Міністерство освіти й науки України

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Звіт з лабораторної роботи №2

з дисципліни «Апаратно-програмні засоби збору та обробки екологічної інформації»

«**Визначення частоти опитування технологічних параметрів за допомогою керуючої обчислювальної машини**»

Варіант 21

Виконав:

Студент 1-го курсу

групи ТР-21мп НН ІАТЕ

Сергєєв Данило Вікторович

Перевірила: Круш О. Є.

Київ-2023

**1. Мета роботи:**

Освоїти методику розрахунку частоти опитування технологічних параметрів за допомогою керуючої обчислювальної машини (КОМ). Визначити періоди опитування датчиків технологічних параметрів об'єктів зі значно відмінними статистичними характеристиками. Дослідити вплив на величину періоду опитування статистичних властивостей вимірюваної величини, похибки її визначення і похибки вимірювального тракту.

**Теоретичні відомості**

Питання вибору необхідної частоти опитування технологічних параметрів за допомогою КОМ виникає при створенні автоматизованої системи управління технологічним процесом (АСУ ТП) на етапі розробки технічного завдання (ТЗ).

***Завищена частота опитування*** призводить до ускладнення системи дискретного контролю та підвищенню завантаження обчислювальної частини КОМ.

***Занижена частота опитування*** практично може звести до нуля результати дискретного контролю, оскільки при цьому неможливо простежити з необхідною точністю за зміною контрольованої величини.

Визначимо конкретний тип рівняння, що використовується для визначення інтервалу опитування в практично найбільш простому і поширеному випадку ступінчатої екстраполяції, при котрому про значення вимірюваної величини у будь-який момент часу судять за виміряним значенням величини в момент попереднього відліку (див. рис. 2.1).

, (2.1)

при

де *To -* період квантування (опитування) процесу *X(t)* за часом

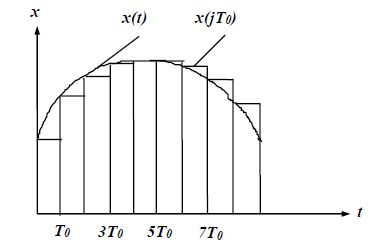


Рисунок 2.1 – Ступінчаста екстраполяція вимірюваної величини

При екстраполяції виникає похибка *ε(t)* оцінки поточного значення вимірюваної величини, що визначається за формулою:

(2.2)

Похибка *ε(t)* є випадковим процесом, особливістю якого є те, що він дорівнює нулю в момент початку кожного періоду квантування (див. рис. 2.1).

Оцінимо середню квадратичну похибку екстраполяції методом (2.1) для випадку, коли вимірюваний процес є стаціонарним.

Трансформуємо рівняння (2.2), додавши в праву частину і віднявши від неї величину математичного очікування процесу *Мx:*

(2.3)

де *X(t)* та *X(jTo)* - відповідні центровані процеси

Зведемо ліву і праву частини виразу (2.3) в квадрат і будемо розглядатиповедінку функції *ε2(t)* на окремих інтервалах часу або, якщо ввести нову змінну τ = t − jT0, то на інтервалі 0≤τ≤T0.

Усереднюючи по множині інтервалів *j*, отримаємо

(2.4)

при 0 ≤ τ ≤ T0

Використовуючи поняття кореляційної функції та дисперсії, в силу стаціонарності випадкового процесу *X(t)* матимемо

(2.5)

де *Kx* і *Dx* - кореляційна функція і дисперсія випадкового процесу.

З урахуванням (2.5) вираз (2.4) буде записано у вигляді

(2.6)

де (𝜏) - середня квадратична похибка визначення величини.

Вираз (2.6) не враховує похибку вимірювального тракту (похибка датчика) *σХИЗМ*. З її врахуванням вираз для (𝜏) матиме вигляд

(𝜏) = 2[𝐷𝑋 − 𝐾𝑋(𝜏)] + (2.7)

З (2.7) слідує, що середня квадратична похибка (𝜏) є функцією часу 𝜏, тобто змінюється у межах періоду квантування. У моменти замірів, тобто при t=jT0 (𝜏 = 0) ця похибка є мінімальною, рівною похибці вимірювального тракту. Максимального значення середня квадратична похибка набуває при екстраполяції в момент, що передує наступному замірі величини Х(t) при τ = T0.

(2.8)

Частота опитування визначається величиною

При τ = T0 рівняння(2.7) набуває вигляду

(2.9)

Формула (2.9) визначає період *Т0* опитування датчика величини *X(t)* за заданою похибкою її визначення , відомої похибки датчика та відомої кореляційної функції процесу.

Варіант завдання – 7.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Вхідні дані | Варіант | 1 | 2 | 3 | 4 | 5 | 6 | **7** | 8 | 9 | 10 |
| Тепловий об'єкт |  | **4,0** | 4,5 | 3,6 | 3,8 | 4,2 | 5,0 | 4,9 | 4,6 | 4,7 | 3,7 |
|  | **1,0** | 0,6 | 0,8 | 0,7 | 0,9 | 1,2 | 1,1 | 0,5 | 1,3 | 1,4 |
| Об'єкт контролю тиску |  | **3,6** | 2,8 | 3,9 | 2,7 | 3,1 | 3,5 | 3,4 | 2,6 | 3,7 | 2,9 |
|  | **0,8** | 0,5 | 1,0 | 0,3 | 0,6 | 0,7 | 0,9 | 0,4 | 0,2 | 1,2 |

Лістинг:

public static class CalculationFunctions  
{  
 *// Метод CalcKxTa розраховує величину Кх(Та) за формулою (2.9).* public static double CalcKxTa(double dx, double qh, double qx)  
 {  
 return (2 \* dx + Math.Pow(qh, 2) - Math.Pow(qx, 2)) / 2;  
 }  
  
 *// Метод GenerateByValue генерує масив значень за заданою формулою.  
 // Параметр size вказує розмір масиву, delimiter - крок між значеннями.* public static double[] GenerateByValue(double value, int size = 5, double delimiter = 10)  
 {  
 double[] arr = new double[size];  
 for (int i = 0; i < arr.Length; i++)  
 {  
 if (i % 2 == 0)  
 {  
 arr[i] = value + i / delimiter;  
 }  
 else  
 {  
 arr[i] = value - i / delimiter;  
 }  
 }  
 Array.Sort(arr);  
 return arr;  
 }  
}

Task1:  
public static void Run()  
{  
 *// Задаємо початкові значення для розрахунків.* const double **qh** = 1.0;  
 const double **qxmax** = 4.0;  
 const double **t0Temp** = 2;  
 const double **dtemp** = 44;  
 const double **t0Press** = 0.1;  
 const double **dpress** = 24;  
  
 double[] qhValues = GenerateByValue(**qh**, 5);  
 double[] qxmaxValues = GenerateByValue(**qxmax**, 5);  
  
 *// Розраховуємо величину Кх(То) за формулою (2.9).* double kxTemp = CalcKxTa(**dtemp**, **qh**, **qxmax**);  
  
 *// Генеруємо масив дельта Т* double[] deltaT = GenerateByValue(**t0Temp**, 5);  
 *// Розраховуємо масив значень Ктх(Т, ΔQх)* double[] ktxTempTDeltaQh = qhValues.Select(x => CalcKxTa(**dtemp**, x, **qxmax**)).ToArray();  
 *// Розраховуємо масив значень Ктх(Т, ΔQmax)* double[] ktxTempTDeltaQMax = qxmaxValues.Select(x => CalcKxTa(**dtemp**, **qh**, x)).ToArray();  
  
 *// Розраховуємо величину Кх(То) за формулою (2.9) для тиску.* double kxPress = CalcKxTa(**dpress**, **qh**, **qxmax**);  
  
 *// Генеруємо масив дельта Р* double[] pDeltaP = GenerateByValue(**t0Press**, 5, 50);  
 *// Розраховуємо масив значень Ктх(Р, ΔQх)* double[] ktxPressTDeltaQh = qhValues.Select(x => CalcKxTa(**dpress**, x, **qxmax**)).ToArray();  
 *// Розраховуємо масив значень Ктх(Р, ΔQmax)* double[] ktxPressTDeltaQMax = qxmaxValues.Select(x => CalcKxTa(**dpress**, **qh**, x)).ToArray();  
  
 Console.WriteLine("1. Визначення періоду опитування датчиків з реалізації випадкових процесів за температурою і тиском");  
 Console.WriteLine("a) Величина Kx(Ta) при заданих QxMax та Qh:");  
 Console.WriteLine($"Kx(Ta) = {kxTemp:F2}");  
 Console.WriteLine();  
 Console.WriteLine("б) За графіком кореляційних функцій, визначаємо значення періодів опитування T0 датчиків температури і тиску:");  
 Console.WriteLine($"T0 для датчика температури: {**t0Temp**:F2}");  
 Console.WriteLine($"T0 для датчика тиску: {**t0Press**:F2}");  
 Console.WriteLine();  
 Console.WriteLine("в) Для постійної величини QxMax визначаємо декілька значень T0 при різних значеннях Qh для датчика температури:");  
 Console.WriteLine($"Kx(T0) = {kxTemp:F2}");  
 Console.WriteLine("QxMax = const");  
 for (int i = 0; i < deltaT.Length; i++)  
 {  
 Console.WriteLine($"K(T - {i}) = {ktxTempTDeltaQh[i]:F4}, T = {deltaT[i]:F2}");  
 }  
 Console.WriteLine();  
 Console.WriteLine("г) Для постійної величини Qh визначаємо декілька значень T0 при різних значеннях QxMax для датчика температури:");  
 Console.WriteLine("Qh = const");  
 for (int i = 0; i < deltaT.Length; i++)  
 {  
 Console.WriteLine($"K(T - {i}) = {ktxTempTDeltaQMax[i]:F4}, T = {deltaT[i]:F2}");  
 }  
 Console.WriteLine();  
 Console.WriteLine("Датчик тиску:");  
 Console.WriteLine("в) Для постійної величини QxMax визначаємо декілька значень T0 при різних значеннях Qh:");  
 Console.WriteLine($"Kx(T0) = {kxPress:F2}");  
 Console.WriteLine("QxMax = const");  
 for (int i = 0; i < pDeltaP.Length; i++)  
 {  
 Console.WriteLine($"K(T - {i}) = {ktxPressTDeltaQh[i]:F4}, T = {pDeltaP[i]:F2}");  
 }  
 Console.WriteLine();  
 Console.WriteLine("г) Для постійної величини Qh визначаємо декілька значень T0 при різних значеннях QxMax:");  
 Console.WriteLine("Qh = const");  
 for (int i = 0; i < pDeltaP.Length; i++)  
 {  
 Console.WriteLine($"K(T - {i}) = {ktxPressTDeltaQMax[i]:F4}, T = {pDeltaP[i]:F2}");  
 }  
 Console.WriteLine("д) Можна зробити висновок, що статичні властивості вимірюваної величини (похибки її визначення і похибки вимірювального тракту) впливають на величину Т0.");  
 Console.WriteLine("Зокрема, при збільшенні похибок вимірювання і похибок вимірювального тракту, значення Т0 збільшується.");  
 Console.WriteLine("Також можна зробити висновок, що збільшення значення Qh при постійному значенні QxMax призводить до зменшення значення Т0, а збільшення значення QxMax при постійному значенні Qh призводить до збільшення значення Т0.");  
 Console.WriteLine();  
}

Task2:

public static async Task Run()  
{  
 *// Задаємо початкові значення констант* const double **qh** = 1.0;  
 const double **qxmax** = 4.0;  
 *// Довжина реалізації випадкового процесу, мм* const double **l** = 100;  
 *// Швидкість руху діаграмного паперу самописця, мм/сек* const double **v** = 10;  
 *// Кількість перетинів випадковим процесом лінії свого математичного очікування* const double **n** = 100;  
 const int **size** = 10;  
 *// Час, протягом якого відбулося N перетинів* double tN = **l** / **v**;  
 *// Середнє число нулів за одиницю часу* double nMiddle = **n** / tN;  
 *// Крок дискретизації випадкового процесу* double deltaT = 0.15 / nMiddle;  
 *// Масив кроків дискретизації випадкового процесу* double[] deltaTArr = GenerateDeltaTArr(deltaT, **size**);  
 *// Масив випадкових цілих чисел* int[] arr = GenerateRandomIntArr(80, 140, **size**);  
 *// Математичне очікування* double mx = (double)arr.Sum() / arr.Length;  
 *// Дисперсія* double dx = CalcDx(arr, mx, **n**);  
 *// Масив значень потужності випадкового процесу* double[] deltaQh = GenerateByValue(**qh**, **size**);  
 *// Масив максимальних значень потужності випадкового процесу* double[] deltaQMax = GenerateByValue(**qxmax**, **size**);  
 *// Масив значень Kx* double[] kx = CalcKx(deltaQh, deltaQMax, dx);  
 *// Значення J0* double j0 = kx.Sum() / kx.Length;  
 *// Шуканий період опитування датчиків* double t0 = j0 \* deltaT;  
  
 Console.WriteLine("2. Визначення періоду опитування датчиків за кривими реалізації випадкового процесу:");  
 Console.WriteLine("а) Визначаємо крок дискретизації випадкового процесу:");  
 Console.WriteLine($"Час, протягом якого відбулося {**n**} перетинів: {tN:F2} мм/сек");  
 Console.WriteLine($"Середнє число нулів за одиницю часу: {nMiddle:F2}");  
 Console.WriteLine($"Крок дискретизації Δτ: {deltaT:F2}");  
 Console.WriteLine();  
 Console.WriteLine("б) Визначаємо статистичні характеристики випадкового процесу:");  
 Console.WriteLine($"Mx (Математичне очікування): {mx:F2}");  
 Console.WriteLine($"Dx (Дисперсія): {dx:F2}");  
 Console.WriteLine($"Kx (Кореляційна функція): [{string.Join(", ", kx.Select(x => x.ToString("F2")))}]");  
 Console.WriteLine();  
 Console.WriteLine("в) Визначаємо період опитування датчика:");  
 Console.WriteLine($"t0 = {t0:F2}");  
 Console.WriteLine($"(J0) = {j0:F2}");  
 Console.WriteLine($"deltaTArr (Масив кроків дискретизації випадкового процесу): [{string.Join(", ", deltaTArr.Select(x => x.ToString("F2")))}]");  
  
 Console.WriteLine("Отримати графік? (Масив значень Kx від кроків дискретизації випадкового процесу)");  
 if (string.IsNullOrEmpty(Console.ReadLine()))  
 {  
 var documentsPath = Environment.GetFolderPath(Environment.SpecialFolder.**MyDocuments**);  
 var filePath = Path.Combine(documentsPath, "chart.png");  
 DrawLineGraph(deltaTArr, kx, filePath);  
 await OpenImageFile(filePath);  
 }  
}  
  
*// Метод для генерації масиву випадкових цілих чисел*public static int[] GenerateRandomIntArr(int min, int max, int size = 13)  
{  
 var random = new Random();  
 return Enumerable.Range(0, size)  
 .Select(\_ => random.Next(min, max + 1))  
 .ToArray();  
}  
  
*// Метод для генерації масиву кроків дискретизації випадкового процесу*public static double[] GenerateDeltaTArr(double deltaT, int size = 13)  
{  
 double[] arr = new double[size];  
 double value = deltaT;  
 for (int i = 0; i < size; i++)  
 {  
 arr[i] = value;  
 value += deltaT;  
 }  
 return arr;  
}  
  
*// Метод для обчислення дисперсії*public static double CalcDx(int[] arr, double mx, double n)  
{  
 double sum = 0;  
 foreach (int x in arr)  
 {  
 sum += Math.Pow(x - mx, 2);  
 }  
 return sum / (n - 1);  
}  
  
*// Метод для обчислення масиву значень Kx*public static double[] CalcKx(double[] deltaQh, double[] deltaQMax, double dx)  
{  
 double[] kx = new double[deltaQh.Length];  
 for (int i = 0; i < deltaQh.Length; i++)  
 {  
 kx[i] = CalcKxTa(dx, deltaQh[i], deltaQMax[i]);  
 }  
 return kx;  
}

Chart generation:

public static void DrawLineGraph(double[] xValues, double[] yValues, string filePath)  
{  
 *// Set up the graph size and padding* int width = 900;*// Width of the graph image* int height = 900;*// Height of the graph image  
 // Create a new SKSurface and SKCanvas for drawing* using (var surface = SKSurface.Create(new SKImageInfo(width, height)))  
 {  
 var canvas = surface.Canvas;  
  
 *// Clear the canvas* canvas.Clear(SKColors.White);*// Define the size of the canvas* SKRect canvasRect = new SKRect(0, 0, canvas.LocalClipBounds.Width, canvas.LocalClipBounds.Height);  
  
 *// Define the range of the x and y values* double xMin = xValues.Min();  
 double xMax = xValues.Max();  
 double yMin = yValues.Min();  
 double yMax = yValues.Max();  
  
 *// Define the size of the plot area* SKRect plotRect = new SKRect(canvasRect.Left + 50, canvasRect.Top + 50, canvasRect.Right - 50, canvasRect.Bottom - 50);  
  
 *// Define the range of the plot area* double plotWidth = plotRect.Width;  
 double plotHeight = plotRect.Height;  
 double xRange = xMax - xMin;  
 double yRange = yMax - yMin;  
  
 *// Draw the coordinate axes* using (SKPaint paint = new SKPaint())  
 {  
 paint.Color = SKColors.Black;  
 paint.StrokeWidth = 2;  
  
 *// Draw the x-axis* canvas.DrawLine(plotRect.Left, plotRect.Bottom, plotRect.Right, plotRect.Bottom, paint);  
  
 *// Draw the y-axis* canvas.DrawLine(plotRect.Left, plotRect.Top, plotRect.Left, plotRect.Bottom, paint);  
  
 *// Draw the x-axis labels* for (double x = xMin; x <= xMax; x += xRange / 10)  
 {  
 float xCoord = (float)(plotRect.Left + (x - xMin) / xRange \* plotWidth);  
 canvas.DrawLine(xCoord, plotRect.Bottom - 5, xCoord, plotRect.Bottom + 5, paint);  
 canvas.DrawText(x.ToString("F2"), xCoord, plotRect.Bottom + 20, paint);  
 }  
  
 *// Draw the y-axis labels* for (double y = yMin; y <= yMax; y += yRange / 10)  
 {  
 float yCoord = (float)(plotRect.Bottom - (y - yMin) / yRange \* plotHeight);  
 canvas.DrawLine(plotRect.Left - 5, yCoord, plotRect.Left + 5, yCoord, paint);  
 canvas.DrawText(y.ToString("F2"), plotRect.Left - 40, yCoord, paint);  
 }  
 }  
  
 *// Draw the grid* using (SKPaint paint = new SKPaint())  
 {  
 paint.Color = SKColors.LightGray;  
 paint.StrokeWidth = 1;  
  
 *// Draw the vertical grid lines* for (double x = xMin; x <= xMax; x += xRange / 10)  
 {  
 float xCoord = (float)(plotRect.Left + (x - xMin) / xRange \* plotWidth);  
 canvas.DrawLine(xCoord, plotRect.Top, xCoord, plotRect.Bottom, paint);  
 }  
  
 *// Draw the horizontal grid lines* for (double y = yMin; y <= yMax; y += yRange / 10)  
 {  
 float yCoord = (float)(plotRect.Bottom - (y - yMin) / yRange \* plotHeight);  
 canvas.DrawLine(plotRect.Left, yCoord, plotRect.Right, yCoord, paint);  
 }  
 }  
  
 *// Draw the line graph* using (SKPaint paint = new SKPaint())  
 {  
 paint.Color = SKColors.Blue;  
 paint.StrokeWidth = 1;  
  
 for (int i = 0; i < xValues.Length - 1; i++)  
 {  
 double x1 = xValues[i];  
 double y1 = yValues[i];  
 double x2 = xValues[i + 1];  
 double y2 = yValues[i + 1];  
 float x1Coord = (float)(plotRect.Left + (x1 - xMin) / xRange \* plotWidth);  
 float y1Coord = (float)(plotRect.Bottom - (y1 - yMin) / yRange \* plotHeight);  
 float x2Coord = (float)(plotRect.Left + (x2 - xMin) / xRange \* plotWidth);  
 float y2Coord = (float)(plotRect.Bottom - (y2 - yMin) / yRange \* plotHeight);  
 canvas.DrawLine(x1Coord, y1Coord, x2Coord, y2Coord, paint);  
 }  
 }  
  
 using (SKPaint paint = new SKPaint())  
 {  
 paint.Color = SKColors.Black;  
 paint.TextSize = 30;  
 paint.IsAntialias = true;  
 paint.Typeface = SKTypeface.FromFamilyName("Arial", SKFontStyleWeight.**Bold**, SKFontStyleWidth.**Normal**, SKFontStyleSlant.**Upright**);  
  
 *// Define the watermark text* string watermarkText = "Danylo Serhieiev";  
  
 *// Calculate the size of the watermark text* SKRect watermarkRect = new SKRect();  
 paint.MeasureText(watermarkText, ref watermarkRect);  
  
 *// Calculate the position of the watermark text* float watermarkX = canvasRect.Right - watermarkRect.Width - 50;  
 float watermarkY = canvasRect.Top + watermarkRect.Height + 50;*// adjust the Y position  
  
 // Draw the watermark text* canvas.DrawText(watermarkText, watermarkX, watermarkY, paint);  
 }  
  
  
 *// Save the image to a file* using (var image = surface.Snapshot())  
 using (var data = image.Encode(SKEncodedImageFormat.**Png**, 100))  
 using (var stream = File.OpenWrite(filePath))  
 {  
 data.SaveTo(stream);  
 }  
 }  
}  
  
public static async Task OpenImageFile(string filePath)  
{  
 var processStartInfo = new ProcessStartInfo  
 {  
 FileName = filePath,  
 UseShellExecute = true  
 };  
  
 using var process = Process.Start(processStartInfo);  
 if (process != null)  
 {  
 await Task.Run(() => process.WaitForExit());  
 }  
}

**Результати розрахунків**

Task1:  
1. Визначення періоду опитування датчиків з реалізації випадкових процесів за температурою і тиском

a) Величина Kx(Ta) при заданих QxMax та Qh:

Kx(Ta) = 36.50

б) За графіком кореляційних функцій, визначаємо значення періодів опитування T0 датчиків температури і тиску:

T0 для датчика температури: 2.00

T0 для датчика тиску: 0.10

в) Для постійної величини QxMax визначаємо декілька значень T0 при різних значеннях Qh для датчика температури:

Kx(T0) = 36.50

QxMax = const

K(T - 0) = 36.2450, T = 1.70

K(T - 1) = 36.4050, T = 1.90

K(T - 2) = 36.5000, T = 2.00

K(T - 3) = 36.7200, T = 2.20

K(T - 4) = 36.9800, T = 2.40

г) Для постійної величини Qh визначаємо декілька значень T0 при різних значеннях QxMax для датчика температури:

Qh = const

K(T - 0) = 37.6550, T = 1.70

K(T - 1) = 36.8950, T = 1.90

K(T - 2) = 36.5000, T = 2.00

K(T - 3) = 35.6800, T = 2.20

K(T - 4) = 34.8200, T = 2.40

Датчик тиску:

в) Для постійної величини QxMax визначаємо декілька значень T0 при різних значеннях Qh:

Kx(T0) = 16.50

QxMax = const

K(T - 0) = 16.2450, T = 0.04

K(T - 1) = 16.4050, T = 0.08

K(T - 2) = 16.5000, T = 0.10

K(T - 3) = 16.7200, T = 0.14

K(T - 4) = 16.9800, T = 0.18

г) Для постійної величини Qh визначаємо декілька значень T0 при різних значеннях QxMax:

Qh = const

K(T - 0) = 17.6550, T = 0.04

K(T - 1) = 16.8950, T = 0.08

K(T - 2) = 16.5000, T = 0.10

K(T - 3) = 15.6800, T = 0.14

K(T - 4) = 14.8200, T = 0.18

д) Можна зробити висновок, що статичні властивості вимірюваної величини (похибки її визначення і похибки вимірювального тракту) впливають на величину Т0.

Зокрема, при збільшенні похибок вимірювання і похибок вимірювального тракту, значення Т0 збільшується.

Також можна зробити висновок, що збільшення значення Qh при постійному значенні QxMax призводить до зменшення значення Т0, а збільшення значення QxMax при постійному значенні Qh призводить до збільшення значення Т0.

2. Визначення періоду опитування датчиків за кривими реалізації випадкового процесу:

а) Визначаємо крок дискретизації випадкового процесу:

Час, протягом якого відбулося 100 перетинів: 10.00 мм/сек

Середнє число нулів за одиницю часу: 10.00

Крок дискретизації Δτ: 0.01

б) Визначаємо статистичні характеристики випадкового процесу:

Mx (Математичне очікування): 102.70

Dx (Дисперсія): 38.12

Kx (Кореляційна функція): [33.32, 32.72, 32.12, 31.52, 30.92, 30.62, 30.02, 29.42, 28.82, 28.22]

в) Визначаємо період опитування датчика:

t0 = 0.46

(J0) = 30.77

deltaTArr (Масив кроків дискретизації випадкового процесу): [0.01, 0.03, 0.04, 0.06, 0.07, 0.09, 0.10, 0.12, 0.14, 0.15]

A screen shot of a computer

Description automatically generated with low confidence

Графік *Kx(τ0)*.

A screen shot of a graph

Description automatically generated

Diagram

Description automatically generated

Рисунок. – Блок-схема алгоритму розрахунку математичного очікування, дисперсії та кореляційної функції

**Контрольні запитання до лабораторної роботи №2**

1. Проблема вибору необхідної частоти опитування технологічних параметрів.

Питання вибору необхідної частоти опитування технологічних параметрів за допомогою КОМ виникає при створенні автоматизованої системи управління технологічним процесом (АСУ ТП) на етапі розробки технічного завдання (ТЗ).

***Завищена частота опитування*** призводить до ускладнення системи дискретного контролю та підвищенню завантаження обчислювальної частини КОМ.

***Занижена частота опитування*** практично може звести до нуля результати дискретного контролю, оскільки при цьому неможливо простежити з необхідною точністю за зміною контрольованої величини.

1. Ступінчаста екстраполяція вимірюваної величини.

Ступінчаста екстраполяція вимірюваної величини - це метод екстраполяції, коли за допомогою лінійної або іншої функції, яка з'єднує кілька даних точок, виконується прогнозування вимірюваної величини за межі експериментальних даних.

Метод полягає у встановленні наближених значень вимірюваної величини на основі вже виміряних даних, які лежать в межах експериментальної зони, а потім використовується проста математична формула, щоб оцінити вимірювану величину за межами цієї зони.

Проте, метод ступінчастої екстраполяції може бути неточним, особливо якщо функція, яку використовують для з'єднання даних точок, не є достатньою, або якщо залежність між вимірюваною величиною і функцією не є лінійною. Також використання цього методу вимагає великої кількості експериментів, щоб отримати достатньо точні результати.

1. Середня квадратична похибка екстраполяції стаціонарному процесі.

Середня квадратична похибка екстраполяції - це показник точності прогнозування, який використовується для оцінки якості екстраполяції даних за межі експериментального діапазону.

При екстраполяції виникає похибка *ε(t)* оцінки поточного значення вимірюваної величини, що визначається за формулою:

(2.2)

Похибка *ε(t)* є випадковим процесом, особливістю якого є те, що він дорівнює нулю в момент початку кожного періоду квантування (див. рис. 2.1).

Оцінимо середню квадратичну похибку екстраполяції методом (2.1) для випадку, коли вимірюваний процес є стаціонарним.

Трансформуємо рівняння (2.2), додавши в праву частину і віднявши від неї величину математичного очікування процесу *Мx:*

(2.3)

де *X(t)* та *X(jTo)* - відповідні центровані процеси

Зведемо ліву і праву частини виразу (2.3) в квадрат і будемо розглядатиповедінку функції *ε2(t)* на окремих інтервалах часу або, якщо ввести нову змінну τ = t − jT0, то на інтервалі 0≤τ≤T0.

Усереднюючи по множині інтервалів *j*, отримаємо

(2.4)

при 0 ≤ τ ≤ T0

Використовуючи поняття кореляційної функції та дисперсії, в силу стаціонарності випадкового процесу *X(t)* матимемо

(2.5)

де *Kx* і *Dx* - кореляційна функція і дисперсія випадкового процесу.

З урахуванням (2.5) вираз (2.4) буде записано у вигляді

(2.6)

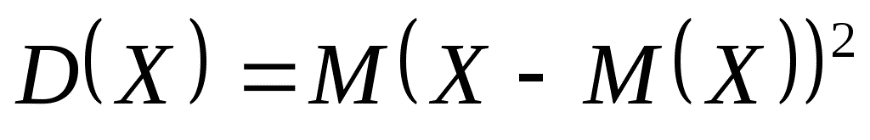
де (𝜏) - середня квадратична похибка визначення величини.

1. Розкажіть алгоритм розрахунку математичного очікування.

Математичне очікування (expected value) - це показник, який вказує на середнє значення випадкової величини.



1. Розкажіть алгоритм розрахунку дисперсії та кореляційної функції.



**Висновок.**

Під час виконання лабораторної роботи я засвоїв методику розрахунку частоти опитування технологічних параметрів за допомогою керуючої обчислювальної машини (КОМ). Я визначив періоди опитування датчиків технологічних параметрів об'єктів, які мають значно відмінні статистичні характеристики. Також я провів дослідження впливу статистичних властивостей вимірюваної величини, похибки її визначення і похибки вимірювального тракту на величину періоду опитування.