduce such errors, a method to improve range-voltage performance of a range sensor is proposed, and its availability is demonstrated. We also show that increase in trial frequency is a better way, although so as not to increase the detection delay. Step height is identified with an error of ± 1.5 mm. It is shown that trench width cannot be reliably measured at this time. It is suggested that an additional method is needed if we have to advance the field of obstacle detection.

1. Introduction

In the last decade, autonomous mobile robots have been attracting wide attention, and technical levels have dramatically advanced (see, for instance, [1]). Many robots for entertainment, room cleaning, and other services have already been developed [2]. To be really practical, robots must be able to acquire environmental events and/or spatial information of their environment. Some robots for entertainment have optical sensors, ultrasonic sensors, touch sensors, and other configurations which have been implemented. To create more autonomous robots that suit future applications, the 2D infrared range sensor [3] and CMOS-imager camera [4] are being studied extensively. In these studies, sensor downsizing is an ongoing concern. However, the newly developed sensors are still expensive, and computing overhead is apt to increase. This is a fundamental problem with the present research roadmap.

2D path planning for mobile robots has also been studied extensively [5, 6]; it is considered that combining a path planning method [7, 8] with a potential-field method [9, 10] or a mapping technique is a promising approach. These techniques are also needed for future self-learning robots.

On the other hand, recently, a passive intelligent walker is proposed using a servo breaks [11]; in that trial, some obstacles (such as steep slope and steps) are detected. However, a user must change his/her front direction when the sensor has found an obstacle. In addition, the robot does not guide a better direction for walking to the user. Therefore, at least now, blind persons cannot use the walker.

In this paper, how to detect and classify obstacles in front of a robot without a camera [12–14] is investigated. The purpose of this paper is (1) to realize a sensor block that can detect the differences between step, slope, and trench, (2) to form arithmetic procedures to estimate characteristic values (step height, slope angle, and trench width), and (3) to propose algorithms that yield reliable judgments. Four infrared-light-source (IR) PSD range sensors are used. Experiments on the sensor block challenge its sensor functions with steps, slopes, and/or trenches.

The electrical or mechanical configuration of the testing robot is described in Section 2. Section 3 describes the measurement accuracy of the IR PSD range sensors used. Section 4 proposes algorithms that allow the robot to detect obstacles and estimate characteristic values. Section 5 describes the results of an obstacle-detection test and the

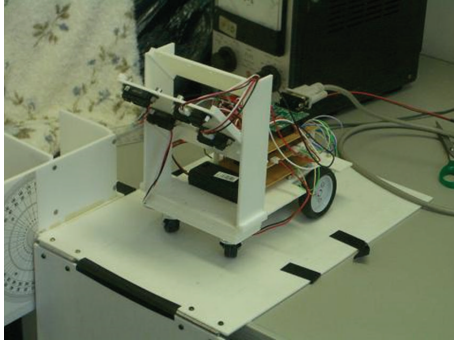


FIGURE 1: Photo of an assembled robot for testing.

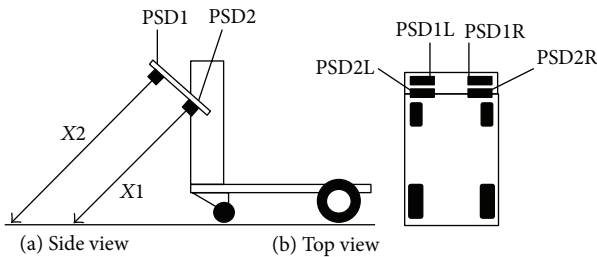


FIGURE 2: Schematic of sensors' layout.

reliability of obstacle recognition. Finally, the remaining issues are summarized.

2. Mechanical and Electrical Architecture of Testing Robot

A picture of the prototype robot to test sensor functions is shown in Figure 1. The testing robot has two nondriven caster wheels at the front and two motor-driven wheels at the back whose rotation speeds are controlled by a motor-drive circuit. The motor-driven wheels have four rotation modes (brake, stop, forward, and back). Since these four functions are implemented on the wheels independently, the robot can move in any direction. Four range sensors are placed on the front of the testing robot (PSD1L, PSD1R, PSD2L, and PSD2R, resp.) to detect obstacles in front of the testing robot (see Figure 2). These four sensors detect distances from the sensor to the floor, and the microcontroller calculates characteristic values, for example, the slope angle θ when the obstacle is a slope.

The electronic architecture of the testing robot is shown in Figure 3. The circuit-mounted board includes a microcontroller (ADuC7026 [15] produced by Analog Devices Corp.) to give the robot a data processing function. The microcontroller has input terminals for up to 12 single-ended A/D converters and other analog processing functions. The microcontroller receives analog signals from sensors through its built-in A/D converters, logically assigns the environment to one of the obstacles or no obstacle, and finally outputs the characteristic value of the obstacle (slope angle θ for the slope, step height h_1 for the step, and so on). The microcontroller

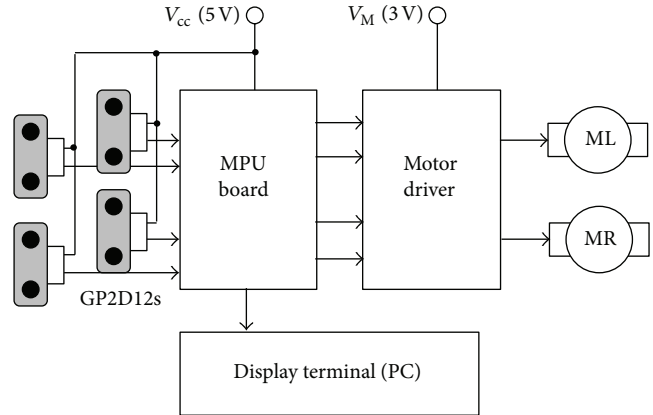


FIGURE 3: Electronic control system for the motor driver and others.

on the MPU board calculates the obstacle's dimensions and transfers the data to a PC via the RC-232 interface.

3. Accuracy and Reproducibility of Output Signal of PSD Sensor

The first step is to evaluate the potential of the IR range sensor (GP2D12 [16] produced by SHARP Corp.) used to detect obstacles; we focus here on the sensor performance attributes not described in the commercial data sheet. This sensor unit has the following features.

- (1) Distance detection range (sensor to object) is 10 to 80 cm in the present case. When GP2Y0A02YK is used, however, the distance detection range is 20 to 150 cm. In this experiment, we employed GP2D12 because of easy verification of proposal.
- (2) IR source signal of one sensor interferes very little with the functioning of the other sensors.
- (3) The sensor is basically insensitive to object color and reflectivity.
- (4) The sensor is basically insensitive to room light.
- (5) Distance from the sensor to the floor can be detected even when the object surface is tilted. However, the variation in range is significant when the tilt angle is large.
- (6) Low cost and small size.

As just described, the IR PSD sensor has many advantages over other sensors. In some cases, however, there is a significant amount of electrical noise in the output signal when we consider some applications that demand the detection of slope angle. This suggests that how accurately the sensor detects distance (X_1 , X_2) before an accurate sensor circuit block is designed has to be examined.

As an example, range data created by transforming the analog signals of the IR PSD sensors are shown in Figures 4 and 5; Figure 4 shows the output of the microcontroller when challenged with an 18 mm high upward step, and Figure 5 shows that for a 20-degree downward slope. In both cases, the

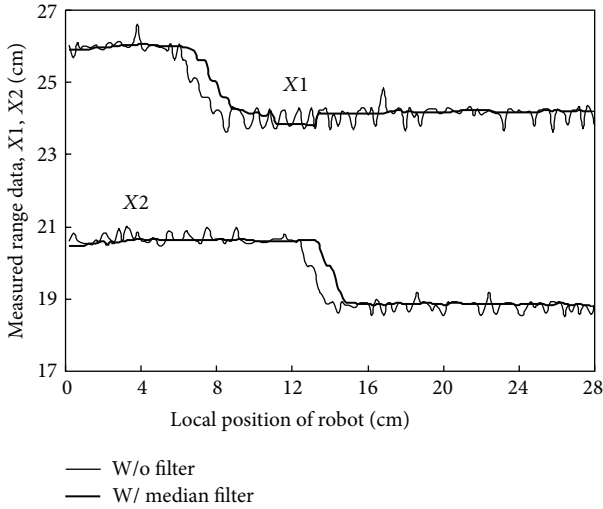


FIGURE 4: Range data evolution when the robot is approaching an upward step.

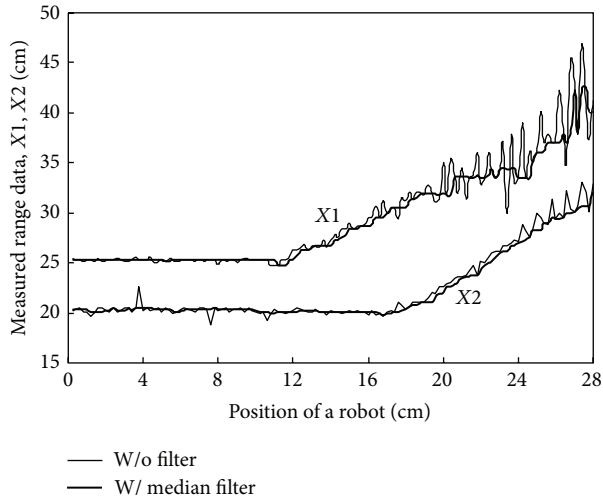


FIGURE 5: Range data evolution when the robot is approaching a downward slope.

testing robot had a constant velocity on the floor. In Figures 4 and 5, the thin lines are the unprocessed digital range data transferred from the microcontroller, while the bold lines are the range data after being passed through a median filter (window number $N1 = 5$) (see the appendix).

Figure 4 shows that the median filter is effective in removing the impulse noise. It also shows that the filter yields a time delay, resulting in a 5 mm local position difference in the case of $N1 = 5$. The noise can be further reduced by increasing $N1$, but at the cost of simultaneously increasing the time delay. Because of this tradeoff, it is preferable to adjust $N1$ to suit the application.

In Figure 5, the impulse noise is sufficiently removed at short distances as well as in Figure 4, but not at distances beyond 50 cm. This is due to the sensor's performance limitation [16]; when $X1 > 50$ cm, even a small voltage shift

of output signal of sensor results in a large variation in range data. When the angle between the IR-light beam from the sensor and the object surface increases, the IR signal returned attenuates, and the influence of room light becomes significant. This means that a downward slope yields a large variation in the detected signal.

4. Method of Extracting Spatial Values

In Section 4.1, how the sensing circuit block identifies steps, slopes, and trenches using the upper and lower sensors (PSD1 and PSD2) is described. Section 4.2 describes the mathematical model that the sensing unit applies to the calculation of step height, slope angle, or trench width. Section 4.3 details the results of experiments on the determination of step height, slope angle, and trench width.

In this chapter, it is assumed that the testing robot directly faces the obstacle (the width of which is taken to be effectively infinite). Note that all the range data ($X1L$, $X1R$, $X2L$, and $X2R$) displayed in the figures are the result of median filtering. Results obtained assuming more practical situations are shown in Section 5.

4.1. How to Classify Slopes, Steps, and Trenches. First, the notations used in this section are explained. $X1$ and $X2$ stand for the distances given by PSD1 and PSD2, respectively. When the testing robot runs on a flat floor, it is assumed that PSD1 and PSD2 yield distance data $X1o$ and $X2o$, respectively. In a practical situation, various noises in the data yielded by the sensors should be taken account of. Accordingly, we introduce positive threshold values of $X1T$ and $X2T$ to improve the detection reproducibility of distance data when determining whether the event (i.e., slope, step, or trench) has occurred. When PSD1 outputs data satisfying the condition of $|X1 - X1o| < X1T$, the testing robot “thinks” that it is on a flat floor. In this case, we say that $S(\text{PSD1}) = \text{“Flat”}$. When PSD1 outputs data satisfying the condition of $|X1 - X1o| > X1T$, the sensing circuit block “thinks” that it may be facing a slope, a step, or a trench. In this case, we say that $S(\text{PSD1}) = \text{“NON-F”}$. In the present experiment, we empirically set $X1T = 0.8$ [cm] and $X2T = 0.5$ [cm] by taking account of the noise level shown in Figures 4 and 5, respectively. For example, the testing robot is running on a flat floor, when the “states” output by the 4 sensors are “flat”, and we use the following descriptions:

$$\begin{aligned}
 S(\text{PSD1L}) &= \text{“Flat”}, \\
 S(\text{PSD1R}) &= \text{“Flat”}, \\
 S(\text{PSD2L}) &= \text{“Flat”}, \\
 S(\text{PSD2R}) &= \text{“Flat”}.
 \end{aligned} \tag{1}$$

Next, how the sensing circuit block uses the trigonometric method shown in Figure 6 in order to extract geometrical

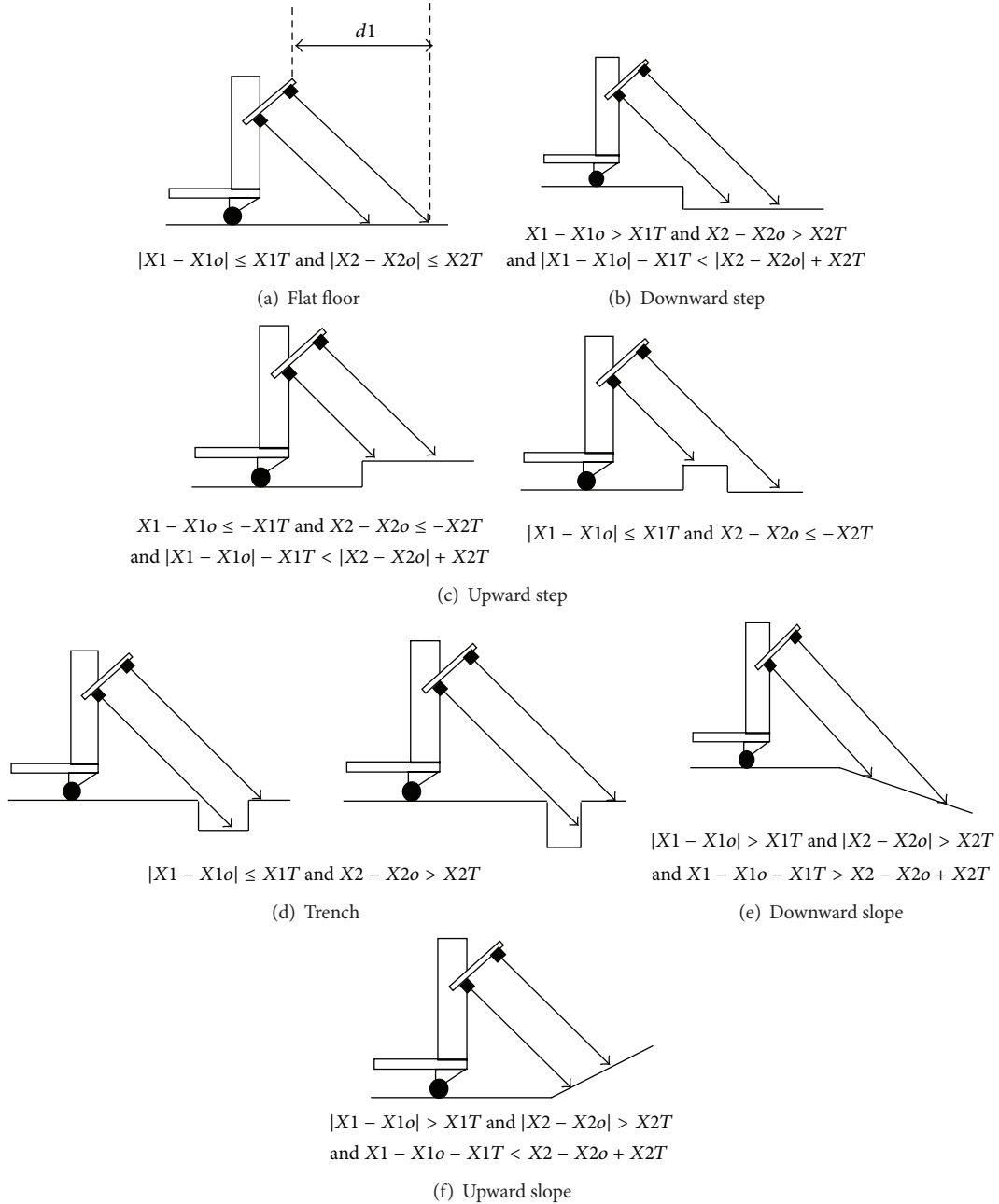


FIGURE 6: How to classify slopes, steps, and trenches. Mathematical algorithms are shown.

parameters of different slopes, steps, and trenches from data obtained is described.

(a) *Flat Floor*. When the sensing circuit block compares the range data to the threshold value given in the previous section, and PSD1 and PSD2 output data satisfying the condition of $S(\text{PSD1}) = \text{"FLAT"}$ & $S(\text{PSD2}) = \text{"FLAT"}$, the sensing circuit block "thinks" that it is on a flat floor (see Figure 6(a)). The equivalent mathematical relationship can be expressed as

$$|X1 - X1o| \leq X1T, \quad |X2 - X2o| \leq X2T. \quad (2)$$

(b) *Downward Step*. When PSD1 and PSD2 output data satisfying the following condition, the sensing circuit block "thinks" that it is facing a downward step (see Figure 6(b)):

$$\begin{aligned} X1 - X1o > X1T, \quad X2 - X2o > X2T, \\ |X1 - X1o| - X1T < |X2 - X2o| + X2T. \end{aligned} \quad (3)$$

(c) *Upward Step*. When PSD1 and PSD2 output data satisfying the following condition, the sensing circuit block "thinks" that it is facing an upward step (see Figure 6(c)):

$$|X1 - X1o| \leq X1T, \quad X2 - X2o \leq -X2T. \quad (4)$$

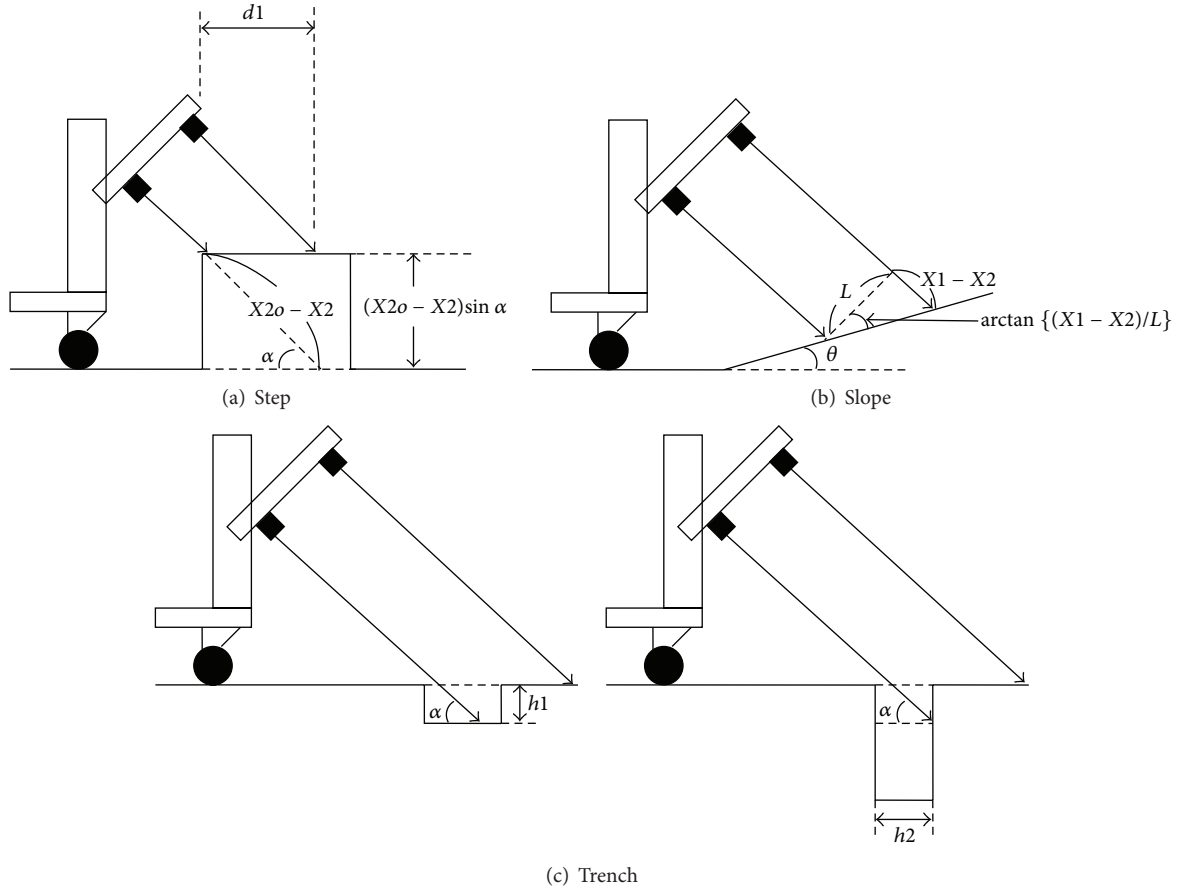


FIGURE 7: How to classify slopes, steps, and trenches. Characteristic parameter extraction method is shown.

(d) *Trench*. When PSD1 and PSD2 output data satisfying the following condition, the sensing circuit block “thinks” that it is facing a trench (see Figure 6(d)). In this case, the PSD sensor receives the IR signal reflected from the “side wall of the trench” or the IR signal reflected from the “bottom of the trench” as shown in Figure 7(c). The algorithm in the following form cannot distinguish these two cases:

$$|X1 - X1o| \leq X1T, \quad X2 - X2o > X2T. \quad (5)$$

Details are discussed later.

(e) *Downward Slope*. When PSD1 and PSD2 output data satisfying the following condition, the sensing circuit block “thinks” that it is facing a down slope (see Figure 6(e)):

$$|X1 - X1o| > X1T, \quad |X2 - X2o| > X2T, \quad (6)$$

$$X1 - X1o - X1T > X2 - X2o + X2T.$$

(f) *Upward Slope*. When PSD1 and PSD2 output data satisfying the following condition, the sensing circuit block “thinks” that it is facing an upward slope (see Figure 6(f)):

$$|X1 - X1o| > X1T, \quad |X2 - X2o| > X2T, \quad (7)$$

$$X1 - X1o - X1T < X2 - X2o + X2T.$$

4.2. *Equations to Calculate Slope Angle, Step Height, and Trench Width*. Slope angle, step height, and trench width can be calculated from range data ($X1, X2$). Figure 7 visualizes the trigonometric techniques used.

(a) *Step Height*. Step height $h1$ is calculated using (8). A schematic is shown in Figure 7(a):

$$h1 = (X2o - X2) \sin \alpha, \quad (8)$$

where α is the angle of sensor signal against a flat floor (here, $\alpha = 45^\circ$). When $h1 < 0$, a downward step is suggested, and when $h1 > 0$, an upward step is suggested.

(b) *Slope Angle*. Slope angle θ is calculated using (9). A schematic is shown in Figure 7(b):

$$\theta = \frac{\pi}{2} - \alpha - \arctan \left(\frac{(X1 - X2)}{L} \right). \quad (9)$$

When $\theta < 0$, a downward slope is suggested, and when $\theta > 0$, an upward slope is suggested.

(c) *Trench Width*. When the robot faces a trench, the sensor output differs between two cases: (i) the sensor signal is reflected from the bottom of the trench and (ii) the sensor signal is reflected from a side wall of the trench. Initially, the sensing circuit block cannot judge which is correct. A possible solution is to force the sensing circuit block to calculate two dimensions, trench depth and trench width. Trench depth $h1$ is calculated using the following equation:

$$h1 = (X2o - X2) \sin \alpha. \quad (10)$$

Trench width $h2$ is calculated by the next equation:

$$h2 = \max \{(X2o - X2) \cos \alpha\}, \quad (11)$$

where $\max\{X\}$ returns the maximal value of X .

Since all sensors are positioned so that their surfaces are angled at 45° against a flat floor, calculated values of $h1$ and $h2$ are identical. When the robot approaches the trench, the judgment of whether it can cross the trench depends on the diameter (D_w) of wheels of testing robot. Consider case (i). When D_w is much larger than the trench depth, the robot may be able to cross the trench. Consider case (ii). When D_w is much larger than the trench width, the robot can go over the trench. Therefore, the robot can pass through the trench for both cases, (i) and (ii), when $h1$ or $h2$ is much smaller than D_w . In other words, it is not necessary for us to distinguish cases (i) and (ii); we can apply (10) to decide whether the testing robot can go forward or not when the robot detects a trench.

In practical applications, the sensors do not always yield precise characteristic values to use the above equations because of various noises (including external disturbance) or spatial dispersion of the emitted IR signal. This suggests the need for some additional method to guarantee the accuracy or the reproducibility of the characteristic values and judgment reliability; detail is given in Section 5.

4.3. Measurement Results: Step Height, Slope Angle, and Trench Width. Measurement results of a step height for which the testing robot should stop in front of a step are summarized in Table 1; 1000 sensing trials were averaged in each event of obstacle discovery, and the medial filter number ($N1$) was 5. As is evident in Table 1, the variation of evaluated step height $h1$ is very small; the difference between the maximal value and the minimal value is about 3 mm for the upward step and about 6 mm for the downward step. We can see that the present evaluation technique does not always yield accurate data.

Measurement results of upward slope angle (θ) are shown in Table 2; 1000 sensing trials were averaged in each event of obstacles, and the medial filter number was 5 or 21. In the experiment, we assumed 3 cases for the horizontal distance ($d1$) between the front edge of the testing robot and the boundary of the flat floor and the slope, that is, 0 cm, 4 cm, and 7 cm. The slope angles were 20° , 15° , and 10° . It can be seen in Table 2 that the averaged value of θ basically increases

TABLE 1: Step height evaluation results in units of cm. 1000 sensing trials are averaged. The medial filter number ($N1$) is 5. The real step height ($h1$) is 1.30 cm.

	Upward step height (cm)	Downward step height (cm)
Mean value	1.30	-1.33
Max. value	1.45	-1.00
Min. value	1.18	-1.56
Variance	0.0057	0.0019

with $d1$. In this study, the slope angle evaluation algorithm does not estimate distance $d1$, and so the robot estimates characteristic values without stopping as it approaches the obstacle, resulting in a slight drop in accuracy. It is also seen in Table 2 that a large $N1$ value reduces the variation in estimated values, although many trials of measurement waste time before judgment. In addition, for $N1 = 21$, the difference between the maximal value and the minimal value is not always reduced.

Table 3 shows measurement results of downward slope angle. As is evident in Table 3, the variation of measurement results is very large in contrast to the upward slope values. This suggests the need to improve judgment reliability for practical applications.

5. Dynamic Detection of Obstacles in a Test Road

In Sections 5.1 to 5.4, we describe an algorithm to be used in practical situations. Section 5.5 details the accuracy of several evaluations.

5.1. Logical Flow. A schematic flow showing how the testing robot avoids obstacles is shown in Figure 8. First, when the left or the right sensor state is "NON-F", the testing robot changes its position so that the front line of the testing robot keeps being parallel to the border line of the obstacle and the flat floor. Next, the testing robot approaches the border line, again detects signals from the obstacle, and subsequently concludes whether the obstacle facing it is a slope, step, or trench. Finally, when the testing robot recognizes that the obstacle is a step, it calculates the tentative step height, compares the calculated value to the threshold value, and then concludes whether it has to avoid the obstacle or not. When the testing robot detects a slope or a trench, the testing robot traces the same logical flow. As just described, in order to successfully classify the obstacle and to get reliable characteristic values, causes of the errors in detecting the signals from the obstacle must be analyzed.

5.2. Aligning the Testing Robot to the Obstacle. In this section, we describe how the testing robot positions itself in the vicinity of the obstacle. For all obstacles, the testing robot should directly face the obstacle to maximize the detection accuracy; this is the most important point in detecting the

TABLE 2: Upward slope angle evaluation results in units of cm. 1000 sensing trials are averaged. The medial filter number ($N1$) is 5 or 21.

Medial filter number ($N1$)	Real slope angle (deg.)	$d1$ (cm)	Mean value (deg.)	Max. value (deg.)	Min. value (deg.)	Variance (deg.)
5	20.0	0.00	17.0	19.4	15.9	0.465
		4.00	18.9	21.0	17.5	0.561
		7.00	20.5	22.2	17.4	1.01
	15.0	0.00	14.7	16.5	13.0	0.218
		4.00	15.3	16.2	14.2	0.121
		7.00	16.1	18.7	12.8	1.89
	10.0	0.00	9.68	12.1	7.74	0.243
		4.00	10.3	13.4	7.84	1.36
		7.00	10.8	13.2	9.26	0.162
21	20.0	0.00	16.9	18.1	16.5	0.0951
		4.00	19.7	20.5	18.9	0.110
		7.00	19.3	22.2	18.4	0.284
	15.0	0.00	14.6	15.3	13.0	0.104
		4.00	14.8	17.6	13.5	0.873
		7.00	15.5	16.3	14.8	0.129
	10.0	0.00	9.80	10.5	9.11	0.0630
		4.00	9.89	10.7	8.21	0.114
		7.00	9.40	11.5	8.59	0.255

TABLE 3: Downward slope angle evaluation results in units of cm. 1000 sensing trials are averaged. The medial filter number ($N1$) is 21.

Real slope angle (deg.)	$d1$ (cm)	Mean value (deg.)	Max. value (deg.)	Min. value (deg.)	Variance (deg.)
-20.0	10.0	-19.9	-13.7	-23.2	2.06
	5.00	-18.9	-10.4	-25.4	7.69
	2.00	-18.4	-10.4	-27.2	10.3
-15.0	10.0	-14.6	-10.1	-18.0	1.41
	5.00	-14.2	-10.0	-21.7	3.76
	2.00	-14.5	-10.0	-19.8	4.35
-10.0	10.0	-10.6	-7.55	-13.7	1.45
	5.00	NA	NA	NA	NA
	2.00	NA	NA	NA	NA

parameters of an obstacle. This process is detailed below (see Figure 9).

- (1) First, the testing robot approaches the obstacle, receives range data, and examines whether the data satisfies the condition $S(\text{PSD2L}) = \text{"NON-F"} \wedge S(\text{PSD2R}) = \text{"FLAT"}$ (see Figure 9(a)).
- (2) The testing robot moves forward slightly, again receives range data, and examines whether the data satisfies the condition $S(\text{PSD2L}) = \text{"NON-F"} \wedge S(\text{PSD2R}) = \text{"NON-F"}$ (see Figure 9(b)).
- (3) When both PSD2L and PSD2R detect the "NON-F" signals, the testing robot moves as follows (see Figure 9(c)).
 - (i) When $S(\text{PSD2L}) = \text{"NON-F"}$, for example, the left motor reverses.

- (ii) When $S(\text{PSD2L}) = \text{"FLAT"}$, the left motor idles.
- (iii) When $S(\text{PSD2R}) = \text{"NON-F"}$, the right motor reverses.
- (iv) When $S(\text{PSD2R}) = \text{"FLAT"}$, the right motor idles.

The above algorithm ensures that the testing robot directly faces the obstacle.

- (4) When both motors stop, the algorithm has successfully terminated, and the testing robot approaches the obstacle again.

In the detecting process from (1) to (2), when the time interval of "NON-F" events of two sensors is longer than a certain value, the testing robot turns around before reaching the expected obstacle. In other words, the testing robot does

TABLE 4: Successful trials in detecting slopes. In the detection of the slope angle, detected data are averaged with 100 trials.

Real slope angle (deg.)	Successful detection rate	Mean value (deg.)	Max. value (deg.)	Min. value (deg.)	Variance (deg.)
-20.0	100/100	-20.0	-18.3	-24.1	0.926
-10.0	100/100	-12.7	-11.4	-14.0	0.261
10.0	100/100	10.3	10.9	9.52	0.0748
20.0	100/100	21.4	22.3	20.1	0.168

not estimate the vertical offset when the incident angle is very small.

The present algorithm yields a small degree of uncertainty on the testing robot's alignment due to the use of the threshold values $X1T$ and $X2T$. In this experiment, we found an alignment error of up to 10° . Later, we evaluate the influence of this alignment error on the determination of characteristic parameters.

5.3. How to Classify Obstacles Using Sensor Pairs. First, we explain how to classify slopes, steps, and trenches (see Figure 8). Using the simple method described in Section 4, the testing robot may, for example, incorrectly classify a real slope as a step or a flat floor. This erroneous judgment comes from sensor noise and relatively large threshold values ($X2T$ and $X1T$). When these threshold values are large, the erroneous judgment becomes more common. To avoid this difficulty, we force the testing robot to calculate characteristic values repeatedly and to get the mean or median value. This flow is described below.

- (1) When the robot detects an object, it calculates the characteristic values 20 times and stores these data in memory.
- (2) The robot classifies the obstacle according to the highest frequency of classification after the 20 trials.
- (3) When the frequency of "trench" exceeds 5 in the 20 trials, the testing robot classifies the obstacle as a trench.

We have confirmed that this majority-decision process reduces the frequency of erroneous judgment.

5.4. How to Calculate the Characteristic Values of a Specific Obstacle. Here we describe a method for calculating the characteristic values.

- (1) When upward (or downward) step height $h1$ is calculated, the testing robot calculates the mean of 20 trials.
- (2) When upward (or downward) slope angle θ is calculated, the testing robot calculates the mean of 20 trials.

Next, when the testing robot stops, the testing robot gets the data set of $X1L$, $X1R$, $X2L$, and $X2R$ 200 times. After determining the mean values of $X1L$, $X1R$, $X2L$, and $X2R$, they are labeled $X1L'$, $X1R'$, $X2L'$, and $X2R'$, respectively. Finally, using values of $X1L'$, $X1R'$, $X2L'$, and $X2R'$ and (9), the testing robot calculates the downward slope angle θ . This technique is very powerful in suppressing noise (as described

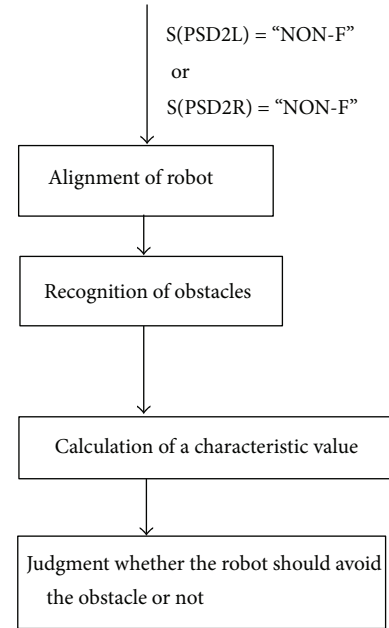


FIGURE 8: Algorithm for the robot to detect an obstacle and to avoid it.

in Sections 3 and 4). This benefit incurs the cost a 5 sec. delay in determining the downward slope angle. This limitation stems from the delay time of the PSD range sensor used in this study. We must employ a fast-processing PSD range sensor in the future.

5.5. Evaluation Results of Characteristic Values. Tables 4 to 6 show the characteristic values yielded by the logical process described in the previous section. Table 4 shows slope angle values extracted from signals given by sensors mounted on the testing robot; the offset value of sensor signals is considered in calculating the characteristic values, and the testing robot logically determines which obstacle has been encountered. As a result, the testing robot showed a very few errors in the classification of obstacles. Erroneous judgment, however, sometimes takes place in case of a gentle slope, which depends on threshold values of $X1T$ and $X2T$. The gentle slope sometimes gives the sensor a noisy signal that cannot be easily detected as meaningful data; in this case, the testing robot fails to correctly determine the slope angle. Raising the values of $X1T$ and $X2T$ yields more conclusive data by sacrificing the detectable range of slope angle.

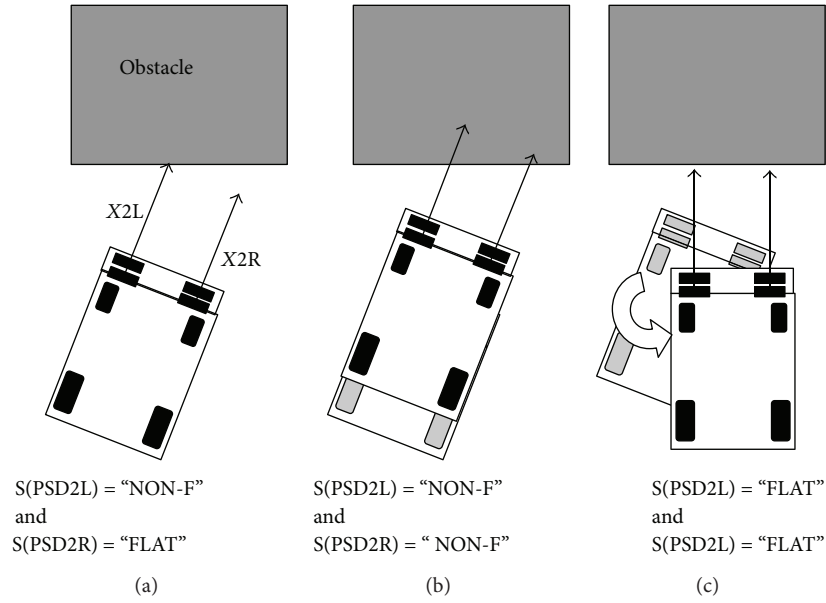


FIGURE 9: Front line alignment of the robot facing the obstacle.

Next, we discuss the accuracy of extracted slope angle θ . In the present experiment, the detectable range of slope angle θ is 20° to -10° , and the deviation of extracted slope angle is at most 2.5° . At $\theta = -20^\circ$, however, the uncertainty in detected angle rises to 4° . When the angle of the sensor-light incident on the object's surface becomes small, the intensity of reflected-light signal becomes very weak; this results in a lower dynamic range in the sensor's output signal. This is basically the same phenomenon described in Section 3.

Since the sensor emits an infrared light signal, the reflection rate of the light depends on the color of the object's surface. In addition, the sensor's output attenuates as the distance of the sensor from the object increases. When the color of the object is dark, the sensor's output falls as does its dynamic range. As a result, it is usually difficult to accurately detect the angle of a steep slope. One possible way to remove this difficulty is to use a PSD sensor whose distance-output-voltage characteristic is almost linear or to widen the window of the median filter, although this would increase the detection time.

Table 5 shows step height values that are recalculated by the sensor module with some offset angle when the testing robot approaches the step. Erroneous detection did not occur. The detected step height was more accurate than that described in Section 4, where the signal-filtering technique was applied to the step-height detection. When the maximal step height that forces the robot to back away is 13 mm, the maximal value of calculated step height should be 11 mm because the maximal variation in the output voltage signal of the sensor is equivalent to the step height of 2 mm.

Table 6 shows trench-width values determined by the testing robot; the offset value of sensor signals was considered in calculating the characteristic values.

The basic algorithm used to detect a trench was described in Section 3. When the testing robot approaches the trench

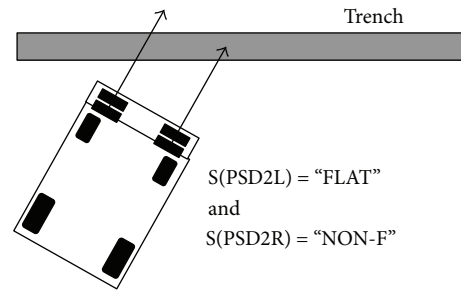


FIGURE 10: How to detect a trench.

at an oblique angle (see Figure 10), the correction of trench depth, used in the algorithm described in Section 5.1, cannot be employed because the algorithm assumes a direct approach to the trench. In this experiment, the testing robot was limited to approaching the trench at nearly 90° . This experiment was made on two trenches with different sizes.

As is seen in Table 6, when the trench width is reduced, the frequency of erroneous judgment rises. One cause is the incompleteness of the detection algorithm; the testing robot incorrectly judges the trench as a flat floor. One way of overcoming this difficulty is to reduce the values of $X1T$ and $X2T$. However, it raises the remaining electrical noise in the output signal. Another approach is to increase the diameter of the wheels of testing robot.

5.6. Advantage of the Method Proposed. Recently many mathematical techniques are proposed for the purpose of monitoring the changing environment [17], multirobot navigation [18], motion tracking [19], self-collision avoidance [20], blind juggler control [21], precise positioning [22], distant control [23], and motion grammar description [24].

TABLE 5: Successful trials in detecting steps. In the detection of the step height, detected data are averaged with 100 trials.

Real step height (cm)	Successful detection rate	Mean value (cm)	Max. value (cm)	Min. value (cm)	Variance (cm)
1.30	100/100	1.37	1.47	1.17	0.0017
-1.30	100/100	-1.21	-1.11	-1.33	0.0022

TABLE 6: Successful trials in detecting trenches. In the detection of the trench width, detected data are averaged with 100 trials.

Real trench dimension	Successful detection rate	Mean value (cm)	Max. value (cm)	Min. value (cm)	Variance (cm)
1.30 (cm) \times 1.30 (cm)	55/100	-0.94	-0.85	-1.26	0.0098
2.00 (cm) \times 2.00 (cm)	90/100	-1.67	-1.24	-2.32	0.0310

In most proposals, the algorithm is complex [17–20], and/or the time derivatives are applied to the parameter analysis [20, 21]. The time derivative frequently yields extra noise in the signal analysis [21]. On the other hand, the use of many controllable degrees of freedom (DOF) [19, 20, 22] leads us to a large RMS of errors. Polynomial formalism for the position control [23] requests many topological definitions to realize reliable forward kinematics. Application of motion grammar to robots [24] also requests many possible logical patterns to avoid undesirable actions.

As is suggested in the previous articles, complex mechanics and complex actions request complex algorithms, resulting in a high cost and much difficulty. We think that the technique applied to some robots (cleaning robot, visitor-guide robot, and so on) requests simplicity of electronics and software from the cost of product. Therefore, the method proposed here has an advantage from the point of view of system volume and its cost.

6. Concluding Remarks

We have proposed a simple method for detecting the step height, slope angle, and trench width using four PSD range sensors (GP2D12) and have examined the reproducibility and accuracy of characteristic parameter detection. Detection error of upward slope angle is about 2.5° , while the detection error for downward slope angles exceeding 20° is very large. To reduce these errors, we have to use a range sensor that offers better range-voltage performance, or we have to increase the trial frequency so as not to increase the detection delay. Step height is extracted with an error of ± 1.5 mm. The current algorithm for trench width is not so accurate. It is suggested that an additional method must be introduced to advance the obstacle detection technique. However, this study has demonstrated that obstacle detection is basically possible without image processing.

Appendix

Median Filter Algorithm

In this paper, we used the following algorithm in order to reduce the electrical noise in the original signal. First, we get

N datum points from the microcontroller. After sorting the N data ($D[1]$ to $D[N]$), we extract the maximal value, D_{\max} , and the minimal value, D_{\min} , from all data and order the data set ($n = 1$ to N); that is, $D[1] = D_{\max}$ and $D[N] = D_{\min}$. Finally, we get $D[N/2]$ as the medial value. Sets of $D[N/2]$ are plotted in Figures 4 and 5.

References

- [1] J. Velagic, B. Lacevic, and B. Perunicic, "A 3-level autonomous mobile robot navigation system designed by using reasoning/search approaches," *Robotics and Autonomous Systems*, vol. 54, no. 12, pp. 989–1004, 2006.
- [2] T. Miyake, H. Ishihara, R. Shoji, and S. Yoshida, "Development of small-size window cleaning robot a traveling direction control on vertical surface using accelerometer," in *Proceedings of the IEEE International Conference on Mechatronics and Automation (ICMA '06)*, pp. 1302–1307, Luoyang, China, June 2006.
- [3] H. Kawata, A. Ohya, S. Yuta, W. Santosh, and T. Mori, "Development of ultra-small lightweight optical range sensor system," in *Proceedings of the IEEE IRS/RSJ International Conference on Intelligent Robots and Systems (IROS '05)*, pp. 1078–1083, August 2005.
- [4] P. Gemeiner and M. Vincze, "Motion and structure estimation from vision and inertial sensor data with high speed CMOS camera," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1853–1858, Barcelona, Spain, April 2005.
- [5] H. Zhang, S. Liu, and S. X. Yang, "A hybrid robot navigation approach based on partial planning and emotion-based behavior coordination," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '06)*, pp. 1183–1188, Beijing, China, October 2006.
- [6] J. Snyder, Y. Silverman, Y. Bai, and M. A. Maclver, "Underwater object tracking using electrical impedance tomography," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '12)*, pp. 520–525, Vilamoura, Portugal, October 2012.
- [7] D. F. Wolf, G. S. Sukhatme, D. Fox, and W. Burgard, "Autonomous terrain mapping and classification using hidden Markov models," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2026–2031, Barcelona, Spain, April 2005.
- [8] H. J. Kwak, D. H. Lee, J. M. Hwang, J. H. Kim, C. K. Kim, and G. T. Park, "Improvement of the inertial sensor-based localization

- for mobile robots using multiple estimation windows filter,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '12)*, pp. 876–881, Vilamoura, Portugal, October 2012.
- [9] D. H. Kim and S. Shin, “Local path planning using a new artificial potential function composition and its analytical design guidelines,” *Advanced Robotics*, vol. 20, no. 1, pp. 115–135, 2006.
- [10] M. Madry, C. H. Ek, R. Detry, K. Hang, and D. Kragic, “Improving generalization for 3D object categorization with global structure histograms,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '12)*, pp. 1379–1386, Vilamoura, Portugal, October 2012.
- [11] Y. Hirata, A. Hara, and K. Kosuge, “Motion control of passive intelligent walker using servo brakes,” *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 981–990, 2007.
- [12] Y. Hirata, A. Hara, and K. Kosuge, “Passive-type intelligent walking support system ‘RT Walker,’” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '04)*, pp. 3871–3876, October 2004.
- [13] O. Y. Chuy Jr., Y. Hirata, Z. Wang, and K. Kosuge, “A control approach based on passive behavior to enhance user interaction,” *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 899–908, 2007.
- [14] A. Kim and R. M. Eustice, “Real-time visual SLAM for autonomous underwater hull inspection using visual saliency,” *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 719–733, 2013.
- [15] <http://pdf1.alldatasheet.com/datasheet-pdf/view/166476/AD/ADUC7027.html>.
- [16] <http://www.findthatpdf.com/search-4444820-hPDF/download-documents-sharppp2d12-15.pdf.htm>.
- [17] S. L. Smith, M. Schwager, and D. Rus, “Persistent robotic tasks: monitoring and sweeping in changing environments,” *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 410–426, 2012.
- [18] H. G. Tanner and A. Boddu, “Multiagent navigation functions revisited,” *IEEE Transactions on Robotics*, vol. 28, no. 6, pp. 1346–1359, 2012.
- [19] N. Hunger, M. Baumann, J. A. Long, and J. Troccaz, “A 3-D ultrasound robotic prostate brachytherapy system with prostate motion tracking,” *IEEE Transactions on Robotics*, vol. 28, no. 6, pp. 1382–1397, 2012.
- [20] A. Dietrich, T. Wimboeck, A. Albu-Schaeffer, and G. Hirzinger, “Integration of reactive, torque-based self-collision avoidance into a task hierarchy,” *IEEE Transactions on Robotics*, vol. 28, no. 6, pp. 1278–1293, 2012.
- [21] P. Reist and R. D’Andrea, “Design and analysis of a blind juggling robot,” *IEEE Transactions on Robotics*, vol. 28, no. 6, pp. 1228–1243, 2012.
- [22] V. Chalvet, Y. Haddab, and P. Lutz, “A microfabricated planner design microrobot for precise positioning based on bistable modules,” *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 641–649, 2013.
- [23] N. Rojas and F. Thomas, “The univariate closure conditions of all fully parallel planar robots derived from a single polynomial,” *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 758–765, 2013.
- [24] N. Dantam and M. Stilman, “The motion grammar: analysis of a linguistic method for robot control,” *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 704–718, 2013.

