

AS-84.3127 Paikannus- ja navigointimenetelmät

Ratkaisut 4.

1.

Robotin ja maamerkkien koordinaatit olivat

$$x_r = 2, y_r = 4, x_1 = 5, y_1 = 7, x_2 = 20 \text{ ja } y_2 = -15.$$

Lisäksi robotin suuntakulma oli $\phi = 30^\circ$.

a) Etäisyys maamerkkeihin saadaan kaavasta

$$r_i = \sqrt{(x_i - x_r)^2 + (y_i - y_r)^2}$$

ja kulma saadaan kaavasta

$$\theta_i = \tan^{-1} \left(\frac{(y_i - y_r)}{(x_i - x_r)} \right) - \phi$$

Sijoitetaan näihin kaavoihin lukuarvot ja saadaan

$$r_1 = \sqrt{(5-2)^2 + (7-4)^2} = 3\sqrt{2}$$

$$r_2 = \sqrt{(20-2)^2 + (-15-4)^2} = 26.17$$

$$\theta_1 = \tan^{-1} \left(\frac{(7-4)}{(5-2)} \right) - 30^\circ = 15^\circ$$

$$\theta_2 = \tan^{-1} \left(\frac{(-15-4)}{(20-2)} \right) - 30^\circ = -76.55^\circ$$

On huomattava että oikeissa mittauksissa on epätarkkuuksia, jotka aiheuttavat mm. sen, että mitatut kulma-arvot voivat olla vain jonkun askeleen monikertoja. Tämä tarkoittaa sitä, että mittauksessa oikea kulma-arvo "pyöristyy" läheiseen monikertaan. Tällöin mittauspisteen aito paikka sijaitsee kaarevalla banaanimuotoisella alueella, mitattua kulma-arvoa ja etäisyyttä vastaavan paikan ympäristössä. "Banaanin" paksuus riippuu etäisyysmittaukseen liittyvästä kohinasta.

b)

Yllä johdettiin mittausyhtälö

$$h(x_r, y_r, \phi_r) = \begin{bmatrix} r \\ \theta \end{bmatrix} = \begin{bmatrix} \sqrt{(x_i - x_r)^2 + (y_i - y_r)^2} \\ \tan^{-1}\left(\frac{y_i - y_r}{x_i - x_r}\right) - \phi \end{bmatrix} \quad (1)$$

tässä tehtävässä on tarkoitus selvittää miten pieni muutos robotin paikassa vaikuttaa mittaukseen. Tätä varten approksimoidaan mittausfunktiota ensimmäisen asteen Taylorin sarjalla, eli

$$h(s) \approx h(s_0) + \frac{dh}{ds}(s - s_0)$$

missä s on vektori $[x_r, y_r, \phi_r]^T$ ja

$$\frac{dh}{ds} = H = \begin{bmatrix} \frac{\partial r}{\partial x} & \frac{\partial r}{\partial y} & \frac{\partial r}{\partial \phi} \\ \frac{\partial \theta}{\partial x} & \frac{\partial \theta}{\partial y} & \frac{\partial \theta}{\partial \phi} \end{bmatrix} = \begin{bmatrix} \frac{dx}{\sqrt{dx^2 + dy^2}} & \frac{dy}{\sqrt{dx^2 + dy^2}} & 0 \\ \frac{dy}{dx^2 + dy^2} & \frac{-dx}{dx^2 + dy^2} & -1 \end{bmatrix} \quad (2)$$

2.

ICP-algoritmista löytyy lukuisia versioita, jotka ovat räätälöityjä tiettyyn sovellukseen. Alla on yksinkertaistettu matlab toteutus ICP-algoritmista. Sen vaiheet ovat:

1. virtuaalisten mittausten muodostaminen viimeisimmillä parametriarvoilla. Katso kaava 1. edellisessä tehtävässä
2. Kullekin karttapisteelle etsitään lähin naapuri mittauspisteistä. Mikäli lähin naapuri löytyy tarpeeksi läheltä, se liitetään niiden pisteparien joukkoon joita käytetään siirtymän laskennassa.
3. Suoritetaan yksi askel levenberg-marquard optimointialgoritmia (tähän tarvitaan Jacobiaania ks. kaava 2. edellisessä tehtävässä.). Saadaan uudet parametriarvot.
4. Jos kierroksia ei ole tarpeeksi(tai joku muu lopetuskriteeri täyttyy), siirrytään kohtaan 1., muuten lopetetaan

```
function t=doICP(t,r,a,ix,iy);
```

```

%t = parametrivektori
%r = vektori mittauspisteiden etäisyyksiä
%a = vektori mittauspisteiden kulmia
%ix = vektori karttapisteiden x arvoja
%iy = vektori karttapisteiden y arvoja

numMapPoints=length(ix);
numMeasPoints=length(r);
Threshold=1;
cyc=5;
i=0;

while(i<cyc)
    %1.

    %t(1)=dx,t(2)=dy,t(3)=dphi
    dx= repmat(t(1:2),1,numMapPoints)-[ix;iy];
    num=dx'*dx;
    range=sqrt(num);
    theta=atan2(dx(2,:),dx(1,:))-t(3);

    %2.
    H=ones(1,3)*0.1;
    z=zeros(1);
    for j=1:numMapPoints
        dr=repmat(range(j),numMeasPoints,1)-r;
        da=repmat(theta(j),numMeasPoints,1)-a;
        [minValue,index]=min((dr/1).^2+(dy/20).^2);
        if(minValue<Threshold)
            dx=x(1:2)-xi;
            num=dx'*dx;
            r=sqrt(num);
            Hi=[dx(1,j)/range(j) dx(2,j)/range(j) 0
                -dx(2,j)/num(j) dx(1,j)/num(j) -1];

            H=[H;Hi];
            z=[z;dr(index);dtheta(index)];
        end
    end
    %3.
    t=t+0.7*(H\z);
    %4.
    i=i+1;
end

```

3.

Robotin paikan määrittäminen sovittamalla mittauksia (esim laserpyyhkäisyä) kartaan on globaali optimointiongelma, jossa optimoitavat parametrit ovat robotin paikkakoordinaatteja ja suuntakulmia. Usein oletetaan että robotti liikkuu tasossa, joten tarvittavat parametrit ovat kaksi paikkakoordinaattia ja suuntakulma. "Brute-force" menetelmä eli tasahaku diskretoisi hakualueen ja sovittaisi karttaa dataan jokaisella diskretoitujen parametrien kombinaatiolla. Tämän menetelmän ongelmana on se että laskentatyön määrä kasvaa hyvin nopeasti tarvittavien parametrien määrän mukana. Tämän ilmiön vaikutus tulee esiin jo kolmen parametrin tapauksessa. Jos esimerkiksi paikkakoordinaateilla ja suuntakulmalla olisi kullakin 1000 diskreettiä arvoa niin tasahaussa täytyisi tutkia $1000 \cdot 1000 \cdot 1000 = 10^9$ parametrikombinaatiota.

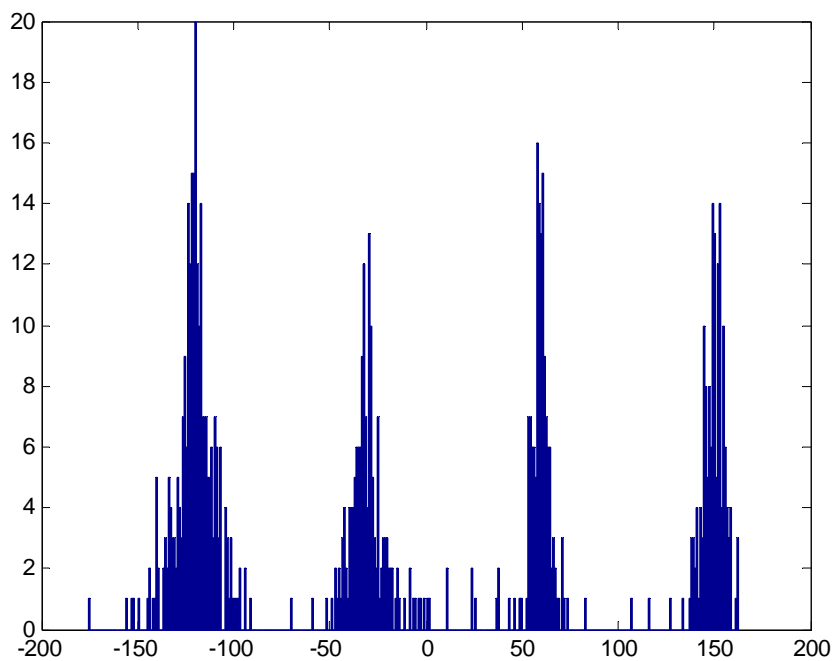
Tässä tehtävässä käsitelty menetelmä pyrkii löytämään kunkin parametrin erillisellä haulla. Tällöin esimerkkitapauksessa täytyisi tutkia $1000 + 1000 + 1000 = 3000$ parametrikombinaatiota. Menetelmän ideana on käyttää kunkin parametrin sovituksessa sellaisia piirteitä, jotka ovat robusteja muiden parametrien arvoista. Tällainen piirre on peräkkäisten etäisyysmittausten osumapisteiden välisen vektorin kulma. Jos laserpyyhkäisyn peräkkäisten mittausten osumapisteiden vektorin kulma-arvoista muodostetaan histogrammi, nähdään siinä kunkin suuntaisia tasopintoja vastaavia pylväitä. Kun kahden, eri paikasta lasketun kulmahistogrammin pylväät asetetaan päällekkäin saadaan kulmahistogrammien välinen suuntakulmaero. Pylväiden korkeus, eli tasopinnalle osuneiden mittausten määrä, riippuu tietenkin robotin paikasta. Käytännössä mittauskohinan takia kulmien laskeminen on epätarkkaa, joten pylväät leviävät kasoiksi. Kohinan vaikutusta voidaan vähentää laskemalla kulmahistogrammit peräkkäisten mittausten sijasta joka n:en välisistä mittauksista.

Robotin suuntakulma on sen x-akselista vastapäivään.

a)

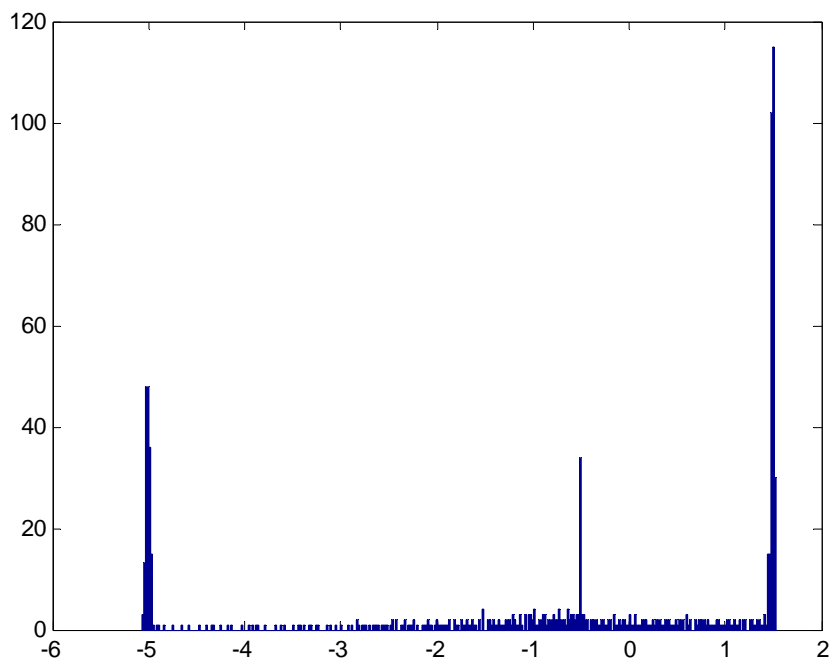
Alla olevassa kuvassa on tehtävän kuvaa vastaava kulmahistogrammi. Kulmat on laskettu joka seitsemännen mittauspisteen välisistä vektoreista. Simuloitu laserskanneri antoi kulma-arvot puolen asteen välein ja etäisyysmittaukseen lisättiin kohinaa jonka varianssi oli 4 cm^2 .

Skanneri näkee ensiksi ylhäällä olevan seinän, josta tulee piikki kulmahistogrammiin kohdassa -120 astetta. Sen jälkeen skanneri näkee oikealla olevan seinän, josta tulee piikki kohtaan 150 astetta. Alhaalla oleva seinä on kohdassa 60 astetta ja vasemmalla oleva seinä kohdassa -30 astetta.



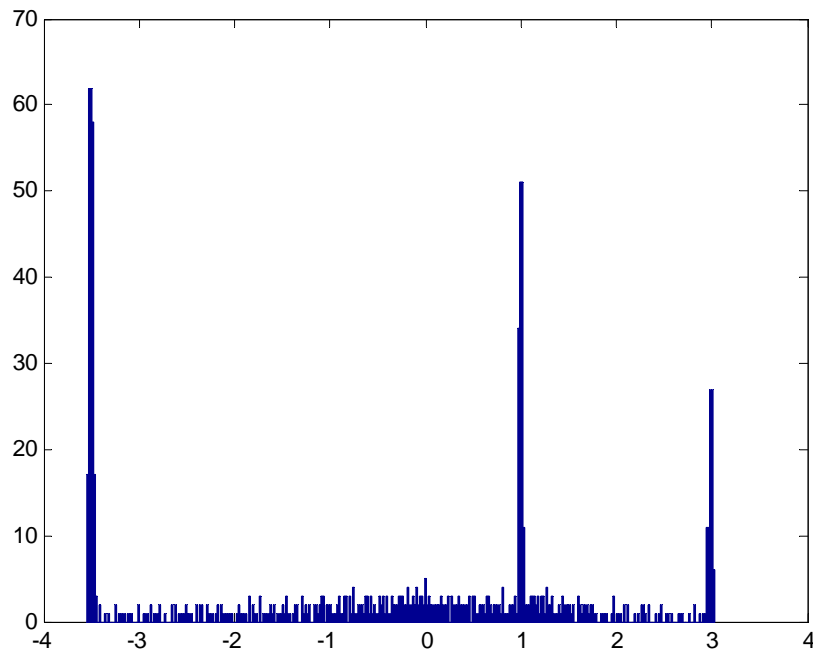
Kuva1: Tehtävän kuvaa vastaava kulmahistogrammi

b) Alhaalla oleva seinä näkyy piikkinä kohdassa -5 ja ylhäällä oleva seinä kohdassa $1,5$. Huoneessa olevan laatikon yläpinta näkyy kohdassa $-0,5$.



Kuva2: Tehtävän kuvaa vastaava x-suuntaisten etäisyyksien histogrammi

Kuvassa näkyy oikeanpuoleinen seinä piikkinä kohdassa $-3,5$ ja vasemman puoleinen seinä kohdassa 3 . Huoneessa oleva laatikon sivuseinä näkyy kohdassa 1 .



Kuva3: Tehtävän kuvaa vastaava y-suuntaisten etäisyyksien histogrammi

AS-84.3127 Localization and navigation methods

Solutions for exercise 4.

1.

The coordinates of the robot and the landmarks were

$$x_r = 2, y_r = 4, x_1 = 5, y_1 = 7, x_2 = 20 \text{ ja } y_2 = -15.$$

Further, the observation angle of the robot was $\phi = 30^\circ$.

a) The distance to the landmarks is given by the equation.

$$r_i = \sqrt{(x_i - x_r)^2 + (y_i - y_r)^2}$$

and we get the angle from

$$\theta_i = \tan^{-1} \left(\frac{(y_i - y_r)}{(x_i - x_r)} \right) - \phi$$

By assigning the numerical values to these equations we get

$$r_1 = \sqrt{(5-2)^2 + (7-4)^2} = 3\sqrt{2}$$

$$r_2 = \sqrt{(20-2)^2 + (-15-4)^2} = 26.17$$

$$\theta_1 = \tan^{-1} \left(\frac{(7-4)}{(5-2)} \right) - 30^\circ = 15^\circ$$

$$\theta_2 = \tan^{-1} \left(\frac{(-15-4)}{(20-2)} \right) - 30^\circ = -76.55^\circ$$

It should be noted that real measurements contain inaccuracies, which cause e.g. that the measured angles can be multiples of an angle step. This means that the correct angle is rounded up or down to the closest multiply of the angle step. In this case the actual (and unknown) location of the observer (i.e. robot) is situated on an area that has the form of a banana, in the neighborhood of the location corresponding to the measured angle and distance. The thickness of the banana is dependent on the size of the measurement noise.

b)

The derived measurement equation is

$$h(x_r, y_r, \phi_r) = \begin{bmatrix} r \\ \theta \end{bmatrix} = \begin{bmatrix} \sqrt{(x_i - x_r)^2 + (y_i - y_r)^2} \\ \tan^{-1}\left(\frac{(y_i - y_r)}{(x_i - x_r)}\right) - \phi \end{bmatrix} \quad (1)$$

The problem addressed here is to find out how a small change in the robot location affects the measured value. For this purpose we approximate the measurement equation with a first order Taylor series expansion, which is

$$h(s) \approx h(s_0) + \frac{dh}{ds}(s - s_0)$$

where s is the vector $[x_r, y_r, \phi_r]^T$ and

$$\frac{dh}{ds} = H = \begin{bmatrix} \frac{\partial r}{\partial x} & \frac{\partial r}{\partial y} & \frac{\partial r}{\partial \phi} \\ \frac{\partial \theta}{\partial x} & \frac{\partial \theta}{\partial y} & \frac{\partial \theta}{\partial \phi} \end{bmatrix} = \begin{bmatrix} \frac{dx}{\sqrt{dx^2 + dy^2}} & \frac{dy}{\sqrt{dx^2 + dy^2}} & 0 \\ \frac{dy}{dx^2 + dy^2} & \frac{-dx}{dx^2 + dy^2} & -1 \end{bmatrix} \quad (2)$$

where we have used

$$dx = x_i - x_r$$

$$dy = y_i - y_r$$

$$\frac{\partial(\tan^{-1}(x))}{\partial x} = \frac{1}{1 + x^2}$$

2.

There are several versions of the ICP-algorithm, which have been tailored for a particular application. Below a simplified realization of the ICP-algorithm is presented. It consists of the following phases:

5. Formation of virtual measurements by using the latest parameter values. Refer to equation 1. of the previous problem.

6. For each target position on the map the closest measurement position is searched for. In case the closest position is close enough it will be used later to compute the sensor location update increment.
7. Carry out an update step by using the Levenberg-Marquard optimization algorithm. For this purpose the Jacobian (equation 2 of the previous problem) is required.
8. If the algorithm has not been repeated enough times or if not some other finishing criteria has been filled go back to phase 1 otherwise end the ICP-algorithm

```
function t=doICP(t,r,a,ix,iy);

%t = parameter vector
%r = vector of distances to the map positions
%a = vector of angles to the map positions
%ix = vector of x coordinates of the map positions
%iy = vector of y coordinates of the map positions

numMapPoints=length(ix);
numMeasPoints=length(r);
Threshold=1;
cyc=5;
i=0;

while(i<cyc)
    %1.

    %t(1)=dx,t(2)=dy,t(3)=dphi
    dx= repmat(t(1:2),1,numMapPoints)-[ix;iy];
    num=dx'*dx;
    range=sqrt(num);
    theta=atan2(dx(2,:),dx(1,:))-t(3);

    %2.
    H=ones(1,3)*0.1;
    z=zeros(1);
    for j=1:numMapPoints
        dr=repmat(range(j),numMeasPoints,1)-r;
        da=repmat(theta(j),numMeasPoints,1)-a;
        [minValue,index]=min((dr/1).^2+(dy/20).^2);
        if(minValue<Threshold)
            dx=x(1:2)-xi;
            num=dx'*dx;
            r=sqrt(num);
            Hi=[dx(1,j)/range(j) dx(2,j)/range(j) 0
                -dx(2,j)/num(j) dx(1,j)/num(j) -1]
```

```

                H=[H;Hi];
                z=[z;dr(index);da(index)];
            end
        end
    %3.
    t=t+0.7*(H\z);
    %4.
    i=i+1;
end

```

3.

Determining the location of the robot by matching measurements (e.g. range scans) to the map is a global optimization problem, where the parameters to be optimized are the position coordinates and the orientation angles of the robot. Often it is assumed that the robot is moving on a plane such that the robot location can be described with the x and y position coordinates and a direction angle. A “brute-force” method would discretize the location search space and would then fit the map to the data with each combination of the three location parameters within the search space. The method has the caveat that the computational load increases very fast as a function of the number of parameters to be searched for. The effects of the phenomenon are already clear already in case of three parameters. For example if each of the three location parameters would have 1000 discrete values the total search space would contain $1000 \cdot 1000 \cdot 1000 = 10^9$ parameter combinations.

The method applied here to solve the problem tries to find the correct value for each of the parameters with a separate search. In this case for our example problem $1000 + 1000 + 1000 = 3000$ parameter combinations have to be checked to find the best match. The idea behind the method is to use such features that are robust with respect to other parameters to find the value for a single parameter. An example of robust features is the orientation angle of the vector drawn between two consecutive range measurements hit points. When a histogram is formed from the orientation angles by considering a whole range scan, in the histogram, peaks corresponding to the planar (wall) surfaces appear. When two angle histograms, corresponding to two different locations in the navigation space, are overlapped the orientation difference of the two locations is the same as the difference of the locations of the corresponding peaks in the two histograms. The height of the peaks, in other words the count of laser beam hit points at a particular planar surface is naturally dependent on the location of the robot within the navigation

space. In practice, due to sensor noise the computation of the angles to form the histogram is inaccurate, therefore the peaks in the histogram spread over a larger angular interval. The effect of the noise in the range measurements can be decreased by computing the angle values between every n^{th} measurement point.

The robot heading angle is defined as anticlockwise from robot x-axis.

a)

In the figure below the angle histogram, corresponding to the example problem, is shown. The angles have been computed from the direction vectors drawn between every 7th measurement point. The simulated laser scanner gave the angles in half degree increments and noise, with standard deviation of 4 cm, was added to the range measurements.

The scanner sees first the wall on the upside and the respective peak in angle the histogram is at -120 degrees. Next the scanner sees the wall on right side and the peak is at 150 degrees. The downside wall is seen at 60 degrees and the wall at left side is seen at -30 degrees.

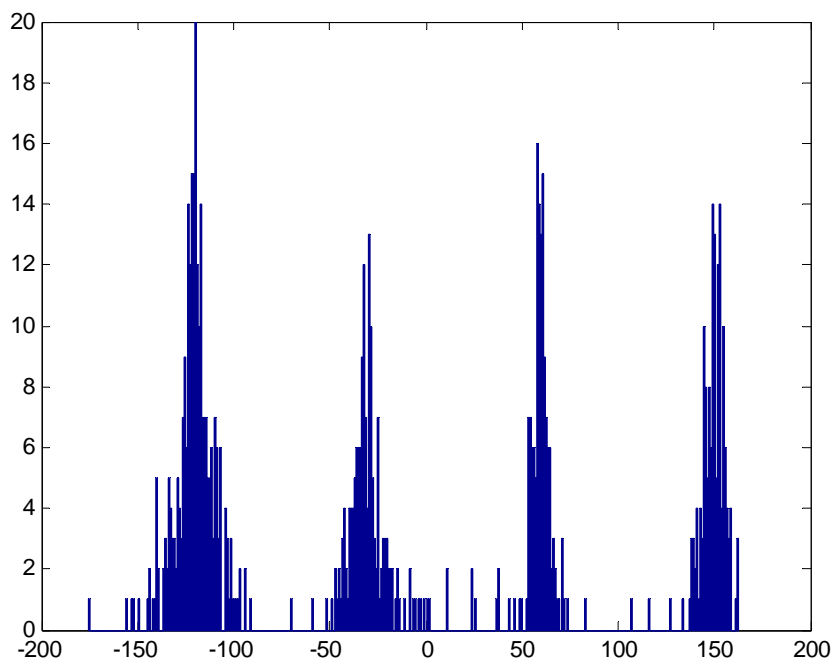


Figure 1: Angle histogram corresponding to the figure in the problem description.

b) The down side wall is seen as peak at -5 and the upside wall is seen at $1,5$. The upside of the box in the room is seen at $-0,5$.

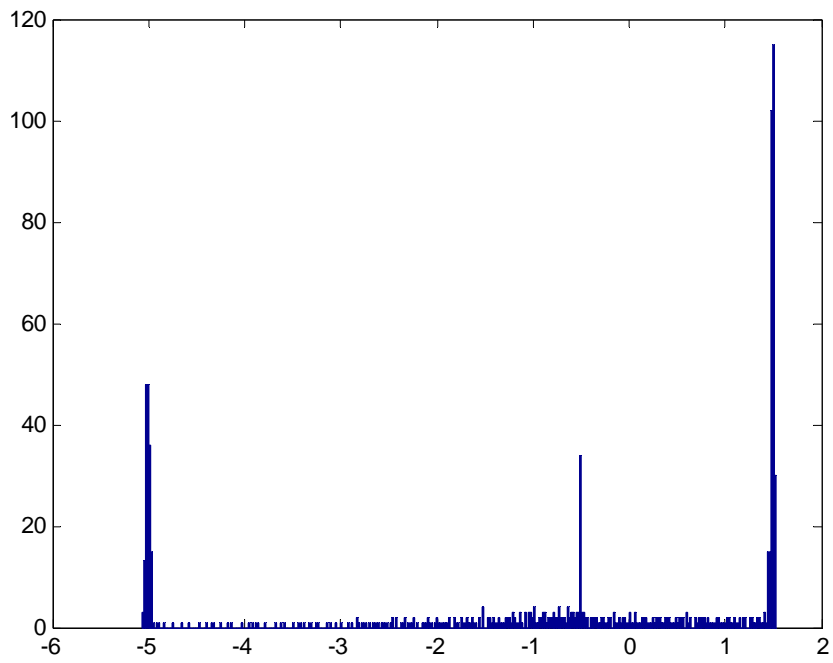


Figure 2: Histogram of measurement point for the x-coordinate direction.

The right side wall is seen as peak at $-3,5$ and the wall on the left side is seen at 3 . The side of the box in the room is seen at 1 .

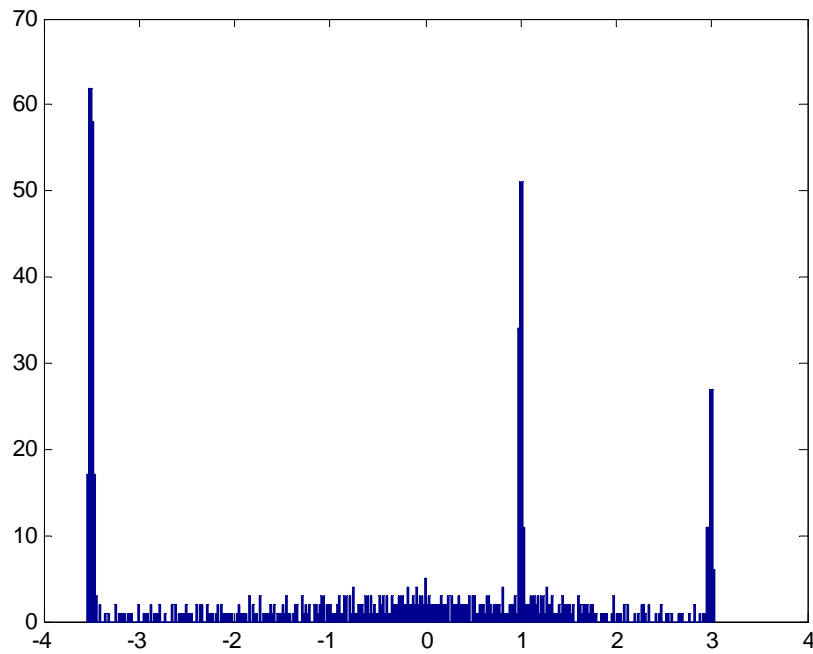


Figure 3: Histogram of measurement point for the y-coordinate direction.