

AN EMBEDDED BEAMFORMER FOR A PID-BASED TRAJECTORY SENSING FOR AN AUTONOMOUS VEHICLE

Patrick Kapita Mvemba¹⁾, Aimé Lay-Ekuakille²⁾, Simon Kidiamboko³⁾,
Md Zia Uhr Rahman⁴⁾

- 1) University of Siena, Department of Information Engineering and Mathematics, 53100 Siena, Italy (kapita@diism.unisi.it)
- 2) University of Salento, Department of Innovation Engineering, 73100 Lecce, Italy (✉ aime.lay.ekuakille@unisalento.it, +39 0832 29 7822)
- 3) ISTA University, Department of Electronics, Kinshasa, DR Congo (simon.kidiamboko@ista.ac.cd)
- 4) KL University, Department of E.C.E., Green Fields, Vaddeswaram, Guntur-522502, India (mdzr55@gmail.com)

Abstract

Beamforming is an advanced signal processing technique used in sensor arrays for directional signal transmission or reception. The paper deals with a system based on an ultrasound transmitter and an array of receivers, to determine the distance to an obstacle by measuring the time of flight and – using the phase beamforming technique to process the output signals of receivers for finding the direction from which the reflected signal is received – locates the obstacle. The embedded beam-former interacts with a PID-based line follower robot to improve performance of the line follower navigation algorithm by detecting and avoiding obstacles. The PID (proportional-integral-derivative) algorithm is also typically used to control industrial processes. It calculates the difference between a measured value and a desired set of points, then attempts to minimize the error by adjusting the output. The overall navigation system combines a PID-based trajectory follower with a spatial-temporal filter (beamformer) that uses the output of an array of sensors to extract signals received from an obstacle in a particular direction in order to guide an autonomous vehicle or a robot along a safe path.

Keywords: autonomous vehicle, beamforming, trajectory follower, ultrasonic obstacle detector, measurements, sensors.

© 2018 Polish Academy of Sciences. All rights reserved

1. Introduction

The simplest line following algorithm is the one that uses only one sensor. The sensor is placed in a position that is a little off-centred to one of the sides. When the sensor detects no line, the *autonomous vehicle* (AV) or robot moves to the left and when the sensor detects the line the AV moves to the right. A drawback of this simplest algorithm is that the line following is not smooth and the AV keeps wavering to the left and right on the track, wasting battery power and time. Adding sensors on both sides of the AV and placing them so that they just sense the line on either side can improve the algorithm. The algorithm will decide to move forward if both the sensors sense the line or to move left if only the left sensor senses the line or trajectory and

moves right if only the right sensor senses the line [1]. An AV with this improved algorithm will follow the line faster than the simplest one with one sensor, however it will still wobble along the line and may not be fast enough. A much better algorithm to follow the line is the one using a PID controller [2, 3]; it makes the line following process much smoother, faster and efficient. The PID is a popular loop feedback control extensively used in industrial control systems. A conventional robot will follow the line by oscillating a lot along the line, wasting valuable time and battery power and will sometimes overshoot the line. In the experiment, a line following robot distinguishing a black line from a white surface and data from infrared reflectance sensors will be used to control the motors with the PID algorithm [4]. In this paper a beamformer is implemented on the line follower robot to enable obstacle avoiding navigation while following the line. The beamformer [5] is an efficient technique in which an array of sensors is employed to achieve maximum reception in a specified direction by a signal arriving from a desired direction and rejecting signals of the same frequency coming from other directions [6]. The most important problem in sensor array signal processing using acoustic scanning, is the estimation of the coordinates of a source emitting a signal (passive localization) or a point target illuminated by an external signal (active localization). A point in a three-dimensional space is defined by three parameters, namely: range, azimuth and elevation. The range is often measured by means of the return time of travel in active systems and by means of time delays measured by a number of sensors in passive systems. The azimuth and elevation angles are obtained from the measurements of *direction of arrival* (DOA) by an array of sensors. A point of power concentration is assumed to be a source. Three methods are mostly used: the beam formation detailed in [7], Capon spectrum and maximum entropy spectrum. The last two methods fall into the nonlinear category, while the first method – to the linear one. The basic difference between the linear and nonlinear methods lies in their response to an input which consists of the sum of two or more uncorrelated signals. The output of a linear method will be the sum of the spectra of input signals, whereas the output of a nonlinear method may contain an additional cross-term [7]. An AV can also use, in order to increase its accuracy, the particle swarm optimization for adapting different trajectories and speed. The controller is designed for a nonlinear vehicle model [8]. The design should take into account energy minimization for the AV by including a specific control system [9]. Other improvements can be obtained by means of a nonlinear *model predictive control* (MPC) to prevent clashing into an obstacle at high speed by considering controlling the centre of gravity [10]. Each vehicle aims at estimating its own state relying on locally available measurements and limited communication with other vehicles in the neighbourhood using a discrete-time Kalman filtering formulation [11] with a sparsity constraint on the gain parameter, and specific simulations of field data referring to speed and traffic volumes [12] from existing roundabouts.

2. Delay and sum beamforming for object detection

A given array of sensors can be organized in any shape in space, where the position of each sensor can be described by its coordinates like $p = (p_x, p_y, p_z)$. If a plane wave signal $f(t, p)$ is arriving at a particular point in space, and positions of each sensor in space are different, the signals received by each sensor will be the same as the original one with a time delay depending on the position of the sensor. To describe the signal received by each sensor, we can use the following vector:

$$\overline{f(t, p)} = \begin{bmatrix} f_{p0}(t) \\ f_{p1}(t) \\ \vdots \\ f_{p_{N-1}}(t) \end{bmatrix} = \begin{bmatrix} f(t - \tau_0) \\ f(t - \tau_1) \\ \vdots \\ f(t - \tau_{N-1}) \end{bmatrix}, \tag{1}$$

where N is the number of elements in the array and τ_i is the time delay associated with the position of the element. The Bartlett beamformer, or delay-and-sum beamforming is a robust beamforming algorithm; it is the conventional beamformer. The delay-and-sum beamformer applies a delay and an amplitude weight to the output of each sensor, and then sums the resulting signals. Delays are chosen to maximize the array sensitivity to incoming waves from a particular direction. By adjusting the delays, the look-direction array can be steered towards the source, and the waveforms captured by individual sensors add constructively. This means that signals at particular angles produce the constructive interference, while others cause the destructive interference [13]. The operation of a delay-and-sum beamformer is depicted in Fig. 1; it consists of applying a delay Δ_m and an amplitude weight ω_m to the output of each sensor, and then summing the resulting signals as described in [14].

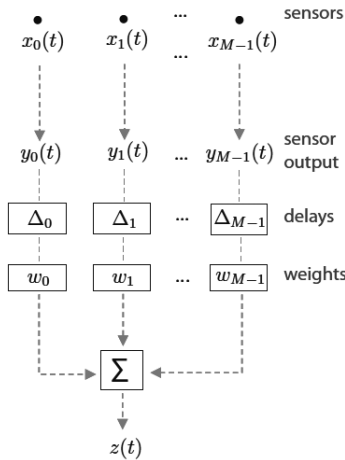


Fig. 1. The conventional anddelay-and-sum beamforming.

The delays are chosen to maximize the array sensitivity to waves propagating from a particular direction. By adjusting the delays, the look-direction array can be steered towards the source, and the waveforms captured by individual sensors add constructively. This operation is sometimes called stacking. Weighting different sensors of the array differently may be seen as a gain factor for individual sensors, and it enhances the shape and reduces sidelobe levels of the listening beam. As opposed to the adaptive methods, the sensor weights for a delay-and-sum beamformer are chosen in advance and independently from the received waveform [15, 16]. The delay-and-sum beamformer output in the time domain is then:

$$z(t) = \sum_{m=0}^{M-1} w_m \cdot y_m(t - \Delta_m). \tag{2}$$

The basic idea behind beamforming is to steer the listening direction of the array on different points in the scanning plane, measure the power from each point, and interpolate the values to create an image. When the steering direction coincides with a source, the maximum output power will be observed [17]. By interpolating the measured output power from all the scanning points, it is possible to colour the spatial power (power across the scanning plane) and make an acoustic image, as shown in Fig. 2.

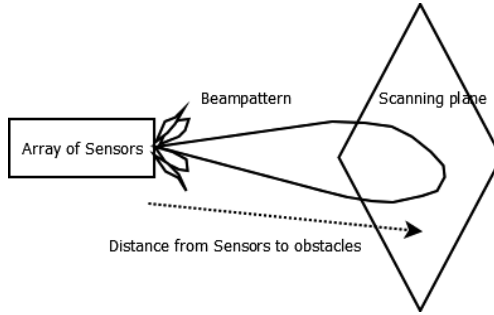


Fig. 2. When the steering direction coincides with a source or target, the maximum output power is observed.

To characterize the array sensitivity of our autonomous vehicle to a single ultrasound frequency wave from an arbitrary incidence angle when using the delay-and-sum beamformer, we assume that the incidence angle in spherical coordinates is then given as elevation θ , which is the normal incidence angle, and azimuth ϕ which is the angle in the XY plane as depicted in Fig. 3.

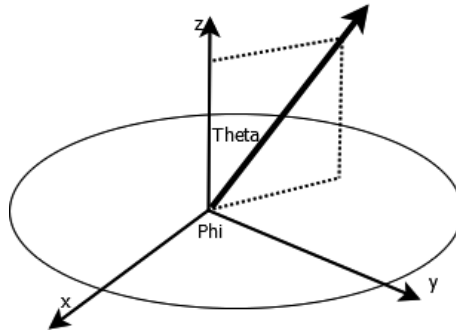


Fig. 3. A spherical coordinate system illustrated with elevation θ and azimuth ϕ .

The input of a single sensor is:

$$y_m(t) = e^{j(\omega^0 t - \vec{k}^0 \cdot \vec{x}_m)}, \tag{3}$$

where $\omega = 2\pi f$ is the frequency of the input signal with frequency f . The wavenumber vector (or wave vector) $\vec{k} = [k_x, k_y, k_z]$ is a propagation vector giving both magnitude and direction of arrival of the incident plane wave. The superscript 0 over \vec{k}^0 and ω^0 is to denote that the wave has a specific frequency ω^0 and a specific direction given by the wave vector \vec{k}^0 , which may be different from the direction \vec{k} which the array is steered to. As before, $\vec{x}_m = [x_m, y_m, z_m]$ is the position in space of the receiving sensor. By using the same input signal as in (3), the delayed signal can be expressed as:

$$\begin{aligned} y_m(t - \Delta_m) &= e^{j(\omega^0(t - \Delta_m) - \vec{k}^0 \cdot \vec{x}_m)} \\ &= e^{j(\omega^0 t - \vec{k}^0 \cdot \vec{x}_m)} \cdot e^{-j\omega^0 \Delta_m} \\ &= y_m(t) \cdot e^{-j\omega^0 \Delta_m}. \end{aligned} \tag{4}$$

The signal $y_m(t)$ is a signal received from an individual sensor, different for different sensors, and $e^{-j\omega^0\Delta_m}$ represents the phase delay associated with the signal from the m -th ultrasound sensor. The delay-and-sum beam-former output can again be expressed as in (1), as follows:

$$z(t) = \sum_{m=0}^{M-1} w_m \cdot y_m(t) \cdot e^{-j\omega^0\Delta_m}. \quad (5)$$

Now, we want to choose a set of delays as the phase shifts steer the beam listening direction to the direction of the vector \vec{k} which can be different from the wave propagation direction \vec{k}^0 . That is, the delays are chosen as follows:

$$\Delta_m = \frac{\vec{k}}{\omega^0} \cdot \vec{x}_m \quad (6)$$

and the total response from (5) may be calculated as follows:

$$\begin{aligned} z(t) &= \sum_{m=0}^{M-1} w_m \cdot y_m(t) \cdot e^{-j\omega^0(\Delta_m = \frac{\vec{k}}{\omega^0} \cdot \vec{x}_m)} \\ &= \sum_{m=0}^{M-1} w_m \cdot y_m(t) \cdot e^{j\vec{k} \cdot \vec{x}_m}. \end{aligned} \quad (7)$$

with $e^{j\vec{k} \cdot \vec{x}_m}$ as the phase delay associated with each individual sensor. Now to characterize the output of the delay-and-sum beam-former, we write $y_m(t)$ as in (3). Let us substitute it into (7):

$$\begin{aligned} z(t) &= \sum_{m=0}^{M-1} w_m \cdot y_m(t) \cdot e^{j\vec{k} \cdot \vec{x}_m} \\ &= \sum_{m=0}^{M-1} w_m \cdot e^{j(\omega^0 t - \vec{k}^0 \cdot \vec{x}_m)} \cdot e^{j\vec{k} \cdot \vec{x}_m} \\ &= \sum_{m=0}^{M-1} w_m \cdot e^{j(\vec{k} - \vec{k}^0) \cdot \vec{x}_m} \cdot e^{j\omega^0 t} \\ &= w(\vec{k} - \vec{k}^0) \cdot e^{j\omega^0 t}, \end{aligned} \quad (8)$$

with

$$w(\vec{k}) = \sum_{m=0}^{M-1} w_m e^{j\vec{k} \cdot \vec{x}_m}. \quad (9)$$

Equation (9) is our array pattern, which is a function of the position of the sensors in the array and the used weights. In the case of uniform shading where the weights are all equal, the array pattern depends only on the array geometry. The function $w(\vec{k} - \vec{k}^0)$ given in (8) is called the beam-pattern of the array. We see how the beam-pattern describing a monochromatic signal $e^{j\omega^0 t}$ propagating in a direction given by \vec{k}^0 with a frequency ω^0 is attenuated by a delay-and-sum beam-former steered towards the direction \vec{k} . The beam-pattern will have the maximum output when the steering direction coincides with the wave direction of propagation, that is we set $\vec{k} = \vec{k}^0$. Returning to the notation given in (7) and including the phase delays in the received signal vector $Y = y_m(t) \cdot e^{j\vec{k} \cdot \vec{x}_m}$, we may write (7) in the vector notation as follows:

$$z(t) = \sum_{m=0}^{M-1} w_m \cdot \left(y_m(t) \cdot e^{j\vec{k} \cdot \vec{x}_m} \right) = w^H Y, \quad (10)$$

where Y is an $M \times 1$ vector of the signals received from all sensors with their associated phase delays:

$$Y = \begin{bmatrix} y_0(t) \cdot e^{j\vec{k}\vec{x}_0} \\ y_1(t) \cdot e^{j\vec{k}\vec{x}_1} \\ \vdots \\ y_{M-1}(t) \cdot e^{j\vec{k}\vec{x}_{M-1}} \end{bmatrix}, \quad (11)$$

where w is an $M \times 1$ vector of weights for individual sensors and H denotes a complex conjugate transpose.

$$w = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{M-1} \end{bmatrix}. \quad (12)$$

By using the vector notation given in (10) and assuming that we have already steered the array to the desired direction, we can calculate the power, or the variance, of the output signal as follows:

$$\begin{aligned} P(z(t)) &= \sigma^2 = E\{|z(t)|^2\} \\ &= E\{(w^H Y)(w^H Y)^H\} \\ &= E\{w^H Y Y^H w\} \\ &= w^H E\{Y Y^H\} w \\ &= w^H R w. \end{aligned} \quad (13)$$

The above equation gives the power of the output of the beam-former in the steered direction, where $R = E\{Y Y^H\}$ is the correlation matrix of the data. Now, let us suppose we want to measure the output power as a function of steering directions, or scanning angles. In (7) the phase delays each associated with an individual sensor $e^{j\vec{k}\cdot\vec{x}_m}$ give a so called steering vector, denoted as e , and governing how we want to steer the beam of our array:

$$e = e^{j\vec{k}\cdot\vec{x}_m} = \begin{bmatrix} e^{j\vec{k}\cdot\vec{x}_0} \\ e^{j\vec{k}\cdot\vec{x}_1} \\ \vdots \\ e^{j\vec{k}\cdot\vec{x}_{M-1}} \end{bmatrix}. \quad (14)$$

The wave vector is related to the Cartesian coordinates by the following equations:

$$\begin{aligned} k_x &= k \sin \theta \cos \phi \\ k_y &= k \sin \theta \sin \phi \\ k_z &= k \cos \theta, \end{aligned} \quad (15)$$

where x – is a component of the wave vector; k_x determines the rate of change of phase of a propagating plane wave in the x -direction. The same definitions are valid for the y - and z -directions. The wavenumber k is equal to $2\pi\lambda$ or $2\pi/f$. Then, the steering vector depends on frequency and propagation direction of the incoming plane wave, and can be expressed in terms of wavelength λ , elevation θ and azimuth ϕ . Usually planar 2D arrays with the elements positioned in

the same plane will be used, so that the z coordinates of the sensors will be equal to zero. This means that the dependence on z and k_z may be omitted, and the steering vector can be written as:

$$e = e^{j\vec{k} \cdot \vec{x}_m} = \begin{bmatrix} e^{j\frac{2\pi}{\lambda}(\sin\theta\cos\phi \cdot x_0 + \sin\theta\sin\phi \cdot y_0)} \\ e^{j\frac{2\pi}{\lambda}(\sin\theta\cos\phi \cdot x_1 + \sin\theta\sin\phi \cdot y_1)} \\ \vdots \\ e^{j\frac{2\pi}{\lambda}(\sin\theta\cos\phi \cdot x_{M-1} + \sin\theta\sin\phi \cdot y_{M-1})} \end{bmatrix}. \quad (16)$$

In (13) we have already assumed that the array was steered to the correct direction before calculating the power. Now, if we want to calculate the energy for an arbitrary direction, instead, we must be aware that since the received signal vector Y in (13) have phase delays included, this must mean that R also is a function of the steering vector e , that is $R(e) = e^H R e$. Now, let us suppose that we want to measure the output power as a function of scanning angle, or rather as a function of steering vector. The output power calculated as a function of steering vector is named the steered response, and it is the power of the beam-former output in the frequency domain. This array output power spectral density may then be expressed by using the correlation matrix and the steering vector as follows:

$$P(e) = w^H R(e) w = w^H (e^H R e) w. \quad (17)$$

For a uniform array where all sensors have an equal weight, the above expression is reduced to:

$$P(e) = e^H R e. \quad (18)$$

3. Implementation of PID algorithm

The PID controller is by far the most common control algorithm. Most feedback loops are controlled by this algorithm or minor variations of it. The PID controller calculates the difference between a measured variable and a desired set point [6, 7, 13]. The controller attempts to minimize the error by adjusting the input. The PID controller has three values: Proportional, Integral and Derivative – denoted P, I, and D, respectively. Heuristically, these values can be interpreted in terms of time: P depends on the present error, I depends on the accumulation of past errors, and D is a prediction of future errors, based on the current rate of change. By simultaneous combining the three terms the algorithm maintains the desired set point with minimal errors. This can be mathematically expressed as follows:

$$con(t) = K \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) + T_d \frac{de(t)}{dt} \right), \quad (19)$$

where con is the controller variable and e is the controller error ($e = y_{sp} - y$). The controller variable is thus the sum of the three terms: the P-term (which is proportional to the error), the I-term (which is proportional to the integral of the error), and the D-term (which is proportional to the derivative error). The controller parameters are: the proportional gain K , the integral time T_i and the derivative time T_d . The proportional action is given as:

$$con(t) = K e(t) + con_b. \quad (20)$$

It is simply proportional to the control error. The variable con_b is a bias or a reset. The integral action is given as:

$$con_0 = K \left(e_0 + \frac{e_0}{T_i} t \right). \quad (21)$$

Its main function is to make sure that the process output agrees with the set point in a steady state. The derivative action associated with the proportional is given as:

$$con(t) = K \left(e(t) + T_d \frac{de(t)}{dt} \right). \quad (22)$$

Its purpose is to improve the closed-loop stability. If we use the Taylor series expansion of $e(t + T_d)$, we get:

$$e(t + T_d) \approx e(t) + T_d \frac{de(t)}{dt}. \quad (23)$$

The control signal is thus proportional to an estimate of the control error at time T_d ahead, where the estimate is obtained by linear extrapolation. In this paper, the measured process variable is the position of the sensor array for the line detection, c is the controller variable. The controller output signal consists of the speeds of the two DC motors. The basic idea of the algorithm is as follows:

previous_error = 0

integral = 0

start:

error = *setpoint* - *measured_value*

integral = *integral* + *error* * *dt*

derivative = (*error* - *previous_error*) / *dt*

output = $K_p * error + K_i * integral + K_d * derivative$

previous_error = *error*

wait(dt)

goto start

In this paper, the C function that performs the PID computations is illustrated below:

```
int pid( ) {
```

```
const int set_point = 250;
```

```
double Kp, Ki, Kd;
```

```
position = int(sensors_average/sensors_sum);
```

```
proportional = position - set_point;
```

```
integral = integral + proportional;
```

```
derivative = proportional - last_proportional;
```

```
last_proportional = proportional;
```

```
output = int(proportional * Kp + integral * Ki + derivative * Kd);
```

```
return output; }
```

Simulink PID was used to tune the controller without using a complex mathematical model of the autonomous vehicle. We adjusted the proportional gain of the controller till a suitable response was obtained. As a rule of thumb, we increased the value of the derivative gain to be equal to $1/10^{th}$ of the proportional gain and then we adjusted the derivative gain and observed the change in the AV behaviour till a suitable behaviour was obtained. Increasing the proportional gain will increase the amount by which the AV turns. An increase of the derivative gain would suppress the magnitude of oscillations by the robot so we set the integral value to 1, tuned k_p to 0.22 and finally k_d to 0.04, in respect to the performance indices, as shown in Fig. 4.

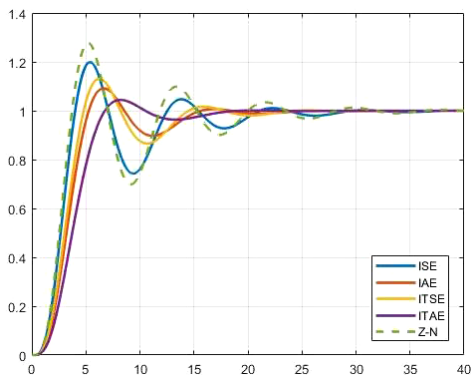


Fig. 4. The performance indices used when tuning the controller are: the *Integral of Squared Error* (ISE), the *Integral of Absolute Error* (IAE), the *Integral of Time Multiply Squared Error* (ITSE), the *Integral of Time Multiply Absolute Error* (ITAE).

4. Results

On one hand the controller computes data from the trajectory sensor array to adjust the AV position, and on the other hand it performs beamforming measurements [18] based on data from the acoustic sensor array to detect an obstacle along the path. The closed-loop system is depicted in Fig. 5. Positions of the sensors are presented in Fig. 6, whereas the trajectory – in Fig. 7.

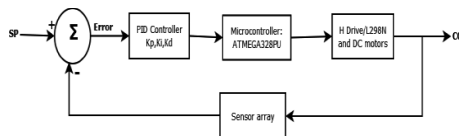


Fig. 5. A diagram of the closed-loop system which performs the line following process. The data from the line sensor array are used to correct the robot position while the Atmega controller handles the data from the acoustic sensor array, related to obstacle detection.

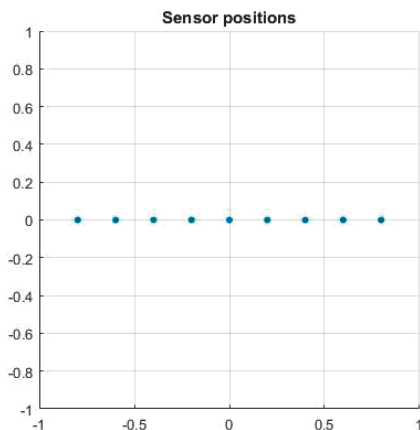


Fig. 6. The acoustic sensor array is mounted in front of the AV so that it can perceive incident signals from objects along the path.

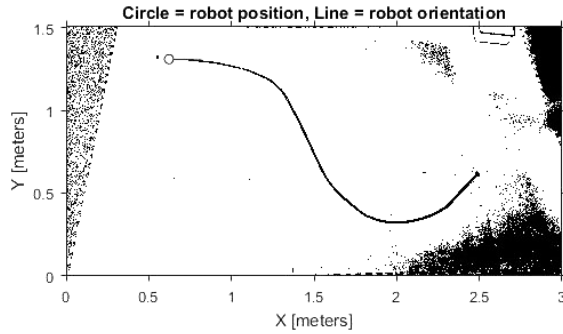


Fig. 7. Part of the AV trajectory during testing.

The corresponding directional pattern and beampattern are shown in Fig. 8 and Fig. 9, respectively. They are useful for showing precision and accuracy of the designed AV beamformer in detecting the trajectory as a directional antenna.

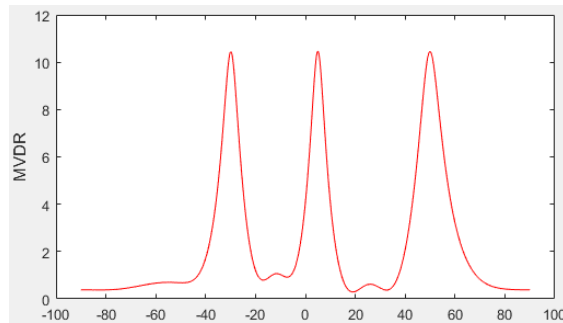


Fig. 8. A theta (degrees) directional pattern of the sensor array sensitivity estimated with MVDR.

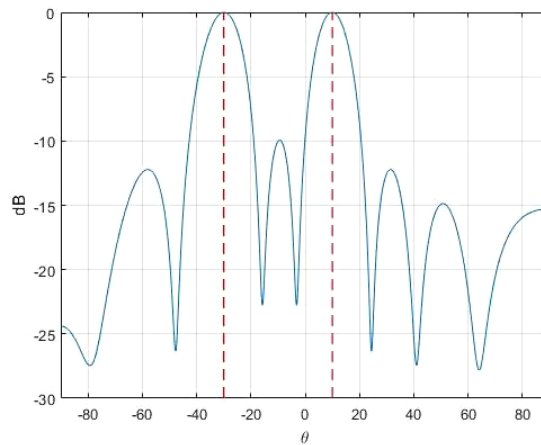


Fig. 9. A beam-pattern of the embedded delay-and-sum beamformer (for the arrival angles of -30 and 10 degrees).

The aperture angle depicted “theta” demonstrates the capability of the embedded beamformer to keep its trajectory (path) with a certain (good) accuracy. The dynamic range represented in Fig. 10 enables simultaneous measurements of the distance with an optimal level of signal received from an obstacle. The value of 6.0 dB also ensures low distortion within an area of 1.5×1.5 m. That confirms that a high dynamic range enables the embedded processor (Atmega) to be very precise in processing. The proposed beamformer is a MVDR (*minimum variance distortionless response*); the system response, as PID, is indicated in Fig. 11 with acceptable smoothing. The dynamic range at 3.0 dB is presented in Fig. 12, in terms of angle.

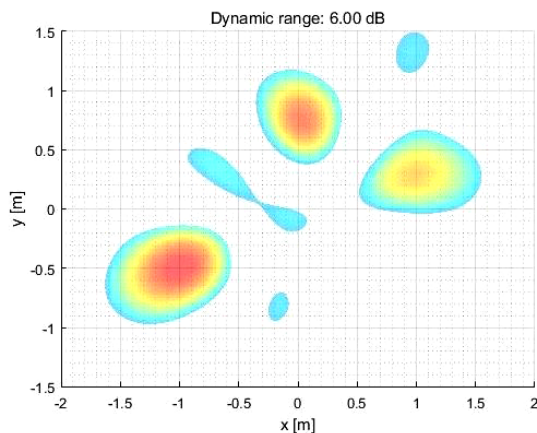


Fig. 10. The embedded beamformer dynamic range at 6.0 dB.

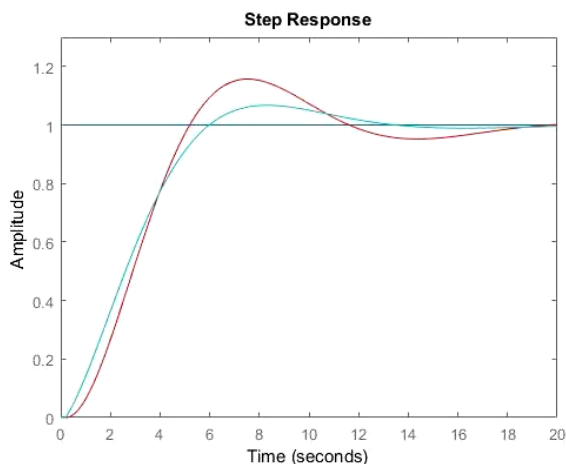


Fig. 11. The blue line represents the PID action with parameters that smooth the line following process, ($P = 0.22$, $I = 1$, and $D = 0.04$) after a fluctuation.

The AV is placed on a black line drawn on a white ground surface, as shown in Fig. 7. It has to follow the trajectory and to sense its surroundings. If there are obstacles spreaded along the path, it decides to leave the line to avoid the nearest obstacle in front of it, and afterwards comes back

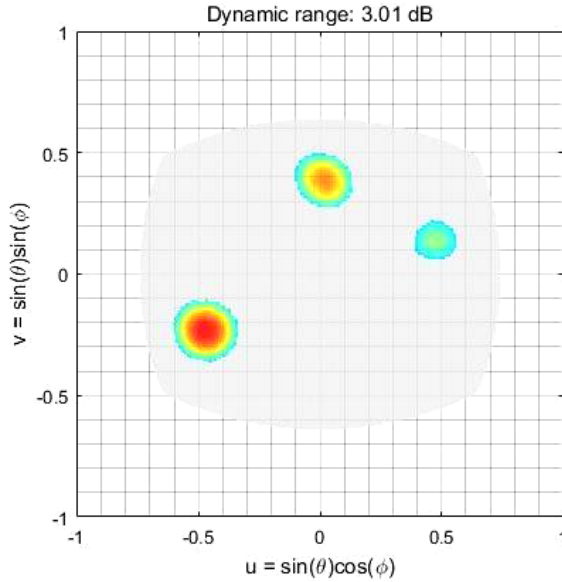


Fig. 12. The embedded beamformer dynamic range at 3.0 dB.

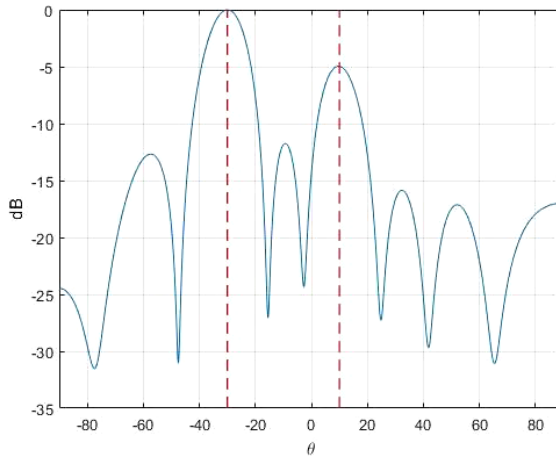


Fig. 13. The beam-pattern minimum variance.

to the line by computing the amount of error accumulated from the previous line position to the actual position. It recalls the direction of avoidance and moves in the opposite direction to reach the trajectory (route) while leaving the obstacle behind it. Fig. 11 shows that the autonomous vehicle comes back to the trajectory after a couple of seconds. We have mounted an array of 9 acoustic sensors in front of the AV, as shown in Fig. 6. These sensors are arranged like shown in Fig. 14 to perceive incident signals from obstacles that are along the path.

The ultrasound wavefront frequency is 40 kHz; it results from (20) that it is in a relation with the sensitivity of the acoustic sensor array for signals coming from a particular direction. We

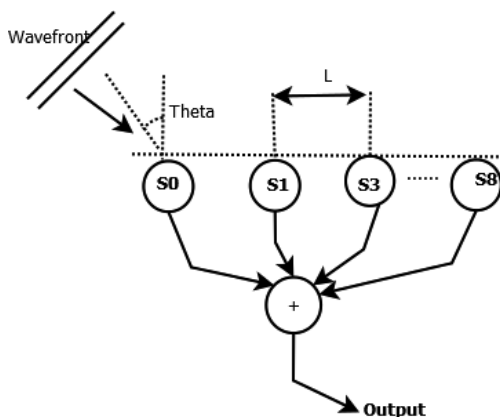


Fig. 14. The angle of arrival is measured from the line perpendicular to the array.

compute the sensitivity of our sensor array as shown in Fig. 14. The array has 9 acoustic sensors separated by a distance l (1 cm). The angle of arrival:

$$Output = 20 \log_{10} \left(\frac{1}{N} \sum_{i=0}^{N-1} e^{\frac{j2\pi f i L \sin \theta}{c}} \right) \quad (24)$$

is measured from the line perpendicular to the array, where *output* is the array's gain for a single frequency f (40 kHz) and an arrival angle θ , c denotes the speed of sound and N is the number (9) of acoustic sensors in the array. Fig. 8 shows the directional pattern of the sensor array sensitivity estimated with MVDR. Fig. 9 also shows the beampattern of the embedded delay-and-sum beamformer where the arrival angle is either -30 or 10 degrees. Fig. 9 shows the embedded beam-former dynamic range at 6.0 dB, and Fig. 10 presents the dynamic range of the embedded beam-former at 6.0 dB. Fig. 13 illustrates the beampattern of delay-and-sum and minimum variance when steered to -10 degrees. The input signal consists of two sources arriving at incidence angles of -30 , and 10 degrees.

5. Conclusions

The PID controller is a mathematically-based routine that processes sensor data and uses them to control the direction (and/ or speed) of an AV to keep it on route. In this paper we use it for sensing a trajectory and avoiding obstacles on the path to enable maneuvering the AV to stay on route, while constantly correcting wrong moves using a feedback mechanism. An additional application could be tailored not only for roads but also for working in harsh and severe environments [19, 20]. The more granular the sensor data are, the more accurately we can measure AV positions over the line. The embedded beamformer is implemented on the line follower AV to enable obstacle avoiding navigation while the AV is following the line. The beamformer is an efficient technique in which an array of sensors is employed to achieve the maximum reception in a specified direction by accepting a signal arriving from a desired direction and rejecting signals of the same frequency coming from other directions; however, some parts of the beamformer can be improved [21]. In this paper we have two input signals arriving at -30 , and 10 degrees respectively, with the array being steered to the incidence angle of the first source, so that the array will

have its mainlobe pointed in this direction. The beampattern of the delay-and-sum beamformer shows how the obtained signal will be distorted by signals arriving at an incidence angle that corresponds to the location of one of the side lobes of the array. The beampattern is forced to have minimum energy at arriving angles corresponding to other sources [22]. The proposed application fits the new trend of using public transportation vehicles without drivers [23], as it has already been done in some cities.

References

- [1] Liu, Y., Peng, Y., Wang, B., Yao, S., Liu, Z. (2017). Review on Cyber-physical System. *IEEE/CAA Journal of Automatica Sinica*, 4(1), 27–40.
- [2] Xu G., Tan, M. (2001). Development Status and Trend of Mobile Robot. *Robot Technique and Application*, 3(5), 7–13.
- [3] Xin, X., Ye, H., Feng, C. (1990). The PID Adaptive Control of Operator. *Robots*, 12(2), 1–7.
- [4] Ying, D., Songshu, S. (2002). Global Stability of the PD + Feedforward Robot Robust Adaptive control. *ACTA Automatic Sinica*, 8(1), 11–18.
- [5] Lay-Ekuakille, A., Vendramin, G., Trotta, A. (2008). Beamforming-Based Acoustic Imaging for Distance Retrieval. *IMTC 2008 – IEEE International Instrumentation and Measurement Technology Conference*, Victoria, Vancouver Island, Canada.
- [6] Jafarov, E.M., Parlakci M.N.A., Istefanopulos, Y. (2005). A new variable structure PID-controller design for robot manipulators. *IEEE Transactions on Control Systems Technology*, 13(1), 122–130.
- [7] Ramasamy, S., Pradhan, H.V., Ramanathan, P., Arulmozhivarman, P., Tatavarti, R. (2012). A novel and pedagogical approach to teach PID controller with LabVIEW signal express. *IEEE International Conference on Engineering Education: Innovative Practices and Future Trends (AICERA)*, Kerala, India.
- [8] Amer, N.H., Hudha, K., Zamzuri, H., Aparow, V.R., Abidin, A.F.Z., Kadir, Z.A., Murrad, M. (2018). Adaptive modified Stanley controller with fuzzy supervisory system for trajectory tracking of an autonomous armoured vehicle. *Robotics and Autonomous Systems*, 105, 94–111.
- [9] Malikopoulos, A.A., Cassandras, C.G., Zhang, Y.J. (2018). A decentralized energy-optimal control framework for connected automated vehicles at signal-free intersections. *Automatica*, 93, 244–256.
- [10] Liu, J., Jayakumar, P., Stein, J.L., Ersal, T. (2017) A nonlinear model predictive control formulation for obstacle avoidance in high-speed autonomous ground vehicles in unstructured environments. *Vehicle System Dynamics*, 56(6), 853–882.
- [11] Viegas, D., Batista, P., Oliveira, P., Silvestre, C. (2018). Discrete-time distributed Kalman filter design for formations of autonomous vehicles. *Control Engineering Practice*, 75, 55–68.
- [12] Tibljaš, A.D., Giuffrè, T., Surdonja, S., Trubia, S. (2018). Introduction of Autonomous Vehicles: Roundabouts design and safety performance evaluation. *Sustainability*, 10(4), 1–14.
- [13] Lay-Ekuakille, A., Vergallo, P., Saracino, D., Trotta, A. (2012). Optimizing and post processing of a smart Beamformer for obstacle retrieval. *IEEE Sensors Journal*, 12(5), 1294–1299.
- [14] Grythe, J. (2010). *Beamforming algorithms – beamformers*. Norsonic AS, Oslo, Norway.
- [15] Johnson, D.H., Dudgeon, D.E. (1993). *Array signal processing: concepts and techniques*. P T R Prentice Hall.
- [16] Um, D., Sriraman, V. (2004). *Teaching basic control systems theory using robots. Proceedings of the 2004 American Society for Engineering Education Annual Conference & Exposition*. Oregon, USA.

- [17] Van Trees, H.L. (2002). *Detection, Estimation, and Modulation Theory, Optimum Array Processing*. Part IV edition Ed. Wiley-Interscience. New York.
- [18] Strakowski, M.R., Kosmowski, B.B., Kowalik, R. (2006). An Ultrasonic Obstacle Detector Based on Phase Beamforming Principles. *IEEE Sensors Journal*, 6(1), 179–186.
- [19] Lay-Ekuakille, A., Palamara, I., Caratelli, D., Morabito, F.C. (2013). Experimental Infrared Measurements for Hydrocarbon Pollutant Determination in Subterranean Waters. *Review of Scientific Instruments*, 84, 015103–1.
- [20] Lay-Ekuakille, A., Trotta, A. (2011). Predicting VOC Concentration Measurements: Cognitive Approach for Sensor Networks. *IEEE Sensors Journal*, 11(11), 3923–3030.
- [21] Karthik, G.V.S., Fathima, Y., Zia Ur Rahman, M., Ahamed, R. (2013). Efficient Signal Conditioning techniques for Brain activity in Remote Health Monitoring Network. *IEEE Sensors Journal*, 13(9), 3276–3283.
- [22] Chahbi, I., Ben Amara, D., Belghith, A. (2013). A novel route guidance algorithm using beamforming techniques for vehicular networks. *Proceedings – Conference on Local Computer Networks, LCN2013*, Sydney, Australia.
- [23] https://www.challenges.fr/france/lyon-experimente-les-bus-autonomes-sans-chauffeur_414889