IMAGE: A MAP OF THE STARS OF THE ORION CONSTELLATION

Print ISSN: 2631-8474 Online ISSN: 2631-8482

JournalPreview

London Journal of Engineering Research

Volume 25 | Issue 1 | Compilation 1.0

Great Britain Journals Press

WWW.JOURNALSPRESS.COM

Journal Content
In this Issue

Great Britain Journals Press

Computation of the Minimum One-Sided Hausdorff Distance between Plane Curves under a Similarity Transformation and Relevant Applications in Roundness Error Evaluation

Dong Lei, Pan Tao & Chen Min yang

Shanghai DianJi university

ABSTRACT

A two-stage (rough and accurate) computation strategy was developed in this study in order to accurately calculate the minimum one-sided Hausdorff distance between continuous plane curves under a similarity transformation. In the rough computation, a mathematical programming model based on the discretisation method was constructed to ascertain the minimum one-sided Hausdorff distance. In addition, the linearisation method in this model was elucidated. Based on that, the solutions were attained through a stable and efficient simplex method and 5 characteristic points were obtained. In the accurate computation, a local iterative accurate algorithm for computing the minimum one-sided Hausdorff distance was established after 4 similarity transformation parameters were separated from 10 curve parameters corresponding to 5 characteristic points. Similar results, which verify the feasibility of this algorithm, were obtained based on rough and accurate computations in a numerical example. Moreover, a roundness error evaluation programming model based on the minimum one-sided Hausdorff distance and relevant linear solution methods was also developed.

Keywords: continuous plane curve; minimum one-sided hausdorff distance; exact algorithm; comparison of computational efficiency; roundness error evaluation.

Classification: FOR Code: 010202

Language: English

Great Britain Journals Press

LJP Copyright ID: 392911
Print ISSN: 2631-8474
Online ISSN: 2631-8482

London Journal of Engineering Research

Volume 25 | Issue 1 | Compilation 1.0

Computation of the Minimum One-Sided Hausdorff Distance between Plane Curves under a Similarity Transformation and Relevant Applications in Roundness Error Evaluation

Dong Lei[α], Pan Tao[σ] & Chen Min yang[ρ]

ABSTRACT

A two-stage (rough and accurate) computation strategy was developed in this study in order to accurately calculate the minimum one-sided Hausdorff distance between continuous plane curves under a similarity transformation. In the rough computation, a mathematical programming model based on the discretisation method was constructed to ascertain the minimum one-sided Hausdorff distance. In addition, the linearisation method in this model was elucidated. Based on that, the solutions were attained through a stable and efficient simplex method and 5 characteristic points were obtained. In the accurate computation, a local iterative accurate algorithm for computing the minimum one-sided Hausdorff distance was established after 4 similarity transformation parameters were separated from 10 curve parameters corresponding to 5 characteristic points. Similar results, which verify the feasibility of this algorithm, were obtained based on rough and accurate computations in a numerical example. Moreover, a roundness error evaluation programming model based on the minimum one-sided Hausdorff distance and relevant linear solution methods was also developed. Furthermore, the numerical examples based on this model were compared with those based on a conventional roundness error computation model. The results revealed that similar computation circle centre coordinates, roundness error, and characteristic points can be obtained based on both models. The computational efficiency can be significantly improved via the method proposed in this study.

Keywords: continuous plane curve; minimum one-sided hausdorff distance; exact algorithm; comparison of computational efficiency; roundness error evaluation.

Author α: Shanghai DianJi university, 300 Shui Hua Rd., Shanghai 201306, People's Republic of China.
σ: Shanghai DianJi university, 300 Shui Hua Rd., Shanghai 201306, People's Republic of China.
ρ: Shanghai DianJi university, 300 Shui Hua Rd., Shanghai 201306, People's Republic of China.

I. INTRODUCTION

Early Hausdorff distance computation is mainly applied to pattern recognition and image processing, including the recognition of faces, fingerprints, characters, handwriting, and licence plates. However, there has been little investigation into the computation of the Hausdorff distance relative to non-point sets. In the conventional method, continuous geometric objects are discretised into point sets, based on which the approximate Hausdorff distance between continuous geometric objects can be indirectly obtained by computing the Hausdorff distance between point sets. More recently the computation of the Hausdorff distance between continuous geometric objects has been studied owing to potential demands in geometric modelling, computational geometry, computer graphics, and other fields. In 2008, Alt et al.[1] from Germany demonstrated four cases of the one-sided Hausdorff distance between two C^1 continuous plane curves. Specifically, the Hausdorff distance may present at two endpoints of two curves, with one of the two-points being the end-point of one curve, double perpendicular foot

points, or the intersection point of the midline of one curve and another curve. Furthermore, they provided nonlinear constraint simultaneous equations corresponding to the four cases. The Hausdorff distance between plane curves was computed with the aid of a standard algebraic equations solver. In the same year, Elber et al.[2] from Israel generalised these four cases of the one-sided Hausdorff distance between two C^1 continuous plane curves to space curves/ curved surfaces. They obtained the corresponding Hausdorff distance using a self-designed algebraic equations solver. In 2010, Chen et al.[3] tackled the lack of research in finding the root of nonlinear simultaneous equations in previous studies [1-2].

They explored the geometric clipping method for calculating the Hausdorff distance between two B-spline curves. As per their algorithm, sufficient conditions allowing the Hausdorff distance to appear at the end of a curve were provided. The curve segmentation technique and rolling circle clipping method were adopted to transform the Hausdorff distance computation between two curves into that of the minimum or maximum distance between points and curves, thus improving the stability and computational efficiency of the algorithm. In the same year, Kim et al.[4] proposed a real-time algorithm to accurately calculate the Hausdorff distance between two plane free-form curves. Firstly, G^1 continuous double circular arcs were employed to approximate the free-form curves under given tolerances. The arcs were subsequently subjected to distance mapping and saved to a graphics hardware depth buffer. Finally, most of the redundant arc segments were trimmed to improve the computational efficiency related to the Hausdorff distance. In 2011, Bai et al.[5] proposed a polyline method to obtain the Hausdorff distance between plane curves. Based on the algorithm, continuous free-form curves were approximated with polylines under a given measurement. An incremental algorithm was subsequently utilised to compute the one-sided Hausdorff distance between line segments and polylines. Since two clipping strategies are used in this method, the ineffective line segments in the final Hausdorff distance computation are clipped which significantly improves the computational efficiency.

In terms of the computation of the minimum Hausdorff distance between geometric objects under a certain transformation, Huttenlocher[6] proposed an algorithm to compute the minimum Hausdorff distance for one- and two-dimensional point sets. The one-dimensional algorithm of the minimum Hausdorff distance between point sets was applied to the comparison computation of planar polygons under an affine transformation by using the affine arc length to represent planar polygons. The two-dimensional algorithm was used to compare digital images. Additionally, the Hausdorff distance definition was extended to enable it to be used for the comparison between partial sets. In further studies, Huttenlocher[7] improved the minimum Hausdorff distance algorithm under a translation transformation between point sets and line segment sets by using a Voronoi diagram. According to two given polygons, Alt[8] proposed that the Hausdorff distance between two polygons can be minimised by the rigid transformation of one of the polygons, and therefore the approximate matching between polygons can be realised. Chew et al.[9] investigated the computational complexity of the minimum Hausdorff distance algorithm between geometric objects under a rigid transformation for planar line segments that are composed of point sets and polygon sets that are composed of point sets. Hur et al.[10] adopted the conic section represented by a quadratic rational Bezier curve to perform the best uniform approximation (BUA) for the convolution curve of two plane curves. The objective was to minimise the Hausdorff distance between the conic section and the convolution curve. Because the weight factor of the rational Bezier curve was the only optimisation variable in the approximation curve in the model, the study revealed that there were two characteristic points on the approximation curve, and both the distance between them and the convolution curve were the Hausdorff distance, and the derivative signs of the two distances with respect to the weight factor were opposite. This constituted the necessary and sufficient conditions for minimising the Hausdorff distance between two curves. An iterative algorithm was also formulated for computing the BUA based on the necessary and sufficient conditions.

To ensure equal precision and computational efficiency, Gu et al.[11] proposed a rough and accurate staged matching method between plane curves under an affine transformation. Liang et al.[12] designed a matching method between plane curves using their projection along any direction based on a weak perspective projection model. Gruen et al.[13] established a similarity transformation model consisting of seven parameters between curved surfaces as per the least squares principle. In addition, they considered the matching method between curved surfaces and applied it to the automatic mosaic of 3D point cloud data. Rodriguez et al.[14] analysed the similarity computation between 3D curves through the string edit distance method. Yamany et al.[15] formulated a surface registration method based on a genetic algorithm. Cao et al.[16] transformed the Hausdorff distance computation between plane curves into that of the minimum distance from points to curves. This method eliminated the drawback of traditional methods, whereby different nonlinear simultaneous equations had to be solved separately under four Hausdorff distance cases between plane curves. Moreover, they constructed an algorithm to compute the one-sided Hausdorff distance between C2 continuous curves via two steps, namely rough and accurate computations. Based on a previous study[16], Cao et al.[17] established a programming model for the minimum one-sided Hausdorff distance between plane curves under a rigid transformation and developed relevant solutions. Furthermore, they applied the model to line profile error evaluation.

As suggested in relevant studies, the computation of the Hausdorff distance between continuous curves and curved surfaces has attracted the attention of researchers in geometric modelling, computational geometry, computer graphics, and other fields in recent years. However, it also prevents the wider application of the Hausdorff distance in engineering. There are several issues in relevant research work. 1) A few studies have been conducted on the computation of the Hausdorff distance between free-form curves and curved surfaces. In most cases, the Hausdorff distance was obtained by solving simultaneous algebraic equations. Since the solution of these algebraic equations was either regarded as a black box or obtained by a standard solver, the solutions with a reference value and geometric significance cannot be fully provided. 2) Due to the complexity of concrete curves and curved surfaces in engineering, research on the computation of the Hausdorff distance between continuous curves or curved surfaces and the minimum Hausdorff distance under a certain transformation is still in its infancy. At the same time the lack of relevant solution methods significantly hinders its extensive application in the engineering field.

In this study, a mathematical programming model for computing the minimum one-sided Hausdorff distance between plane curves under a similarity transformation was constructed based on a previous study[17] by supplementing the scale transformation parameters of approximation objects. A solution strategy composed of two-stage (rough and accurate) computation was proposed for this model. Moreover, a local iterative accurate algorithm for computing the minimum one-sided Hausdorff distance for continuous curves was also constructed. On that basis, a roundness error evaluation programming model, based on the minimum one-sided Hausdorff distance, was established and then verified numerically.

II. COMPUTATION OF THE MINIMUM ONE-SIDED HAUSDORFF DISTANCE BETWEEN PLANE CURVES UNDER A SIMILARITY TRANSFORMATION

2.1 Mathematical Programming Model

It can be inferred from the definition of the one-sided Hausdorff distance that the computation of $h(A,B)$ involves the shortest distance from a point a on Curve A to Curve B. As per geometric error evaluation theory and curve/curved surface approximation theory, this shortest distance is generally called an error. When Curve B is allowed to undergo a similarity transformation, its position and scale

are transformed. The change in the parameter of the point on Curve B corresponding to the shortest distance from a point on Curve A to Curve B is termed parameter transformation. When the shortest distance from Point a to the transformed Curve B is also transformed, it is termed error transformation. Therefore, the computation of $h(A,B)$ under a similarity transformation incorporates position, scale, parameter, and error transformations.

The representation of Curve B under a similarity transformation is subsequently discussed. The similarity transformation of the given plane curve B includes three motions; namely translation, rotation, and scaling. The following homogeneous coordinates can be used to represent the translation matrix, rotation matrix, and scaling matrix, respectively, including

$$\mathbf{T}(t_x,t_y)=\begin{bmatrix}1 & 0 & t_x\\ 0 & 1 & t_y\\ 0 & 0 & 1\end{bmatrix}\quad \mathbf{R}(\theta)=\begin{bmatrix}\cos\theta & -\sin\theta & 0\\ \sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix}\quad \mathbf{S}(s_x,s_y)=\begin{bmatrix}s_x & 0 & 0\\ 0 & s_y & 0\\ 0 & 0 & 1\end{bmatrix} \tag{1}$$

where, t_x,t_y represents the translation coordinates along the X and Y axes; θ represents the rotation angle; s_x,s_y represents the scaling rations in the X and Y directions; To maintain the shape of the graph unchanged before and after scaling, equal scaling would be adopted in the following discussion, namely $s_x=s_y=s$. These parameters t_x,t_y,θ,s are collectively called the similarity transformation parameter $\mathbf{x}$.

After the above three motions, the expression of Curve B can be obtained as:

$$\mathbf{B}(b,\mathbf{x})=\mathbf{T}(t_x,t_y)\cdot\mathbf{S}(s)\cdot\mathbf{R}(\theta)\cdot\mathbf{B}(b)=\begin{bmatrix}s\cos\theta & -s\sin\theta & t_x\\ s\sin\theta & s\cos\theta & t_y\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}X\\ Y\\ 1\end{bmatrix} \tag{2}$$

$$=(Xs\cos\theta-Ys\sin\theta+t_x)\mathbf{i}+(Xs\sin\theta+Ys\cos\theta+t_y)\mathbf{j}$$

where, X,Y denotes the coordinates of Curve B in Frame $\{o,\mathbf{ij}\}$; they are all functions of the curve parameter b.

From the perspective of the BUA of curves or curved surfaces, the one-sided Hausdorff distance between two geometric objects is the maximum deviation of the approximation object from the approximation target. Minimising the maximum deviation under some transformation is the optimisation objective of the BUA. The process of minimising the maximum deviation is characterised by saddle points and can therefore be explored by saddle point programming. Under a similarity transformation, if the scale and position parameters of the approximation object are taken as optimisation variables and the one-sided Hausdorff distance from the approximation object to the approximation target is minimised as the optimisation objective, the definition of the minimum one-sided Hausdorff distance can be expressed as the following mathematical programming form:

$$\begin{aligned}&\min_{\mathbf{x}} \quad h\\ &s.t. \quad h=\max_a \min_b d(\mathbf{A}(a),\mathbf{B}(b,\mathbf{x}))\end{aligned} \tag{3}$$

The first line in Equation (3) represents the objective function of the optimisation model and h is the characteristic parameter. The second line represents the optimisation model constraint function; namely that the characteristic parameter h shall be the one-sided Hausdorff distance between two curves under a certain set of transformation parameters $\mathbf{x}$. $d(\mathbf{A}(a),\mathrm{B}(b,\mathbf{x}))=\left\|\mathbf{A}(a)-\mathbf{B}(b,\mathbf{x})\right\|$ represents the Euclidean distance between Point a on Curve A and Point b on Curve B under the transformation parameter X.

The constraint function in the above model is the one-sided Hausdorff distance between two curves under the transformation parameter $\mathbf{x}$. According to the definition of the one-sided Hausdorff distance between continuous curves in a previous study[16], it can be found that a pair of points satisfying the one-sided Hausdorff distance is the point pair satisfying the maximisation of the minimum distance and it also has the saddle point characteristics. Moreover, a saddle point programming model for the Hausdorff distance computation between two curves can be established if the curve parameters a and b of two curves are taken as optimisation variables and the minimum distance from a point on one curve to the other is taken as the optimisation objective. In other words, the computation of the minimum one-sided Hausdorff distance is an optimisation in Type $\min\max\min f$. To simplify the optimisation process, Curve A would be discretised to complete the computation of the shortest distance from each discrete point on Curve A to Curve B. On that basis, the following mathematical programming model can be constructed to obtain the solution to the minimum one-sided Hausdorff distance:

Following the discretisation of Curve A, the programming model given by Equation (3) can be expressed as:

$$\begin{aligned} &\min_{\mathbf{x}} \quad f(\mathbf{y})=h \\ &s.t. \quad G_i(\mathbf{y})=d_i(\mathbf{A}(a_i),\mathbf{B}(b_i,\mathbf{x}))-h\le 0 \qquad i=1,2,\ldots,m \end{aligned} \tag{4}$$

where, $d_i(\mathbf{A}(a_i),\mathbf{B}(b_i,\mathbf{x}))$ represents the shortest distance from the discrete point a_i on Curve A to the transformed curve $B(b,\mathbf{x})$, as presented in Figure 1. This can be computed according to the shortest distance algorithm from the point to the curve in a previous study[16]; $b_i=b_i(a_i)$ represents the curve parameter corresponding to the minimum normal distance mapping point of the discrete point a_i on Curve $B(b,\mathbf{x})$; i represents the discrete point serial numbers, and m represents the total number of discrete points on Curve A. $\mathbf{x}=(t_x,t_y,\theta,s)$ represents a similarity transformation parameter, and $\mathbf{y}=(h,\mathbf{x})=(h,t_x,t_y,\theta,s)$ represents the optimisation variables of the saddle point programming model.

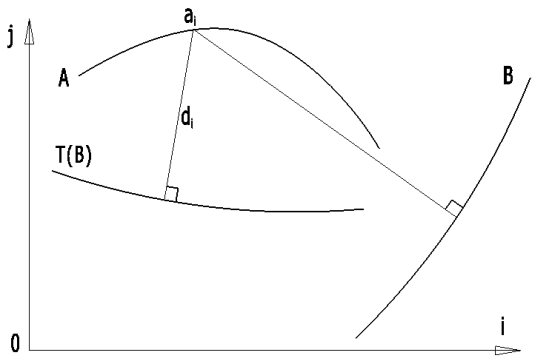


Fig. 1: Planar Curves under Similarity Transformation

It can be revealed from Equations (2) and (3) that the optimisation model constraint function given in Equation (4) is not a linear combination of optimisation variables, but a nonlinear optimisation which needs to be linearised.

2.2 Linearisation Solution to the Mathematical Programming Model

It can be seen from the above optimisation model that the nonlinear part of the optimisation variable is included in the shortest distance d_i. If the shortest distance in Equation (4) is regarded as a function of the transformation parameter $\mathbf{x}$, it can be expressed as follows:

$$d_i = d_i(\mathbf{A}(a_i),\mathbf{B}(b_i,\mathbf{x})) = \left\|\mathbf{A}(a_i)-\mathbf{B}(b_i,\mathbf{x})\right\| = g(\mathbf{x}) = g(t_x,t_y,\theta,s) \tag{5}$$

If the transformation parameter t_x,t_y,θ,s is regarded as a first-order small quantity, it is called a small error. After $g(\mathbf{x})$ in Equation (5) is subject to Taylor's expansion at the zero point $\mathbf{x}_0=(t_{x0},t_{y0},\theta_0,s_0)^{\mathrm{T}}$ of transformation parameters and their second-order and higher-order small quantities are omitted, the following expression can be obtained:

$$g(\mathbf{x}) = g(\mathbf{x}_0)+\frac{\partial g}{\partial t_x}\Delta t_x+\frac{\partial g}{\partial t_y}\Delta t_y+\frac{\partial g}{\partial \theta}\Delta\theta+\frac{\partial g}{\partial s}\Delta s \tag{6}$$

In the equation,

$$g(\mathbf{x}_0) = \left\|\mathbf{A}(a_i)-\mathbf{B}(b_i,\mathbf{x})\right\|_{\mathbf{x}=\mathbf{x}_0} \tag{7}$$

$$\frac{\partial g}{\partial t_x} = -\frac{\left[\mathbf{A}(a_i)-\mathbf{B}(b_i,\mathbf{x})\right]\cdot\mathbf{B}'_x(b_i,\mathbf{x})}{\left\|\mathbf{A}(a_i)-\mathbf{B}(b_i,\mathbf{x})\right\|}\bigg|_{\mathbf{x}=\mathbf{x}_0} = -\mathbf{N}_i\cdot\mathbf{i} = -N_{xi} \tag{8}$$

$$\frac{\partial g}{\partial t_y} = -\frac{\left[\mathbf{A}(a_i)-\mathbf{B}(b_i,\mathbf{x})\right]\cdot\mathbf{B}'_y(b_i,\mathbf{x})}{\left\|\mathbf{A}(a_i)-\mathbf{B}(b_i,\mathbf{x})\right\|}\bigg|_{\mathbf{x}=\mathbf{x}_0} = -\mathbf{N}_i\cdot\mathbf{j} = -N_{yi} \tag{9}$$

$$\frac{\partial g}{\partial \theta} = -\frac{\left[\mathbf{A}(a_i)-\mathbf{B}(b_i,\mathbf{x})\right]\cdot\mathbf{B}'_\theta(b_i,\mathbf{x})}{\left\|\mathbf{A}(a_i)-\mathbf{B}(b_i,\mathbf{x})\right\|}\bigg|_{\mathbf{x}=\mathbf{x}_0} = -\mathbf{N}_i\cdot(-Y_i\mathbf{i}+X_i\mathbf{j}) = Y_iN_{xi}-X_iN_{yi} \tag{10}$$

$$\frac{\partial g}{\partial s} = -\frac{\left[\mathbf{A}(a_i)-\mathbf{B}(b_i,\mathbf{x})\right]\cdot\mathbf{B}'_s(b_i,\mathbf{x})}{\left\|\mathbf{A}(a_i)-\mathbf{B}(b_i,\mathbf{x})\right\|}\bigg|_{\mathbf{x}=\mathbf{x}_0} = -\mathbf{N}_i\cdot(X_i\mathbf{i}+Y_i\mathbf{j}) = -X_iN_{xi}-Y_iN_{yi} \tag{11}$$

where, $g(\mathbf{x}_0)$ represents the shortest distance from a point a_i on Curve A to Curve B before a similarity transformation; Δt_x, Δt_y, $\Delta\theta$, and Δs represent the first-order increment of the transformation parameter t_x,t_y,θ,s; X_i,Y_i represents the coordinates of the point on Curve B; N_{Xi},N_{Yi} represents the coordinates of the unit normal vector at a point on Curve B; $\mathbf{B}'_x,\mathbf{B}'_y,\mathbf{B}'_\theta,\mathbf{B}'_s$ represents the first-order partial derivatives of Equation (2) for the transformation parameter t_x,t_y,θ,s.

After Equations (6) - (11) are substituted into Equation (4), the following linear programming model for computing the minimum one-sided Hausdorff distance between plane curves can be obtained:

$$\begin{aligned} &\min_{\mathbf{x}} \quad f(\mathbf{y}) = h \\ &s.t. \quad G_i(\mathbf{y}) = \delta_i - N_{xi}\Delta t_x - N_{yi}\Delta t_y - (X_i N_{yi} - Y_i N_{xi})\Delta\theta - (X_i N_{xi} + Y_i N_{yi})\Delta s - h \le 0 \\ &\quad i = 1, 2, \cdots, m \end{aligned} \tag{12}$$

where, h represents the characteristic parameter, that is, the one-sided Hausdorff distance from curve A to curve B; $\delta_i = g(\mathbf{x}_0)$; and m represents the number of discrete points on the actual curve.

The above optimisation model has the variable $h \ge 0$. However, $\Delta\theta, \Delta t_x, \Delta t_y, \Delta s$ are all free variables without non-negative requirements. To achieve the solutions by the simplex linear programming method, it can be arranged that $\mathbf{X}_0 = (\Delta\theta, \Delta t_x, \Delta t_y, \Delta s)^{\mathrm{T}} = \mathbf{X}_1 - \mathbf{X}_2$ and $\mathbf{X}_1, \mathbf{X}_2 > 0$. Then, the slack variable ($\mathbf{X}_3 \ge 0$) is introduced and the inequality constraint is transformed into an equality constraint, finally an initial basic feasible solution can be obtained. Hence, the above linear programming problem can be expressed in the following standard form:

$$\left.\begin{array}{lll} \min & \mathbf{C}^{\mathrm{T}}\mathbf{X} & \\ \text{s.t.} & \mathbf{AX} = \mathbf{b} & \mathbf{X} \ge 0 \end{array}\right\} \tag{13}$$

where, $\mathbf{C} = (1, 0, ..., 0)^{\mathrm{T}}$, $\mathbf{X} = (h, \mathbf{X}_1^{\mathrm{T}}, \mathbf{X}_2^{\mathrm{T}}, \mathbf{X}_3^{\mathrm{T}})^{\mathrm{T}}$ represents a dimensional column vector of $(m+9)$, $\mathbf{A} = (\mathbf{I}_0, \mathbf{A}_0, -\mathbf{A}_0, -\mathbf{I})$ represents a matrix of $m \times (m+9)$, and $\mathbf{I}$ represents an m-order unit matrix; $\mathbf{b} = (\delta_1, ..., \delta_m)^{\mathrm{T}}$, $\mathbf{I}_0 = (1, ..., 1)^{\mathrm{T}}$, and $\mathbf{X}_3$ represent m-dimensional column vectors, and $\mathbf{X}_0, \mathbf{X}_1, \mathbf{X}_2$ represents a 4-dimensional column vector; The upper corner mark T represents the matrix transposition.

$$\mathbf{A}_0 = \begin{bmatrix} N_{X1} & N_{Y1} & X_1 N_{Y1} - Y_1 N_{X1} & X_1 N_{X1} + Y_1 N_{Y1} \\ \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{M} \\ N_{Xm} & N_{Ym} & X_m N_{Ym} - Y_m N_{Xm} & X_m N_{Xm} + Y_m N_{Ym} \end{bmatrix}_{m \times 4}$$

After the above model is solved by the simplex linear programming method, the characteristic parameter h and the similarity transformation parameter $\Delta\theta, \Delta t_x, \Delta t_y, \Delta s$ of Curve B can be obtained. The value of the characteristic parameter can be considered as the minimum one-sided Hausdorff distance from curve A to curve B.

To reduce any error caused by the linear geometric model, curve B can be rotated, translated, and scaled, using the computation result of $\Delta\theta, \Delta t_x, \Delta t_y, \Delta s$. The transformed curve can be regarded as a new curve and the above process can be repeated until the absolute value of the transformation parameters obtained from the computation of Equation (12) is less than the given precision.

Under the similarity transformation, one of the two curves (e.g., Curve B) can usually be translated, rotated, and scaled, so there are four degrees of freedom in Curve B. Therefore, when $h(A,B)$ reaches its minimum, the equidistant line of Curve B has at least 5 points of tangency to Curve A, as shown in Figure 2. Under this condition, due to the continuous changes in the Curve B scale parameter s, when $h(A,B)$ reaches the minimum, both the symmetric equidistant curves B_o and B_i of Curve B have points of tangency to Curve A. In other words, Curve A is bidirectionally enclosed by the inner and outer equidistant lines of Curve B and the maximum and minimum normal errors of Curve A relative to Curve B are equal in value and opposite in sign. The distribution ratio of the tangency points for the inner and outer equidistant curves B_o and B_i to Curve A can be 1:4 (4:1) or 2:3 (3:2).

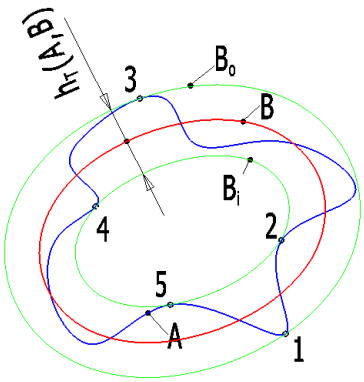


Fig. 2: Minimum one-sided HD from curve A to curve B where curve A is fixed and curve B subjects to the similarity transformation

2.3 Precise Iterative Model

The mathematical programming model in the previous section can be utilised to compute the minimum one-sided Hausdorff distance from discretised curve A to curve B. When the curve is originally provided in the form of discrete points, for example, when the curve is a data point obtained by measurement or the computational precision is not high, this method can meet the relevant requirements. However, when Curves A and B are presented as continuous curves, in order to obtain the precise minimum one-sided Hausdorff distance between both curves, it is necessary to conduct local iterative optimisations based on the previous section. Finally, the precise minimum one-sided Hausdorff distance from curve A to curve B can be obtained.

The approximate solution which satisfies the optimal condition can be obtained via the algorithm in the previous section. There are 5 characteristic points on discretised Curve A which are optimally arranged with the inner and outer equidistant lines of Curve B. It is necessary to construct a local iterative optimisation model, so that the parameters of Curves A and B, corresponding to 5 equivalent characteristic points, can meet the requirements of the one-sided Hausdorff distance.

As the local optimisation model contains 4 similarity transformation parameters and 10 curve parameters corresponding to 5 characteristic points, with a total of 14 optimisation variables, it can be considered a complex nonlinear problem. If these parameters can be solved together iteratively, the solution process is prone to oscillation, which may induce convergence failure in the computation. To reduce the computational difficulty and ensure computational stability, the precise iterative model is iteratively solved using two steps (similarity transformation parameters and curve parameters).

2.3.1 Similarity Transformation Parameter Iterative Solution

There are 5 initial points $\mathbf{P}_i(i=1,\ldots,5)$ on Curve A and the mapping points $\mathbf{Q}_i(i=1,\ldots,5)$ with the shortest distance of $\mathbf{P}_i$ on Curve B can be obtained by the algorithm of the shortest distance from points to curves. After the position of Point $\mathbf{P}_i$ is fixed, the distance between corresponding $\|\mathbf{P}_i\mathbf{Q}_i\|$ can be equalised by changing the position, direction, and scale of Curve B. Therefore, the conditional expression can be provided:

$$\left.\begin{aligned}\|\mathbf{P}_1\mathbf{Q}_1\| &= \|\mathbf{P}_2\mathbf{Q}_2\| \\ \|\mathbf{P}_2\mathbf{Q}_2\| &= \|\mathbf{P}_3\mathbf{Q}_3\| \\ \|\mathbf{P}_3\mathbf{Q}_3\| &= \|\mathbf{P}_4\mathbf{Q}_4\| \\ \|\mathbf{P}_4\mathbf{Q}_4\| &= \|\mathbf{P}_5\mathbf{Q}_5\|\end{aligned}\right\} \tag{14}$$

where, the position of Point $\mathbf{Q}_i$ is related to the similarity transformation parameter of Curve B and its curve parameter b. Since Point $\mathbf{Q}_i$ is the normal mapping point with the shortest distance of $\mathbf{P}_i$ on Curve B, its curve parameter is not independent, so $b=b(a)$. Hence, Equation (14) can be expressed as:

$$\left.\begin{aligned}f_1(t_x,t_y,\theta,s)&=0\\ f_2(t_x,t_y,\theta,s)&=0\\ f_3(t_x,t_y,\theta,s)&=0\\ f_4(t_x,t_y,\theta,s)&=0\end{aligned}\right\} \tag{15}$$

After Equation (15) is subject to Taylor's expansion near the initial point and the small quantity above the second order is omitted, the following expression can be obtained:

$$\left.\begin{aligned}f_{10}+f'_{1x}\Delta t_x+f'_{1y}\Delta t_y+f'_{1\theta}\Delta\theta+f'_{1s}\Delta s&=0\\ f_{20}+f'_{2x}\Delta t_x+f'_{2y}\Delta t_y+f'_{2\theta}\Delta\theta+f'_{2s}\Delta s&=0\\ f_{30}+f'_{3x}\Delta t_x+f'_{3y}\Delta t_y+f'_{3\theta}\Delta\theta+f'_{3s}\Delta s&=0\\ f_{40}+f'_{4x}\Delta t_x+f'_{4y}\Delta t_y+f'_{4\theta}\Delta\theta+f'_{4s}\Delta s&=0\end{aligned}\right\} \tag{16}$$

where, $f_{i0}(i=1,2,3,4)$ represents the function value of Function $f_i(t_x,t_y,\theta,s)$ at the initial point; $f'_{ix},f'_{iy},f'_{i\theta},f'_{is}$ represents the first-order partial derivative of Function $f_i(t_x,t_y,\theta,s)$ with respect to t_x,t_y,θ,s, respectively. Equation (16) can be represented in a matrix form and the following expression can be obtained:

$$\begin{bmatrix}f'_{1x} & f'_{1y} & f'_{1\theta} & f'_{1s}\\ f'_{2x} & f'_{2y} & f'_{2\theta} & f'_{2s}\\ f'_{3x} & f'_{3y} & f'_{3\theta} & f'_{3s}\\ f'_{4x} & f'_{4y} & f'_{4\theta} & f'_{4s}\end{bmatrix}\begin{bmatrix}\Delta t_x\\ \Delta t_y\\ \Delta\theta\\ \Delta s\end{bmatrix}=\begin{bmatrix}-f_{10}\\ -f_{20}\\ -f_{30}\\ -f_{40}\end{bmatrix} \tag{17}$$

The linear equations (17) can be solved to obtain the increments $(\Delta t_x, \Delta t_y, \Delta\theta, \Delta s)$ of similarity transformation parameters and these increments can be employed to update the Curve B similarity transformation parameters. Furthermore, the iterative process can be repeated until each similarity transformation parameter increment is smaller than the given computation precision ε .

2.3.2 Iterative Solution of Curve Parameters

When the minimum one-sided Hausdorff distance from curve A to curve B is achieved, the optimality condition and the one-sided Hausdorff distance condition are satisfied. The previous similarity transformation parameter iterative process ensured that the optimality conditions were satisfied. The condition satisfying the one-sided Hausdorff distance is discussed in this section. As reported in a previous study[16], the one-sided Hausdorff distance between plane curves is generally presented in four cases; double perpendicular foot points, one-to-two points, EA endpoints, and EB endpoints. Different constraint equations exist for different cases. The corresponding constraint equations are established according to the different situations of the 5 points on the actual Curve A before the iterative solution is performed. In this study, the one-sided Hausdorff distance between plane curves under the perpendicular foot point is selected as the example, and its corresponding conditions are:

$$(\mathbf{P}_i - \mathbf{Q}_i) \cdot \mathbf{B}_i' = 0 \tag{18}$$

$$(\mathbf{P}_i - \mathbf{Q}_i) \cdot \mathbf{A}_i' = 0 \qquad i = 1, 2, \mathrm{L}\ , 5 \tag{19}$$

Since the conditions of Equation (18) would definitely be satisfied, it is only necessary to locally optimise the parameters of Curve A at the characteristic point. After the Curve B similarity transformation parameters and the corresponding parameters $b_i (i = 1, 2, ..., 5)$ of characteristic points on Curve B are set as initial points, the parameters of Curve A are the only undetermined variables in Equation (19). Hence, Equation (19) can be expressed as:

$$g_i(a_i) = 0 \qquad i = 1, 2, ..., 5 \tag{20}$$

where, $a_i (i = 1, 2, ..., 5)$ represents the curve parameter corresponding to the 5 characteristic points on Curve A.

Since the solution of Equation (20) is performed based on the previous programming algorithm and has a favourable initial point, the curve parameter increments $\Delta a_i (i = 1, 2, .., 5)$ corresponding to the 5 characteristic points on Curve A can be directly obtained by the efficient univariate Newton's iterative method. On that basis, each Curve A parameter can be updated and the iterative process can be repeated until the parameter increment is smaller than the given computation precision ε .

Due to the interaction between the iterative solution of the similarity transformation parameters and that of the curve parameters, it is necessary to conduct these two iterative processes alternately until the convergence conditions of the two iterative processes satisfy the requirements.

1.3 Numerical examples

The feasibility of the above algorithm is verified via numerical examples in this section. The original A and B curves are closed curves represented by the cubic uniform B spline, incorporating 90 control points. As shown in Figure 3, the red line represents Curve A and the blue line represents Curve B. Curve A is presented as an outline of a Liriodendron chinense leaf. The Curve B control points are

obtained by superimposing an error function with a 2mm amplitude on 90 Curve A control points. Furthermore, the Curve B is transformed according to the following similarity transformation parameters: t_{x0} =15mm, t_{y0} =15mm, θ_0 =20°, and s_0 =1.2.

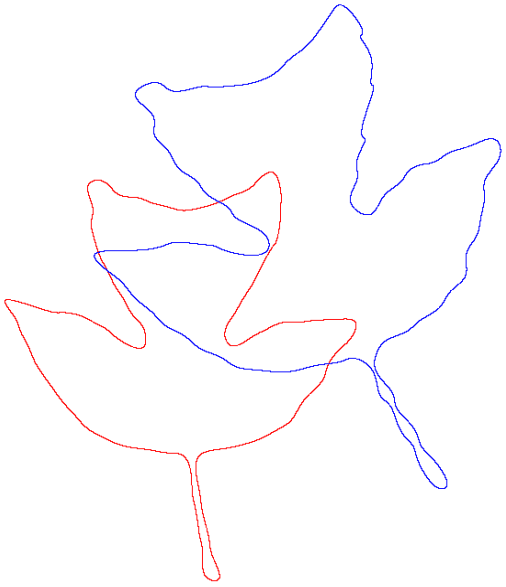


Fig. 3: The Original Curves A and B

The minimum one-sided Hausdorff distance $h_T(A,B)$ from Curve A to Curve B can be calculated using the following process. Firstly, Curve A is discretised according to its parameters and Curve B is transformed from a B-spline form to a cubic piecewise Bezier curve through the algorithm in a previous study[18]. Subsequently, the shortest distance from each discrete point on Curve A to Curve B is computed through the algorithm in a previous study[16], which can be regarded as the original normal error δ_i of Curve A relative to Curve B. The linear programming algorithm in Section 1.1 is then employed to compute the minimum one-sided Hausdorff distance from curve A to curve B under a similarity transformation. Subsequently, the corresponding transformation parameters, minimum one-sided Hausdorff distance $h_T(A,B)$, and corresponding characteristic point coordinates can be obtained. The computation precision is 10^{-10} and the results are presented in Figure 4. The 5 pairs of blue points represent the corresponding characteristic points obtained by the mathematical programming method, and $h_T(A,B)$ =2.899780. The corresponding transformation parameters include Δt_x =16.488852mm, Δt_y =15.939305mm, $\Delta\theta$ =19.803339°, and Δs =1.1960891. Based on the programming model computation results, $h_T(A,B)$ can be solved precisely based on the local optimisation model in Section 1.2. The final characteristic points are presented as 5 pairs of red points in Figure 4, and $h_T(A,B)$ =2.914359mm, which is highly similar to the mathematical programming method results. The corresponding transformation parameters include Δt_x =0.008716mm, Δt_y =0.001226mm, $\Delta\theta$ =0.007190°, and Δs =1.000028.

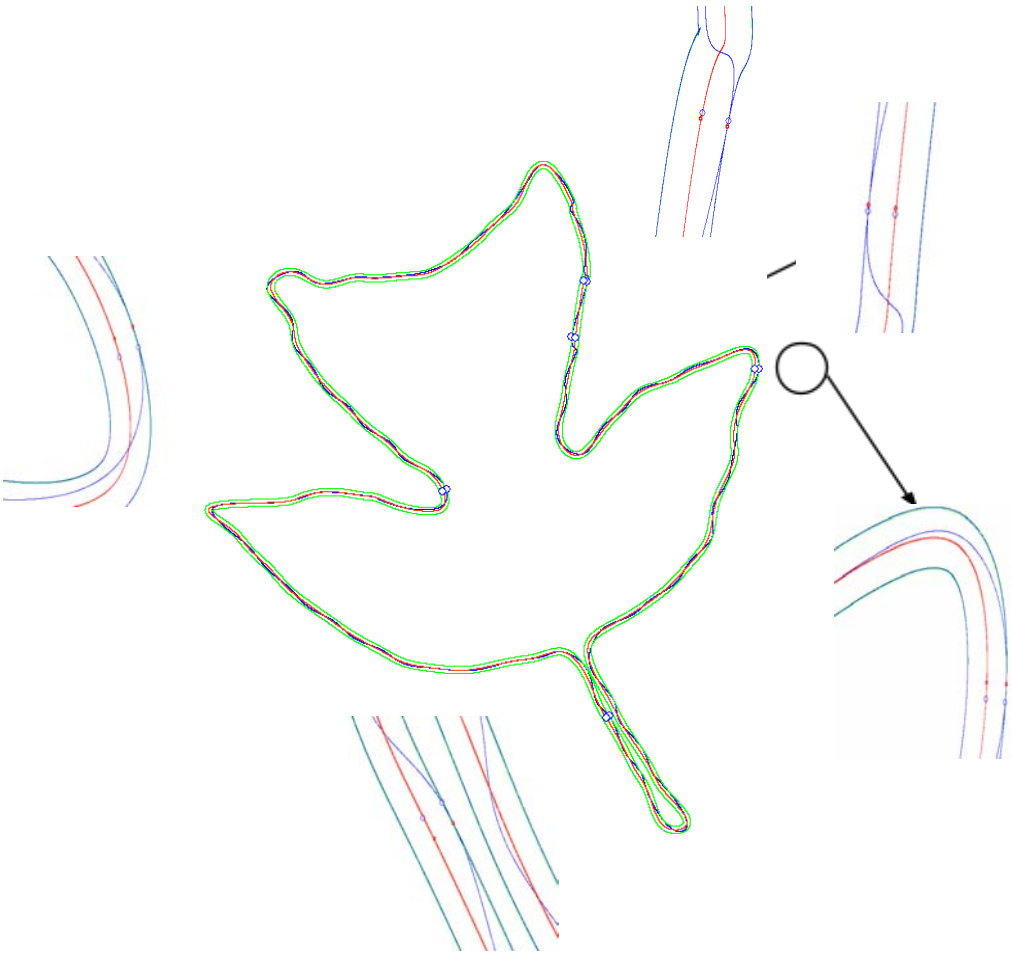


Fig. 4: Refinement of the Characteristic Points by using the Local Optimization Approaches

III. ROUNDNESS ERROR EVALUATION BASED ON THE MINIMUM ONE-SIDED HAUSDORFF DISTANCE

2.1 Introduction

In addition to being applied to the aforementioned pattern recognition and curve matching, the mathematical programming model based on the minimum one-sided Hausdorff distance can also be used in the geometric error evaluation. The relevant geometric error evaluation model is simpler and improves computational efficiency.

As the most common fits, cylindrical fits are reported to account for 80-85% of engineering applications. The measurement and evaluation of cylindrical parts are important technical approaches to ensure part precision. Currently, roundness errors are often evaluated using the minimum circumcircle, maximum incircle, least squares, and minimum area methods[19]. Among them, the minimum area method obtains the smallest unique evaluation results, which is consistent with the roundness error definition in ANSI[20], ISO[21], and national standards[22]. As per the minimum area evaluation method definition, two ideal concentric circles are used to contain the actual measured elements and the minimum width of the containment domain can be used to represent the roundness error. In the above evaluation process, the centre position and radius of the ideal circle can be changed. The principle for evaluating the roundness error with the minimum area method is that when the measured contour is contained by two concentric circles, at least four measured points are distributed inside and outside of two concentric circles. Since the specific algorithm used to obtain the minimum roundness error is not specified in any of the above standards[20-22], some researchers have proposed

geometric error evaluation models and algorithms since the publication of these standards. In some existing studies[23-25] the roundness error evaluation is regarded as a double inclusion evaluation according to the criteria of the minimum area evaluation method. In other words, there are two ideal elements that contain the actual measured elements. Based on that, the following programming algorithm can be constructed;

$$\begin{aligned} \min_{\mathbf{u}\in R^2} \quad & \xi_1 - \xi_2 \\ s.t. \quad & \sqrt{(x_i-u_1)^2+(y_i-u_2)^2} \le \xi_1 \\ & \sqrt{(x_i-u_1)^2+(y_i-u_2)^2} \ge \xi_2 \\ & i=1,2,\ldots,n \end{aligned} \tag{21}$$

where, $[x_i, y_i]^{\mathrm{T}}$ represents the coordinates of a measurement point and n represents the number of measurement points; $\mathbf{u}=[u_1,u_2]^{\mathrm{T}}$ represents the centre of an ideal circle; ξ_1,ξ_2 represents the radius of the maximum and minimum containment circles. If $\xi_1 = R+\zeta_1, \xi_2 = R+\zeta_2$ is arranged, R represents the radius of an ideal circle and ζ_1,ζ_2 represents the maximum and minimum normal errors. On that basis, the model from Equation (21) can be transformed into the saddle point programming model provided by a previous study[19],

$$\begin{aligned} \min_{\mathbf{u}\in R^2} \quad & \zeta_1 - \zeta_2 \\ s.t. \quad & \zeta_1 \ge \sqrt{(x_i-u_1)^2+(y_i-u_2)^2} - R \\ & \zeta_2 \le \sqrt{(x_i-u_1)^2+(y_i-u_2)^2} - R \\ & i=1,2,\ldots,m \end{aligned} \tag{22}$$

where, ζ_1,ζ_2 respectively represent the maximum and minimum normal errors from the point on the actual measured element to the ideal element.

The roundness error evaluation model based on the minimum one-sided Hausdorff distance is explored in the following section. Finally, the computation precision and efficiency are compared with those in Equations (21) and (22) using numerical examples.

2.2 Roundness Error Evaluation Programming Model Based on the Minimum One-Sided Hausdorff Distance

If the measured actual contour is presented as Curve A and the ideal element is presented as Curve B, the one-sided Hausdorff distance $h(A,B)$ from Curve A to Curve B represents the maximum deviation from Curve A to Curve B. During the roundness error evaluation, the centre position and radius of the ideal element B can be changed and the relevant vector equation can be expressed as:

$$\mathbf{B}(b,\mathbf{x}) = (X + R\cos b)\mathbf{i} + (Y + R\sin b)\mathbf{j} \tag{23}$$

where, $\mathbf{x}=[X,Y,R]^{\mathrm{T}}$ represents the transformation parameter vector and its components represent the circle centre coordinates and radius of the ideal element B.

From the perspective of the BUA, the one-sided Hausdorff distance between two geometric objects is the maximum deviation of the approximation object from the approximation target. The optimisation objective of the BUA is to minimise the maximum deviation under a transformation. If the approximation object scale and position parameters are selected as optimisation variables and the minimum one-sided Hausdorff distance from the approximate object to the approximation target is selected as the optimisation objective, the following roundness error evaluation programming model can be obtained after Curve A is discretised:

$$\begin{aligned} &\min_{\mathbf{x}} \quad f(\mathbf{y}) = h \\ &s.t. \quad G_i(\mathbf{y}) = d_i(\mathbf{A}(a_i), \mathbf{B}(b_i, \mathbf{x})) - h \le 0 \qquad i = 1, 2, ..., m \end{aligned} \tag{24}$$

where, $d_i(\mathbf{A}(a_i), \mathbf{B}(b_i, \mathbf{x}))$ represents the shortest distance from the discrete point a_i on Curve A to the ideal circle $\mathbf{B}(b, \mathbf{x})$ after transformation; $b_i = b_i(a_i)$ represents the curve parameter corresponding to the minimum normal distance mapping mapping point of the discrete point a_i on the ideal circle $\mathbf{B}(b, \mathbf{x})$; i represents the discrete point serial number, and m represents the total number of discrete points on Curve A. $\mathbf{x} = [X, Y, R]^{\mathrm{T}}$ represents the similarity transformation parameter and $\mathbf{y} = [h, \mathbf{x}]^{\mathrm{T}} = [h, X, Y, R]^{\mathrm{T}}$ represents the programming model optimisation variables.

It can be revealed that the optimisation model constraint function given in Equation (24) is not a linear combination of optimisation variables, but a nonlinear optimisation which needs to be linearised.

2.3 Linear Solution Method

It can be seen from the above optimisation model that the nonlinear part of the optimisation variable is included in the shortest distance d_i. If the shortest distance in Equation (24) is regarded as a function of the transformation parameter $\mathbf{x}$, it can be expressed as follows:

$$d_i = d_i(\mathbf{A}(a_i), \mathbf{B}(b_i, \mathbf{x})) = \left\| \mathbf{A}(a_i) - \mathbf{B}(b_i, \mathbf{x}) \right\| = g_i(\mathbf{x}) = g_i(X, Y, R) \tag{25}$$

If the distance function $d_i = g_i(\mathbf{x})$ is subject to Taylor's expansion at the transformation parameter zero point $\mathbf{x}_0 = (X_0, Y_0, R_0)^{\mathrm{T}}$ and their second-order and higher-order small quantities are omitted, the following expression can be obtained;

$$g_i(\mathbf{x}) = g_i(\mathbf{x}_0) + \frac{\partial g_i}{\partial X}\Delta X + \frac{\partial g_i}{\partial Y}\Delta Y + \frac{\partial g_i}{\partial R}\Delta R \tag{26}$$

In the equation,

$$g_i(\mathbf{x}_0) = \left\| \mathbf{A}(a_i) - \mathbf{B}(b_i, \mathbf{x}) \right\|_{\mathbf{x}=\mathbf{x}_0} \tag{27}$$

$$\frac{\partial g_i}{\partial X} = -\frac{\left[\mathbf{A}(a_i) - \mathbf{B}(b_i, \mathbf{x})\right] \cdot \mathbf{B}'_x(b_i, \mathbf{x})}{\left\| \mathbf{A}(a_i) - \mathbf{B}(b_i, \mathbf{x}) \right\|}\Bigg|_{\mathbf{x}=\mathbf{x}_0} = -\mathbf{N}_i \cdot \mathbf{i} = -N_{xi} \tag{28}$$

$$\frac{\partial g_i}{\partial Y}=-\frac{\left[\mathbf{A}(a_i)-\mathbf{B}(b_i,\mathbf{x})\right]\cdot\mathbf{B}'_y(b_i,\mathbf{x})}{\left\|\mathbf{A}(a_i)-\mathbf{B}(b_i,\mathbf{x})\right\|}\Bigg|_{\mathbf{x}=\mathbf{x}_0}=-\mathbf{N}_i\cdot\mathbf{j}=-N_{yi} \tag{29}$$

$$\frac{\partial g_i}{\partial R}=-\frac{\left[\mathbf{A}(a_i)-\mathbf{B}(b_i,\mathbf{x})\right]\cdot\mathbf{B}'_R(b_i,\mathbf{x})}{\left\|\mathbf{A}(a_i)-\mathbf{B}(b_i,\mathbf{x})\right\|}\Bigg|_{\mathbf{x}=\mathbf{x}_0}=-\mathbf{N}_i\cdot(\cos b_i\mathbf{i}+\sin b_i\mathbf{j})=\pm1 \tag{30}$$

where, $g(\mathbf{x}_0)$ represents the shortest distance from a point a_i on Curve A to the ideal circle B before the transformation; $\Delta X,\Delta Y,\Delta R$ represents the first-order increment of the transformation parameter X,Y,R at the zero point $\mathbf{x}_0=(X_0,Y_0,R_0)^{\mathrm{T}}$; N_{Xi},N_{Yi} represents the coordinates of the unit normal vector at a point on the ideal curve; $\mathbf{B}'_x,\mathbf{B}'_y,\mathbf{B}'_R$ represents the first-order partial derivative of Equation (23) for the transformation parameter X,Y,R.

After Equations (26) - (30) are substituted into Equation (24), the following linear programming model to compute the minimum one-sided Hausdorff distance between plane curves can be obtained:

$$\begin{aligned}\min_{\mathbf{x}}\quad & f(\mathbf{y})=h\\ s.t.\quad & G_i(\mathbf{y})=g_i(\mathbf{x}_0)-N_{xi}\Delta X-N_{yi}\Delta Y\pm\Delta R-h\le 0\\ & i=1,2,\ldots,m\end{aligned} \tag{31}$$

where, h represents the characteristic parameter, namely the one-sided Hausdorff distance from Curve A to Curve B and m represents the number of discrete points on the actual curve.

2.4 Optimal Conditions for the Minimum Hausdorff Distance

In terms of the programming model given in Equation (24), the conditions that should be satisfied at the optimal solution point are called optimality conditions. According to the optimisation theory[26-27], the optimality condition in nonlinear programming is the K-T condition. If $\mathbf{Y}^*$ is the model local optimal solution, it is required to have the following K-T condition;

$$\left.\begin{aligned}&\nabla f(\mathbf{Y}^*)+\sum_{i\in I(\mathbf{X}^*)}\lambda_i^*\nabla G_i(\mathbf{Y}^*)=0\\ &\lambda_i^*\ge 0\qquad i\in I(\mathbf{Y}^*)\end{aligned}\right\} \tag{32}$$

where, $I(\mathbf{Y}^*)=\left\{i\middle|G_i(\mathbf{Y}^*)=0,1\le i\le m\right\}$ is termed a tight constraint index set and the constraint ($G_i(\mathbf{Y}^*)=0$) corresponding to the element (i) in the set is termed a tight constraint.

The following expressions can be obtained from Equation (31);

$$\nabla f(\mathbf{Y}^*)=\begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}^T \tag{33}$$

$$\nabla G_i(\mathbf{Y}^*)=\begin{bmatrix}-1 & \dfrac{\partial g}{\partial X} & \dfrac{\partial g}{\partial Y} & \dfrac{\partial g}{\partial R}\end{bmatrix} \tag{34}$$

After Equations (33) and (34) are substituted into the K-T condition equation (32), it can be seen that:

$$\left.\begin{aligned}&\sum_{i=1}^{4}\lambda_i = 1\\&\sum_{i=1}^{4}\lambda_i \mathbf{b}_i = 0\\&\sum_{i=1}^{4}\lambda_i \mathbf{N}_i \cdot \mathbf{e}_i = 0\end{aligned}\right\} \qquad (35.a.b.c)$$
$$\lambda_i \geq 0$$

where, $\mathbf{b}_i = \begin{bmatrix} N_{xi} & N_{yi} \end{bmatrix}^T$ represents the vector corresponding to the tight constraint; $\mathbf{e}_i = \cos b_i \mathbf{i} + \sin b_i \mathbf{j}$.

It is evident from previous analysis that if the characteristic point falls on the outer containment circle B_o, it can be taken that $\mathbf{N}_i \cdot \mathbf{e}_i = 1$; If the characteristic point falls into the inner containment circle B_i, it can be taken that $\mathbf{N}_i \cdot \mathbf{e}_i = -1$. When $\lambda_i \geq 0$ ($i = 1,2,3,4$), it can be found from Equation (35.c) that the 4 characteristic points cannot completely fall on the outer containment circle B_o or the inner containment circle B_i. This indicates that there is no characteristic point with a distribution ratio of 0:4 (4:0).

In a situation where the characteristic points' distribution ratio is 1:3 (3:1), it is safe to assume that the first characteristic point is on B_o and the other three characteristic points are on B_i. It can be obtained from Equations (35.a) and (35.c) that:

$$\lambda_1 = \lambda_2 + \lambda_3 + \lambda_4 = \frac{1}{2} \qquad (36)$$

When Equation (36) is substituted into Equation (35.b), it can be taken that:

$$\mathbf{b}_1 = \frac{1}{\lambda_1}(\lambda_2 \mathbf{b}_2 + \lambda_3 \mathbf{b}_3 + \lambda_4 \mathbf{b}_4) \qquad (37)$$

The geometric meaning of the above equation is that the convex combination of characteristic points 2, 3, and 4 shall contain the vector $\mathbf{b}_1$ corresponding to the characteristic point 1. Since the 4 characteristic points are distinct tight constraint points, the case represented by Equation (37) cannot exist, specifically that there is no characteristic point with a distribution ratio of 1:3 (3:1).

Consequently, the characteristic points distribution ratio of the roundness error, evaluated by the minimum area method on the inner and outer containment circles B_i and B_o, can only be 2:2. Accordingly, the 1st and 2nd characteristic points can be arranged on B_o and the other 2 characteristic points can be arranged on B_i. It can be deduced from Equation (35.b) that:

$$\lambda_1 + \lambda_2 = \lambda_3 + \lambda_4 = \frac{1}{2} \qquad (38)$$

Under this circumstance, it can be taken that

$$\lambda_1 \mathbf{b}_1 + \lambda_2 \mathbf{b}_2 = \lambda_3 \mathbf{b}_3 + \lambda_4 \mathbf{b}_4 \qquad (39)$$

As suggested in Equation (39), when the roundness error evaluation results are optimal, the characteristic points on the inner and outer containment circles B_i and B_o are distributed as

intersecting lines, which is consistent with the roundness error evaluation criteria in the minimum area method.

The K-T condition provides the necessary conditions for the optimal constrained nonlinear optimisation solution. When the optimisation objective and constraint function in the optimisation model are convex functions, the locally optimal solution is the global optimal solution. According to the optimisation theory[26-27], linear programming belongs to convex programming. Meanwhile, the linear programming model can guarantee higher precision under the assumption of small errors. Hence, the minimum condition given in Equation (35) can be regarded as a necessary and sufficient condition for reaching the global minimum.

2.5 Numerical Examples

Numerical example 1

The measured data is quoted from a previous study[28], as presented in Table 1. The roundness errors in the study are as follows: The centre coordinates and the radius of the ideal circle would be $\left[X^*, Y^*, R^*\right]^{\mathrm{T}} = \left[40.0007, 50.0015, 30.0000\right]^{\mathrm{T}}$ mm and the roundness error is 29.2816 μm. Based on the data in Table 1, the conventional roundness error model (Equation 21) and the Hausdorff distance-based model (Equation 31) are compared in terms of their computation precision and efficiency.

The average value of the coordinates of 25 data points listed in Table 1 is selected as the initial circle centre coordinates $\left[X_0, Y_0\right]^{\mathrm{T}}$. The average distance from each measurement point to the initial circle centre coordinates is selected as the initial radius of the ideal circle R_0. A computer with a Pentium IV processor with a frequency of 2.8GHz and a 256MB memory is adopted during this process. Based on the model in Section 1.1, when the computation precision ε is 10^{-10}, the centre coordinates and the radius of the ideal circle can be obtained as $\left[X^*, Y^*, R^*\right]^{\mathrm{T}} = \left[40.000739, 50.001530, 30.000063\right]^{\mathrm{T}}$ mm.

The minimum one-sided Hausdorff distance is $h_T(A, B) = 14.640087$μm, the roundness error is 29.280175μm, the high characteristic points are the 11th and 18th points, and the low characteristic points are the 6th and 13th points. The average time for ten computations is 0.32ms. The results (circle centre coordinates, roundness errors, and characteristic points) from the model given in Equation (21) are similar to those computed by the model based on the minimum one-sided Hausdorff distance in Section 1.1. The radii of the maximum and minimum containment circles are 30.014702620mm and 29.985422445mm, respectively, but the average time for ten computations based on the latter is 1.73ms, which is about 5.4 times that based on the former.

Tab. 1: Initial data of Example 1

Serial number of measure points	X-coordinate /mm	Y-coordinate/mm
1	70.0150	50.0000
2	68.7900	58.4734
3	65.4060	65.9372
4	59.5675	72.7493
5	51.3791	77.7452
6	44.7944	79.6013
7	40.8903	79.9958
8	32.0312	78.9306
9	27.2296	77.1385
10	20.3993	72.7076
11	16.1556	68.2304
12	12.7184	62.4905
13	10.6380	56.0806
14	10.0183	49.2149
15	11.4275	40.8264
16	14.1050	34.8682
17	18.8168	28.7427
18	24.6321	24.2200
19	31.6833	21.1862
20	39.1626	20.0207
21	45.5204	20.5021
22	55.3996	24.2692
23	62.3561	30.0114
24	67.3540	37.6492
25	69.6190	45.2028

Numerical Example 2

The measured data is quoted from a previous study[25], as listed in Table 2. The computed results based on the steepest descent algorithm in that study are presented as follows: The ideal circle centre coordinates would be $\left[X^*, Y^*\right]^{\mathrm{T}} = \left[82.990941,\ 97.008387\right]^{\mathrm{T}}$ mm and the roundness error is 38.231μm.

The initial centre coordinates and the radius of the ideal circle are constructed via the same method as Example 1. When the computation precision ε is 10^{-10}, the centre coordinates and radius of the ideal circle computed based on the minimum one-sided Hausdorff distance model would be $\left[X^*, Y^*, R^*\right]^{\mathrm{T}} = \left[82.990941, 97.008387, 30.029726\right]^{\mathrm{T}}$ mm, the minimum one-sided Hausdorff distance is $h_T(A, B) = 19.115472$m, and the roundness error is 38.230944μm. The 8th and 20th points are the high characteristic points and the 1st and 16th points are the low characteristic points. The average time for ten computations is 0.28ms. The results (circle centre coordinates, roundness errors, and characteristic points) computed by the model given in Equation (21) are similar to those computed by the model based on the minimum one-sided Hausdorff distance. The radii of the maximum and minimum containment circles are 30.048841860mm and 30.010610916mm, respectively, but the average time for ten computations based on the latter is 1.64ms, which is about 5.9 times that based on the former.

Tab. 2: Initial Data of Example 2

Serial number of measure points	X-coordinate /mm	Y-coordinate /mm
1	107.5811	114.2119
2	102.2909	119.9906
3	95.6848	124.2034
4	88.2128	126.5634
5	80.3826	126.9159
6	72.7251	125.2311
7	65.7612	121.6196
8	59.9721	116.3233
9	55.7576	109.7039
10	53.4073	102.2180
11	53.0774	94.3816
12	54.7849	86.7302
13	58.4107	79.7824
14	63.7075	74.0083
15	70.3176	69.8019
16	77.7899	67.4519
17	85.6152	67.1081
18	93.2669	68.7926
19	100.2245	72.4009
20	106.0093	77.6929
21	110.2199	84.3073
22	112.5676	91.7864
23	112.8977	99.6156
24	111.2129	107.2695

IV. CONCLUSION

1. In this study, a two-stage (rough and accurate) computation strategy was proposed to achieve the accurate computation of the minimum one-sided Hausdorff distance between continuous plane curves under a similarity transformation. A programming model for computing the minimum one-sided Hausdorff distance was developed for when the curve is provided in the form of measurement points or extremely high computation precision is not required. In addition, a programming model linearisation method was also formulated and a stable and efficient simplex method[26] was adopted to obtain solutions. Under the conditions of continuous curves with the requirement to precisely solve the minimum one-sided Hausdorff distance, a local iterative precision algorithm, based on the programming algorithm, was established to separate the similarity transformation parameters from the curve parameters corresponding to characteristic points. Subsequently, the minimum one-sided Hausdorff distance between continuous plane curves can be accurately computed.
2. The geometrical meaning of the one-sided Hausdorff distance between plane curves and the minimum one-sided Hausdorff distance under a similarity transformation was explored. The geometrical denotation $h(A,B)$ is that curve A is contained by two symmetrical equidistant curves of curve B. When the inner and outer symmetrical equidistant curves completely contain curve A with the narrowest width, half of the width value is called the one-sided Hausdorff distance from curve A to curve B. Under a similarity transformation, the geometric meaning $h_{\mathrm{T}}(A,B)$ is that when the width of the symmetric containment domain of curve B reaches the minimum, there are at least 5 tangency points between curve A and the inner and outer symmetric equidistant curves of

Curve B. The tangency points distribution ratio on the two equidistant curves can be 1:4 (4:1) or 2:3 (3:2). The maximum and minimum normal errors of curve A relative to curve B are equal in value and opposite in sign.

3. A mathematical programming model based on the minimum one-sided Hausdorff distance was applied to roundness error evaluation. Based on the local optimum condition (the K-T condition) of constrained optimisation in optimisation theory, it can be demonstrated that the minimum condition of one-sided Hausdorff distance from the actual curve to the ideal circle is equivalent to the roundness error evaluation criteria in the minimum area method specified in international and national standards. Compared with the conventional roundness error evaluation model based on the minimum area method, the optimisation objective function is the radius difference between the maximum and minimum containment circles and the number of constraint functions in the model based on the minimum one-sided Hausdorff distance is only half that in the conventional model. Therefore, the model constructed in this study can significantly improve the roundness error evaluation efficiency. Furthermore, this method can also be applied to the evaluation of other geometric errors by the minimum area method.

Declaration

Funding: This work was supported by Natural Science Foundation of Shanghai, and the Award Number is 15ZR1417200.

Conflicts of interest: We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.
Availability of data and material: Not applicable

Code availability: Not applicable
Authors' contributions: Improve processing efficiency and machining precision of centrifugal 3D impeller
Ethics approval: Not applicable
Consent to participate: Yes
Consent for publication: Yes

REFERENCES

1. Alt H., Scharf L. Computing the Hausdorff distance between curved objects[J]. International Journal of Computational Geometry & Applications , 2008,(18):307–320.
2. Elber G., Grandine T. Hausdorff and minimal distances between parametric freeforms in R2 and R3 [J]. Lecture Notes in Computer Science, 2008,4975:191–204.
3. Chen X.D., Ma W.Y., Xu G., et al. Computing the Hausdorff distance between two B-spline curves[J]. Computer-Aided Design, 2010,(42):1197–1206.
4. Kim Y.J., Oh Y.T., Yoon S.H., et al. Precise Hausdorff distance computation for planar freeform curves using biarcs and depth buffer[J]. The Visual Computer, 2010,(26):1007–1016.
5. Bai Y.B., Yong J.H., Liu C.Y., et al. Polyline approach for approximating Hausdorff distance between planar free-form curves[J]. Computer-Aided Design, 2011,43 (6) :687–698.
6. Huttenlocher D.P., Kedem K. Computing the minimum Hausdorff distance for point sets under translation[C]. Proceedings of the sixth ACM symposium on computational geometry, 1990:340-349.
7. Huttenlocher D.P., Kedem K., Sharir M. The upper envelope of Voronoi surface and its applications[J]. Discrete Computational Geometry, 1993, 9 (1):267-291.
8. Alt H., Behrends B., Blomer J., et al. Approximate matching of polygonal shapes[J]. Annals of Mathematics and Artificial Intelligence, 1995,(13):251–265.

9. Chew L.P., Goodrich M.T., Huttenlocher D.P., et al. Geometric pattern matching under Euclidean motion[J]. Computational Geometry, 1997,7 (1-2):113-124.
10. Hur S., Kim T.W. Finding the best conic approximation to the convolution curve of two compatible conics based on Hausdorff distance[J]. Computer-Aided Design,2009,41 (7):513-524.
11. Gu Y.H., Tjahjadi T. Coarse-to-fine planar object identification using invariant curve features and B-spline modeling[J]. Pattern Recognition, 2000,33 (9):1411-1422.
12. Liang B.J., Wallace A.M. Viewpoint independent matching of planar curves in 3D space[J]. Pattern Recognition, 2004,37 (3):525-542.
13. Gruen A., Akca D. Least squares 3D surface and curve matching[J]. ISPRS Journal of Photogrammetry & Remote Sensing, 2005,59 (3):151-174.
14. Rodriguez W., Last M., Kandel A., et al. Horst Bunke. 3-Dimensional curve similarity using string matching[J]. Robotics and Autonomous Systems, 2004,49 (3-4):165–172.
15. Yamany S.M., Ahmed M.N., Farag A.A. A new genetic-based technique for matching 3-D curves and surfaces[J]. Pattern Recognition, 1999,32 (10):1817-1820.
16. Cao L.X., Dong L., Cao J.J. Computation of Hausdorff distance between planar curves. Journal of Dalian University of Technology[J]. 2014,54(2):182-188.
17. Cao L.X., Qin L.J. Evaluation Method of Line Profile Error Based on Minimum Directed Hausdorff Distance[J]. Journal of Mechanical Science and Technology for Aerospace Engineering. 2015,34(10):1547-1552.
18. Piegl L., Tiller W. The NURBS Book [M]. Berlin : Springer, c1997, 2nd ed.
19. Liu J., Wang XM. Saddle point programming and geometric error evaluation [M]. Dalian: Dalian University of Technology Press, 1996.
20. ANSI/ASME Y14.5M. Dimensioning and Tolerancing[M]. New York: American National Standards Institute; The American Society of Mechanical Engineers, 1982.
21. ISO/R 1101-1983. Technical drawings—Geometrical tolerancing—Guidelines[M]. ISO, Geneva, 12–01.
22. Chinese Standard. Tolerance and Position Tolerance [M].GB1182~1184-80, GB1958-80.
23. Weber T., Motavalli S., Fallahi B., et al. A unified approach to form error evaluation[J]. Precision Engineering, 2002,26: 269–278.
24. Ding Y., Zhu L.M., Ding H. A unified approach for circularity and spatial straightness evaluation using semi-definite programming[J]. International Journal of Machine Tools & Manufacture, 2007,(47): 1646–1650.
25. Zhu L.M., Han Ding H., Xiong Y.L. A steepest descent algorithm for circularity evaluation[J]. Computer-Aided Design, 2003,(35):255-265.
26. Tang H.W. Introduction to Practical mathematical programming[M]. Dalian: Dalian University of Technology Press, 1986.
27. Donald M. Simmons. Nonlinear programming for operations research[M]. New Jersey :Prentice-Hall, c1975.
28. Samuel G.L., Shunmugam M.S. Evaluation of circularity from coordinate and form data using computational geometric techniques[J]. Precision Engineering, 2000,(24):251–263.

This page is intentionally left blank

Design, Simulation, and Real-Time Implementation of a DC Microgrid Powered by Renewable Hybrid Energy Sources

Samir Abood & Islam Khalid

A&M University

ABSTRACT

When large amounts of renewable energy sources are integrated, DC microgrids face difficulties with voltage regulation, energy management, inertia control, and uncertainty management. Numerous approaches have been offered to address these difficulties, such as droop control, centralized control, distributed control, virtual inertia control, and uncertainty management algorithms. Despite these efforts, a comprehensive overview that offers a panoramic understanding of the existing techniques and forthcoming trends in controlling renewable energy-integrated DC microgrids has been lacking. This paper involves designing, simulating, and implementing a DC Microgrid based on hybrid renewable energy sources to achieve high efficiency and sustainability in energy systems. The research focuses on integrating solar and wind energy sources with the design of an advanced energy management system that ensures network stability and reduces electrical losses.

Keywords: NA

Classification: DDC Code: 621.312136

Language: English

Great Britain Journals Press

LJP Copyright ID: 392912
Print ISSN: 2631-8474
Online ISSN: 2631-8482

London Journal of Engineering Research

Volume 25 | Issue 1 | Compilation 1.0

Design, Simulation, and Real-Time Implementation of a DC Microgrid Powered by Renewable Hybrid Energy Sources

Samir Abood[α] & Islam Khalid[σ]

ABSTRACT

When large amounts of renewable energy sources are integrated, DC microgrids face difficulties with voltage regulation, energy management, inertia control, and uncertainty management. Numerous approaches have been offered to address these difficulties, such as droop control, centralized control, distributed control, virtual inertia control, and uncertainty management algorithms. Despite these efforts, a comprehensive overview that offers a panoramic understanding of the existing techniques and forthcoming trends in controlling renewable energy-integrated DC microgrids has been lacking. This paper involves designing, simulating, and implementing a DC Microgrid based on hybrid renewable energy sources to achieve high efficiency and sustainability in energy systems. The research focuses on integrating solar and wind energy sources with the design of an advanced energy management system that ensures network stability and reduces electrical losses. A comprehensive mathematical model was developed to study network dynamics and analyze its response to operating conditions, such as climate and load changes. High-precision numerical simulation techniques were adopted to analyze the performance and improve the compatibility between the network components to achieve maximum operational efficiency. The research includes designing and implementing a controller to manage the real-time dynamic balance between energy generation and consumption. Practical tests of the proposed system were conducted in realistic operating environments to study the network's performance in multiple cases, such as peak loads and sudden outages.

Author α: Department of Electrical and Computer Engineering Department, Prairie View A&M University, USA.

σ: University of Diyala Engineering College, Diyala, Iraq.

I. INTRODUCTION

Environmentally friendly and sustainable alternatives to conventional fossil fuels encompass solar, wind, hydro, and geothermal energy sources. Their significance is in reducing greenhouse gas emissions and decreasing our dependency on scarce resources to mitigate the negative impacts on climate. However, the intermittent nature of renewable energy sources (RESs) like solar and wind has challenges, necessitating improved grid stability and effective energy storage integration. A comprehensive strategy incorporating technological innovation, helpful policies, and public involvement is required to overcome these challenges and establish a sustainable energy future. DC microgrids are very important for integrating renewable energy sources like solar panels because they manage the direct current (DC) output from these sources well, reducing conversion losses. Implementing this streamlined strategy improves overall energy efficiency, making it suitable for localized power distribution networks and facilitating the integration of diverse renewable sources. A DC Microgrid is a small-scale power system designed to manage the distribution and consumption of electrical energy locally using direct current (DC). It is considered an efficient and flexible option for reliable energy supply, especially in remote areas or applications relying on renewable energy sources. The rising integration of renewable energy sources and the increasing complexity of power distribution

London Journal of Engineering Research

systems have underscored the significance of hybrid AC/DC microgrids in contemporary energy infrastructure. However, these microgrids face simultaneous challenges of preserving frequency stability and reaching economic efficiency. Conventional solutions to these challenges independently handle energy storage optimization and control strategies, leading to suboptimal microgrid performance. To address these issues, this work proposes a novel unified framework that integrates energy storage optimization with sophisticated control algorithms in a synergistic way. This research aims to provide a comprehensive solution that enhances the operational stability and financial performance of hybrid AC/DC microgrids by optimizing energy storage deployment and balancing dynamic interactions between AC and DC subsystems. The proposed unified framework has the potential to significantly advance sustainable energy systems and pave the way for future developments in microgrid technology.

Challenges of DC Microgrid Voltage fluctuations are caused by varying loads and energy sources with measurement of balanced energy distribution among different loads, which need device integration for standardization among microgrid components to decrease the initial cost and higher infrastructure cost compared to traditional grids. In this case study, we will focus on two sources. The first type we will discuss is solar power and how it was the foundation of the study. The second type would be wind turbine power. In both cases, A DC power source (solar, wind) produces and maintains a constant voltage converted to AC power. Since we know DC power does not have a loss, converting that power to AC becomes a challenge. Having the ability to create a balance between voltage regulation and current sharing while reducing current circulation.

1.1 Contributions

Combining multiple energy sources improves grid stability overall. This approach allows either source to add the needed power if one source is not producing at the rated level. Increased energy output is significant due to both sources generating electricity simultaneously. This makes the system more efficient and robust compared to a stand-alone source. Load capacity is increased with a combined system since more electricity is being produced. Cost is reduced by using free fuel sources such as the sun and wind to power the systems.

II. METHODOLOGY

The system's performance in the most unfavorable conditions. Energy storage devices are essential for reducing variations in renewable energy production and improving the stability of the system. Artificial intelligence methodologies, such as machine learning and neural networks, assist in predicting renewable energy production and electricity consumption, thereby enabling proactive decision-making. Furthermore, uncertainty modeling techniques such as probabilistic and possibilistic methodologies allow for the measurement and examination of uncertainties related to renewable energy sources, grid components, and demand patterns. This ensures that microgrids can operate reliably in dynamic contexts. Several recent reviews summarized in The academic literature underscore a significant gap in current research regarding the control, energy management, inertia support, uncertainty management of DC microgrids, and economic and environmental benefits, particularly within the framework of integrating renewable energy sources. While numerous studies delve into these aspects individually, a comprehensive systematic review that encompasses all aspects of DC microgrids is notably lacking. To bridge these gaps, this review provides comprehensive documentation on voltage control, uncertainty management, inertia support, environmental and economic benefits, and energy management of large-scale renewable energy-integrated DC microgrids, along with specific and comprehensive recommendations aimed at supporting greater integration of renewable energy sources with DC microgrids. The review encompasses cutting-edge technologies and methodologies, shedding light on emerging trends while also pinpointing areas requiring further investigation. Additionally, the article furnishes a compilation of prospective research recommendations to guide and inspire scholars in

advancing this domain. This paper has the following structure: The bibliometric analysis with a systematic approach to developing this review is discussed in Section 2, Section 3 provides a description of various major components of renewable energy-integrated DC microgrids, Section 4 provides control and energy management algorithms for DC microgrids, Section 5 documents inertia support techniques for high-level integration of renewable energy with the DC microgrids, Section 6 summarizes uncertainty issues, Section 7 discusses economic and environmental benefits of RESs integrated DC microgrids, Section 8 provides future recommendations, and finally Section 9 concludes this work. Solar cell power as a renewable energy source was unique in its perception. In this module, I used a solar panel emulator 1.5kw, 500V (CO3208-1P), equivalent to the Solar module with solar altitude emulator (CO3208-1B). Both modules would yield the same results. We used the solar panel emulator because the light on the solar altitude emulator was getting hot over a sustained period. Nonetheless, the setup includes a 3-phase industrial photovoltaic converter (CO3208-1N). In this work, we tested the characteristics of the solar panel emulator and the effect it would have on the solar altitude emulator by using 0,30,60,100 de-rating. The de-rating acts as the sun's movement across the solar cell panel. At each level of the de-rating, the converter adjusted to keep the voltage stable as I fluctuated from 100% to 0%. Once I ran the test and received the results that were yielded, I began to see how the system could be improved for robustness.

The next renewable energy source explored in this work was the Wind Turbine System. The wind turbine system was more difficult to navigate. It uses an asynchronous double-fed induction generator (DFIG). (CO3208-3A). It is connected to a 3-phase motor generator (SE2662-6W), a digital controller, a brake, and the Active-Servo software. (CO3636-6W / CO2663-6U). It uses a 3-phase isolating transformer (CO3203-3B), Digital meter (CO5127-3B), Power Supply (CO3212-5U), Rectifier (CO3636-3B8), Inverter (CO3636-3A8). With the generator running at 1600 rpm, the goal was to lower the voltage to tie in with the solar panel module. The rectifier and inverter were used to accomplish this goal. I used trial and error to decrease the voltage without overloading this system.

The micro-DC Grid ties together Solar and Wind Turbine renewable energy sources. Once both Solar and Wind modules output 120 volts, this DC grid used two busbars (CO3301-5R) in this work.

III. THE EQUIPMENT

The equipment used in this work is as follows:

3.1 Solar Cell Equipment

The solar panel altitude emulator simulates the changing angle of the sun in the sky is shown in Figure 1a. It is used to test the panel's power output and determine how much its results would vary based on its altitude or the sun's position.

Parameters: 10 W polycrystalline solar module, Open-circuit voltage: 26 V, Short-circuit current: 650 mA, Adjustable module inclination, Adjustable solar altitude, Adjustable solar azimuth 500 W halogen lamp (with dimmer

The transformer-less PV-inverter is designed especially for a three-phase power supply as shown in Figure 1b. It has a graphical display for visualizing energy yield values, current power, and operating parameters of the photovoltaic system. When starting the system, you must set the time, date, and year.

The mains power supply for DC, AC, and three-phase machines and synchronous machine excitation is shown in Figures 1 c and e. The power supply unit is specially designed for the operation of electrical machines.

The solar panel emulator is a controllable DC power supply source that provides the required DC voltage, current, and power to the inverter analogous to PV modules. It can be remotely operated using the "Solar Panel" virtual instrument, which allows the simulation of the behavior of a photovoltaic system.

Variable Ohmic Load CO33301-3F shown in Figure 1h has three synchronously adjustable ring

rheostats (bank winding) with scale 100 - 0% and a fuse in the sliding-contact connection. Suitable for parallel, series, star, and delta circuits, Resistance: 3 x 750 ohms, Current: 3 x 2 A

Inputs/outputs: 4mm safety sockets, Dimensions: 297 x 456 x 125 mm (HxWxD

This variable transformer CO3301-3P, shown in Figure 1f, compensates for voltage fluctuations in the event of changes in load by modifying the transformation ratio. The component can be used as a step-up or step-down transformer. Control is performed via the inputs of the power quality meter CO5127-1S.

The transformer in Figure 1 d and i are feeds the transmission line model; the scale factor is 1:1000 for secondary current and voltage.

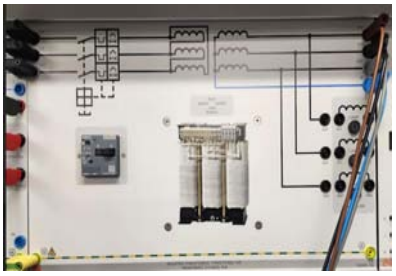
The three-phase power quality meter permits the measurement and display of all relevant grid parameters. It can carry out single, two-phase, or three-phase measurements. Display and operation are performed via menu navigation with an LC display or the integrated Ethernet interface. The optional SCADA software provides for the display of all readings and allows the implementation and analysis of intelligent power grids (smart grids). The "Smart Meter" acts as a digital electricity meter at the endpoints of the electricity grid to measure electricity consumption and can be used to turn consumers on or off, depending on circumstances.

Table 1 Shows the Specifications of the Devices used

(a) Solar Panel Altitude Emulatore

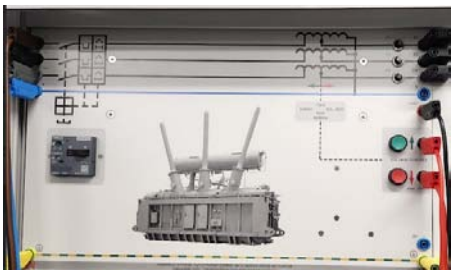
(b) PV inverter

(c) Main Power Supply



(d) Isolating 3-Phase Transformer

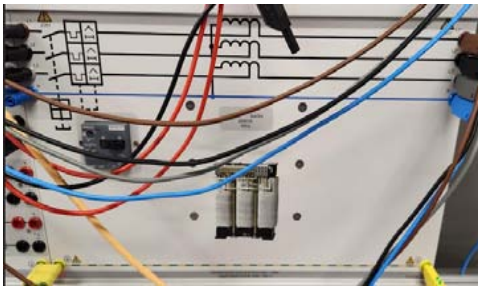
(e) Solar Panel Emulator



(f) Variable Transformer

(g) Power Quality Meter

(h) Variable Ohmic Load



(i) Station Transformer

Figure 1: Solar Cell Equipment

Table 1: Specifications of the devices used.

#	**Item**
1	DC-PV Input
2	Voltage: 250 - 1000 V
3	MPP voltage: 300 - 800 V
4	Maximum current: 11A
5	AC output
6	Voltage: 3 x 230, 50/60 Hz
7	Power factor: 0.8 – 1
8	Max. Current: 7A
9	Maximum Power: 3200 W
10	Maximum efficiency: 98.6%
11	European efficiency: 98.2%
12	MPP-efficiency: >99.7% (static), > 99% (dynamic)
13	De-rating/ power limiting occurs automatically when:
14	input power > max. recommended PV power.
15	cooling is inadequate.
16	input current is too high.
17	grid current is too high.
18	internal or external de-rating is performed.
19	grid frequency is too high (according to country setting)
20	limiting signal is received via an external interface.
21	output power is limited (set at the inverter)
22	Communication interfaces:
23	1 x RJ45 socket (RS485)
24	2 x R J45 socket (Meteocontrol WEB log or solar log; Ethernet interface)

3.2 Wind Turbine Equipment

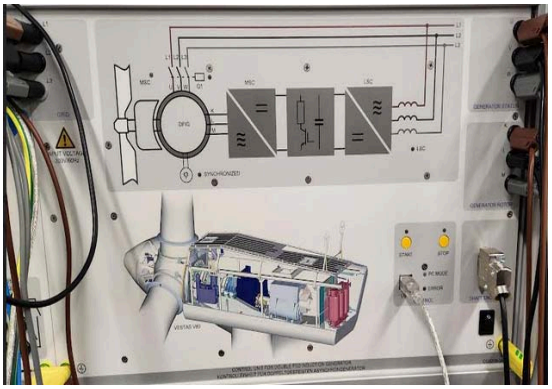
The three-phase power quality meter permits the measurement and display of all relevant grid parameters. It can carry out single, two-phase, or three-phase measurements. Display and operation are performed via menu navigation with an LC display or the integrated Ethernet interface. The optional SCADA software provides for the display of all readings and allows implementation and analysis of intelligent power grids (smart grids). The "Smart Meter" acts as a digital electricity meter at the endpoints of the electricity grid to measure electricity consumption and can be used to turn consumers on or off, depending on circumstances.

The servo-machine test bench shown in Figure 2b is a complete testing system for examining electrical machines and drives. It consists of a

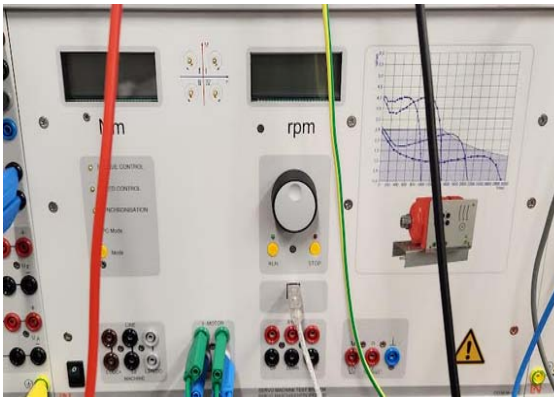
digital controller, a brake and the ActiveServo software. The system combines state-of-the-art technology with ease of operation. The system also allows manual and automated synchronization to be carried out.

Figure 1d shows a Double-Fed 3-Phase Generator and Motor. Double-fed generator 3-phase and motor. The motor and sensor leads are connected via polarity-safe plugs. The machine has thermal monitoring, and, in conjunction with the controller, it constitutes a driving and braking system that is free of drift and requires no calibration.

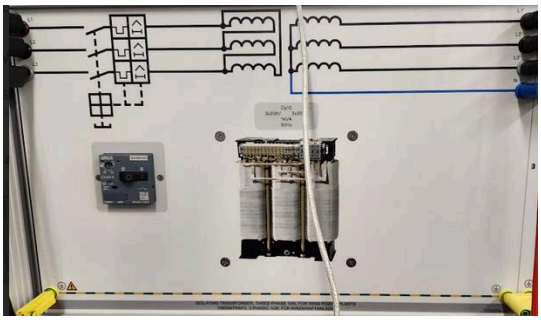
Figure 2a shows a Double-Fed Inductive Generator Control Panel. Control panel unit for double-fed Inductive generator.
Figure 2e shows a DC to AC Inverter using one transistor. The inverter uses one transistor to simulate DC to AC. I used one transistor to get this effect. Figure 2f shows an AC to DC Rectifier.
Figure 3 shows the three Phase Double Busbar, which is used as a DC busbar.



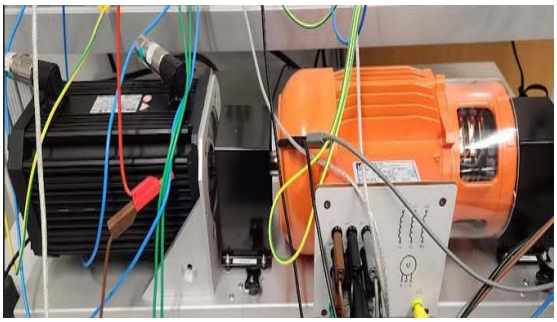
(a) DFIG



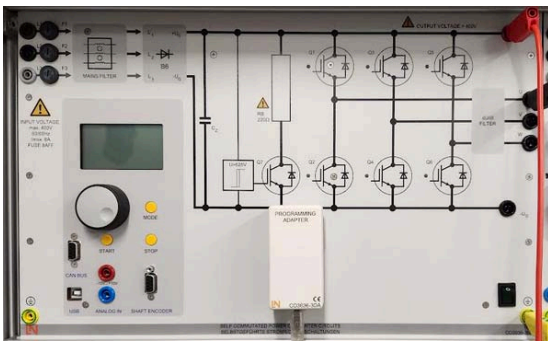
(b) Active Servo Control Panel



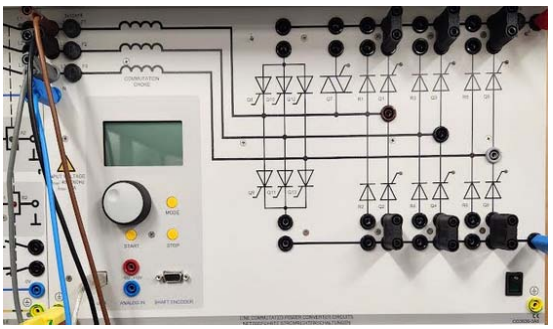
(c) Isolating 3-Phase Transformer



(d) Double-Fed 3-Phase Generator and Motor



(e) DC to AC Inverter



(f) AC to DC Rectifier

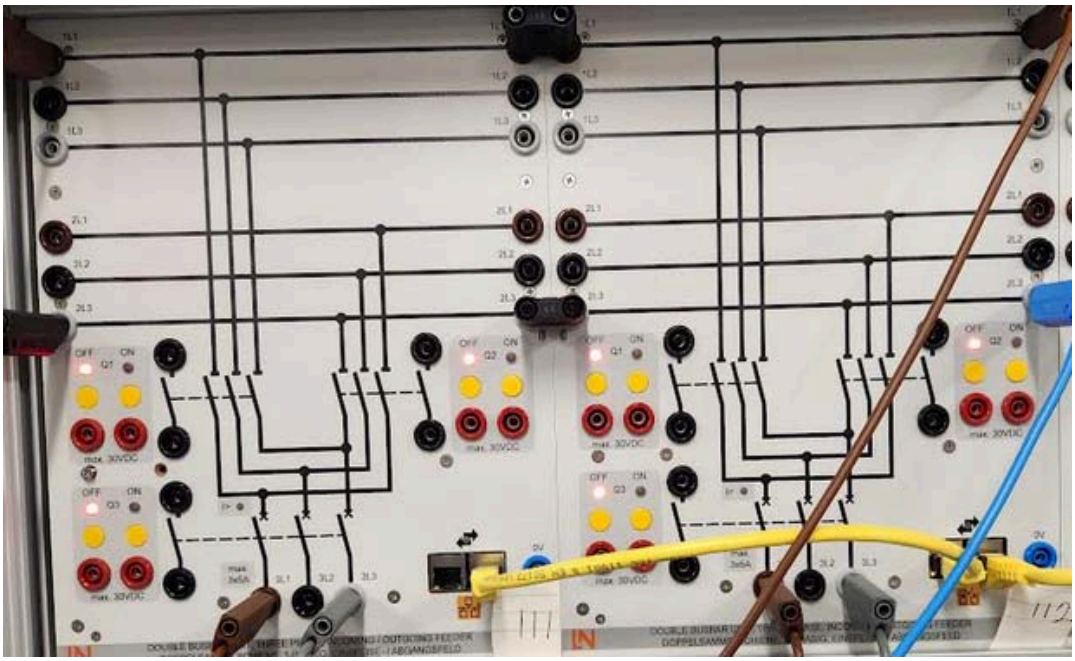


Figure 3: The three Phase Double Busbar

IV. THREE-PHASE FULL CONVERTER

DC microgrids and their components The DC microgrids function in either grid-connected mode, where the utility grid links to the shared DC bus through a bidirectional voltage source converter (VSC), or in islanded mode, operating au tenuously without utility grid connection. Solar PV and wind systems, DC loads, AC loads, fuel cells, and energy storage devices are the main components of the DC microgrids. The DC microgrids face low inertia issues due to large-scale renewable energy sources. This phenomenon is particularly pronounced in regions with high renewable energy penetration rates, where renewable energy contributes significantly to the overall electricity generation mix by replacing conventional synchronous generators. This reduced inertia challenges microgrid stability and reliability, especially during sudden changes in power demand or supply fluctuations. To tackle this, several advanced inertia support techniques, along with energy storage devices, are proposed. Notably, batteries and supercapacitors, among various energy storage devices, are frequently employed for bidirectional energy exchange with the DC bus. The grid VSC regulates DC bus voltage in grid-connected mode, whereas the battery or supercapacitor manages voltage in islanded mode. Due to its high energy density, the battery focuses exclusively on providing or absorbing steady-state power. Conversely, with its high power density, the supercapacitor offers or absorbs transient power briefly to make a power balance. In the context of DC microgrids, the utilization of a fuel cell can offer a consistent and reliable power supply, especially when integrated with sustainable energy sources such as solar and wind. The absence of combustion in its electricity generation process renders it an ecologically conscious alternative, enhancing energy diversity and overall sustainability. A three-phase full converter is a fully controlled bridge-controlled rectifier using six thyristors connected in a full-wave bridge configuration, as shown in Figure 4. All six thyristors are controlled switches turned on at appropriate times by applying suitable gate trigger signals.

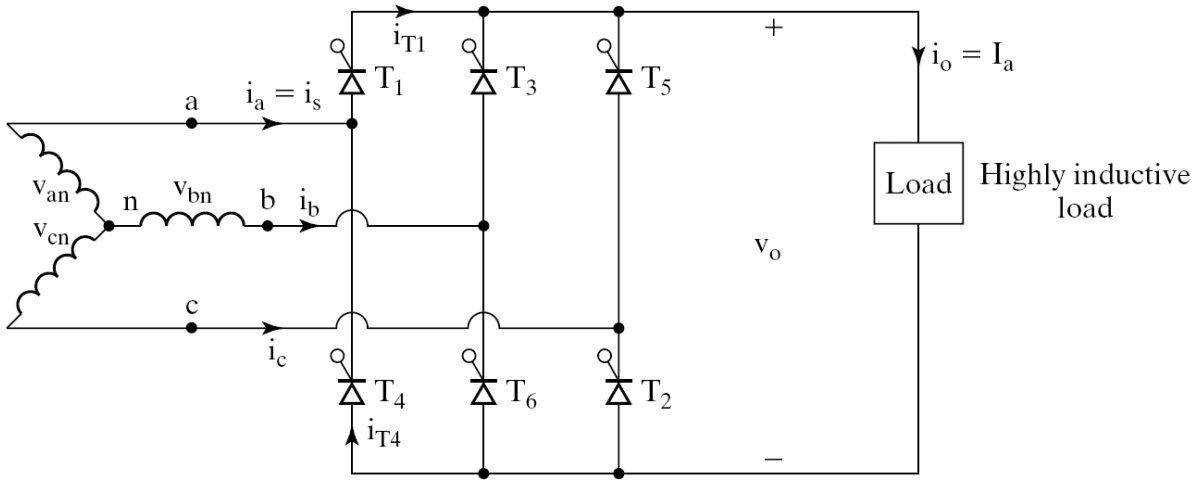


Figure 4: The three-phase full converter

At $\omega t = \left(\frac{\pi}{6} + \alpha\right)$, thyristor T_6 is already conducting when the thyristor T_1 is turned on by applying the gating signal to the gate of T_1. During the time $\omega t = \left(\frac{\pi}{6} + \alpha\right) to \left(\frac{\pi}{2} + \alpha\right)$, thyristors T_1 and T_6 Conduct together, and the line-to-line supply voltage. v_{ab} Appears across the load.

At $\omega t = \left(\frac{\pi}{2} + \alpha\right)$, the thyristor T_2 is triggered and T_6 Is reverse biased immediately, and T_6 Turns off due to natural commutation. During the period $\omega t = \left(\frac{\pi}{2} + \alpha\right) to \left(\frac{5\pi}{6} + \alpha\right)$, thyristor T_1 and T_2 Conduct together, and the line-to-line supply voltage. v_{ac} Appears across the load. Figure 5 shows the waveform of a 3-phase full converter.

The thyristors are numbered in the circuit diagram in the order in which they are triggered. The thyristors' trigger sequence (firing sequence) is 12, 23, 34, 45, 56, 61, 12, 23, and so on. The figure shows the waveforms of three-phase input supply voltages, output voltage, the thyristor current through T_1 and T_4 The supply current is through line 'a'. We define three-line neutral voltages (3-phase voltages) as follows.

$$v_{RN} = v_{an} = V_m \, sin \, \omega \, t; V_m = \text{ Max. Phase Voltage} \tag{1}$$

$$v_{YN} = v_{bn} = V_m \sin sin \left(\omega t - \frac{2\pi}{3}\right) = V_m \sin sin \left(\omega t - 120^0\right) \tag{2}$$

$$v_{BN} = v_{cn} = V_m \sin sin \left(\omega t + \frac{2\pi}{3}\right) = V_m \sin sin \left(\omega t + 120^0\right) == V_m \sin sin \left(\omega t - 240^0\right) \tag{3}$$

Where V_m Is the peak phase voltage of a star (Y) connected source? The corresponding line-to-line voltages are

$$v_{RY} = v_{ab} = \left(v_{an} - v_{bn}\right) = \sqrt{3} V_m \sin sin \left(\omega t + \frac{\pi}{6}\right) \tag{4}$$

$$v_{YB} = v_{bc} = \left(v_{bn} - v_{cn}\right) = \sqrt{3} V_m \sin sin \left(\omega t - \frac{\pi}{2}\right) \tag{5}$$

$$v_{BR} = v_{ca} = \left(v_{cn} - v_{an}\right) = \sqrt{3} V_m \sin sin \left(\omega t + \frac{\pi}{2}\right) \tag{6}$$

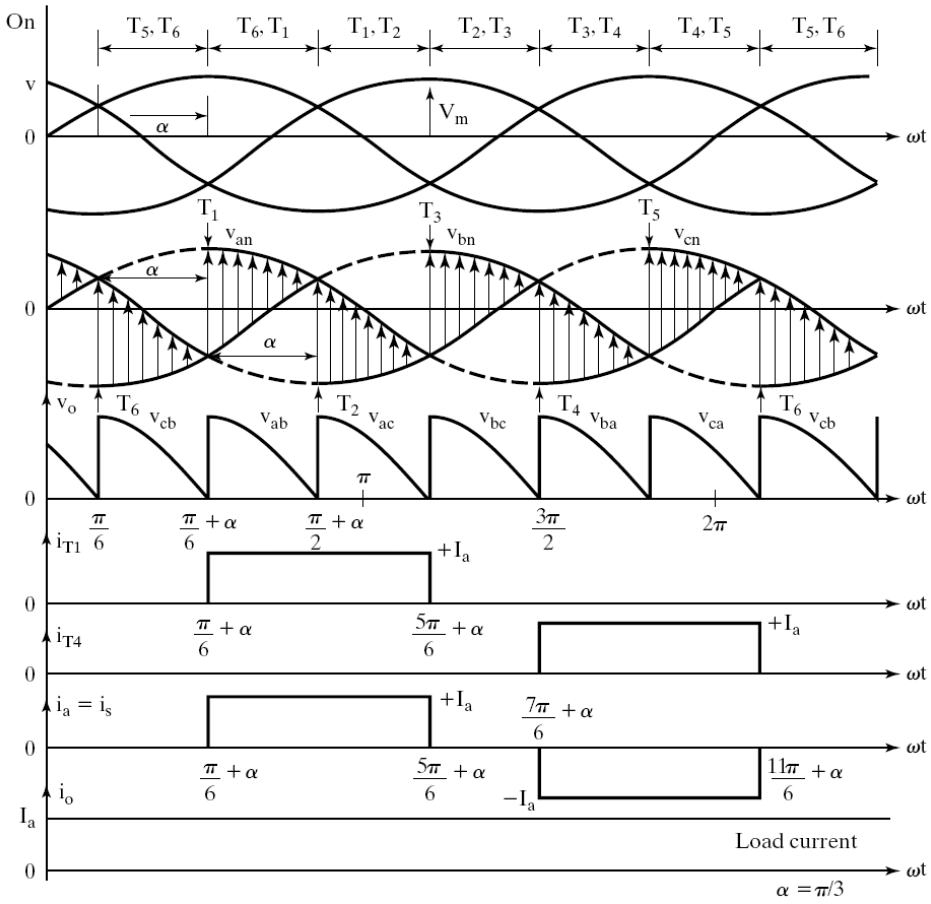


Figure 5: Waveform of 3-phase full converter

The output load voltage consists of 6 voltage pulses throughout 2π radians. Hence, the average output voltage is calculated as

$$V_{O(dc)} = V_{dc} = \frac{6}{2\pi}\int_{\frac{\pi}{6}+\alpha}^{\frac{\pi}{2}+\alpha} v_O . d\omega t \tag{7}$$

$$V_{dc} = \frac{3}{\pi}\int_{\frac{\pi}{6}+\alpha}^{\frac{\pi}{2}+\alpha} \sqrt{3}V_m \sin sin\left(\omega t + \frac{\pi}{6}\right) . d\omega t \tag{8}$$

$$V_{dc} = \frac{3\sqrt{3}V_m}{\pi} cos\ \alpha = \frac{3V_{mL}}{\pi} cos\ \alpha \tag{9}$$

Where $V_{mL} = \sqrt{3}V_m = Max.\ line - to - line\ supply\ voltage$

$$V_{O(rms)} = \left[\frac{6}{2\pi}\int_{\frac{\pi}{6}+\alpha}^{\frac{\pi}{2}+\alpha} v_O^2 . d(\omega t)\right]^{\frac{1}{2}} \tag{10}$$

$$V_{O(rms)} = \sqrt{3}V_m\left(\frac{1}{2} + \frac{3\sqrt{3}}{4\pi} cos\ cos\ 2\ \alpha\right)^{\frac{1}{2}} \tag{11}$$

V. SUPERVISORY CONTROL AND DATA ACQUISITION

The basic control strategies of DC microgrids are shown in Fig. 5. Compared to decentralized and centralized control, utilizing a distributed approach in DC microgrids offers many benefits, such as the distribution of decision-making over numerous. Figure 6 shows the control strategies of DC microgrids.

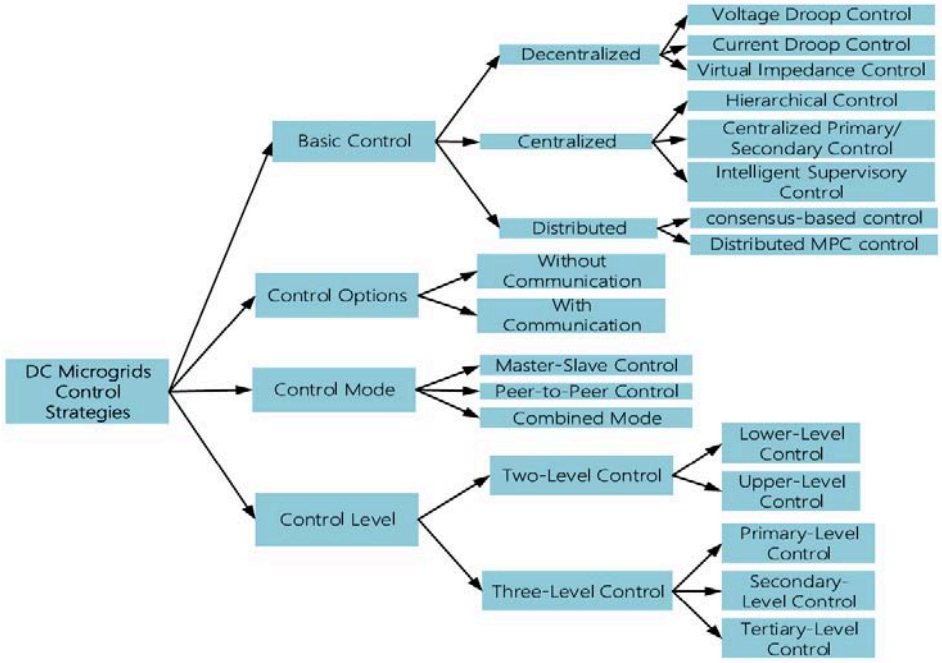


Figure 6: Control Strategies of DC Microgrids

Applications

Remote Areas:

- Providing electricity in locations far from national grids.

Smart Buildings:

- Buildings utilizing solar energy and electronic devices.

Industries:

- Operating machinery that works on DC to improve efficiency.

Renewable Energy Networks:

- Enhancing the integration of solar and wind energy.

Transportation Infrastructure:

- Electric vehicle charging networks.

SCADA Technology

SCADA stands for Supervisory Control and Data Acquisition; software allows the supervision and control of industrial processes, including manufacturing, production, power generation, fabrication, and refining.

A simple personal computer (PC) can become a workstation through SCADA programs to watch and control processes and communicate with the field. Programmable Logic Controllers (PLCs) and Remote Telemetry Units (RTUs) or intelligent devices allow for the graphic representation of a plant, the supervision of its processes, modification of certain parameters, elaboration of reports, emission of alarms, and other functions.

The processes can be controlled remotely or locally to make changes to the parameters of the process on-site (local way) or adjustments in a control room (remote way).

Substation control: Substations are essential for sustaining the flow of electricity and managing load. The SCADA system continually assesses the condition of different substation components and then sends them the appropriate control signals. It carries out tasks including bus voltage regulation, bus load balancing, circulating current regulation, and overload regulation.

Remote Communications Networks

The SCADA host in the field office or central control center receives data from distant RTU/PLCs in the field or along the pipeline via the remote communication network. Communication is the SCADA system's "glue" or "linking component," and it is crucial to the system's functioning since assets are dispersed over a wide geographic region. A SCADA system's ability to effectively handle communication with distant assets is essential to its success, and the fact that these communication mediums are still prone to malfunction is a significant problem. Figure 7 shows an RTU/PLC system connection.

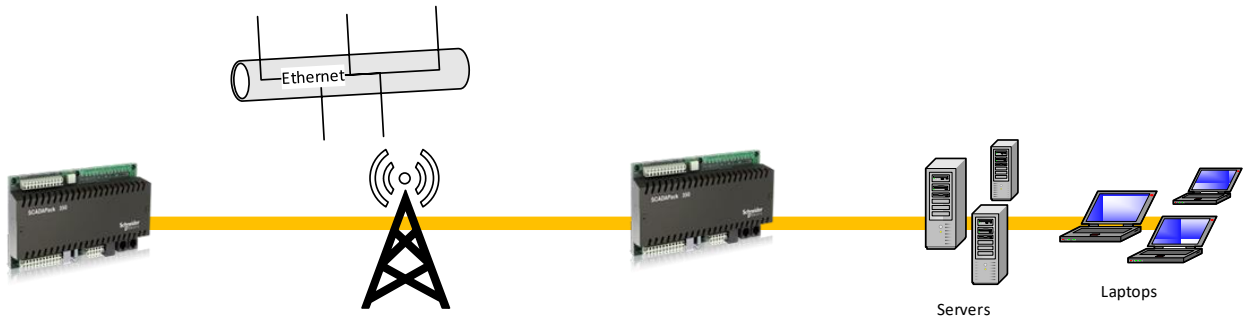


Figure 7: RTU/PLCs system connection

Any sophisticated SCADA may be broken down into its smallest components and linked through communication protocols, much like a distributed architecture. The system may be dispersed over many process control networks (PCNs), geographically isolated LAN networks in a networked architecture. A network architecture might consist of many distributed architecture SCADAs operating concurrently, each with a single supervisor and historian. This enables a more economical option in very large-scale systems. Internet of Things, fourth generation

Modern SCADA systems are driven to provide operational benefits from the SCADA host down to the instrumentation, not only in controlling and retrieving data but also in engineering, implementing, operating, and maintaining these assets.

- Provide instrumentation and RTUs/PLCs for easily managed asset or process solutions.
- Create and implement open standards utilizing best practices established by open organizations rather than a single manufacturing body to facilitate the integration of assets inside a SCADA system. As a result, the cost of owning SCADA will go down.
- Implement several practices and procedures, in addition to technological solutions, to create safe environments for SCADA systems, their assets, and processes.

Figure 6 shows the proposed control schematic's time response using MATLAB/Simulink. In the last step of the work, the hardware is implemented, and the power system framework of a microgrid is developed for the power laboratory-based SCADA using Lucas-Nülle power engineering laboratory equipment to model and simulate the distribution-level smart grids/microgrids. Figure 7 shows a proposed DC microgrid system with an energy management system. Figure 8 shows a schematic diagram of conventional droop control of a DC microgrid with the energy management system, which represents a hybrid AC/DC microgrid. Figure 9 shows a Time response of the proposed control schematic using MATLAB/Simulink.

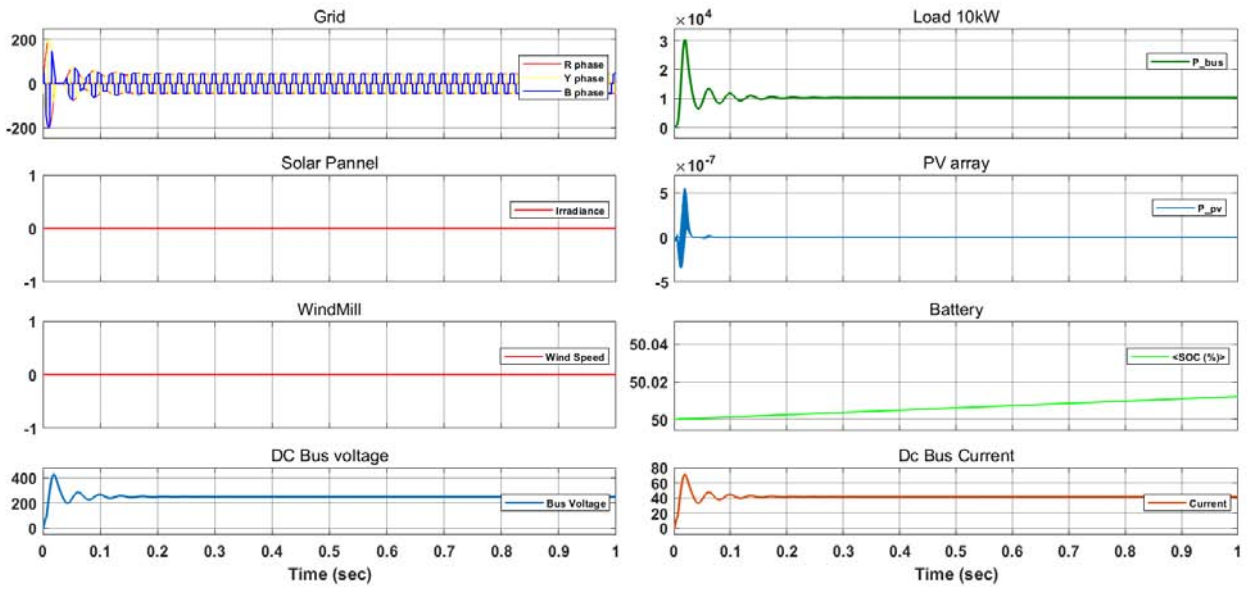


Figure 6: Time response of the proposed control schematic using MATLAB/Simulink

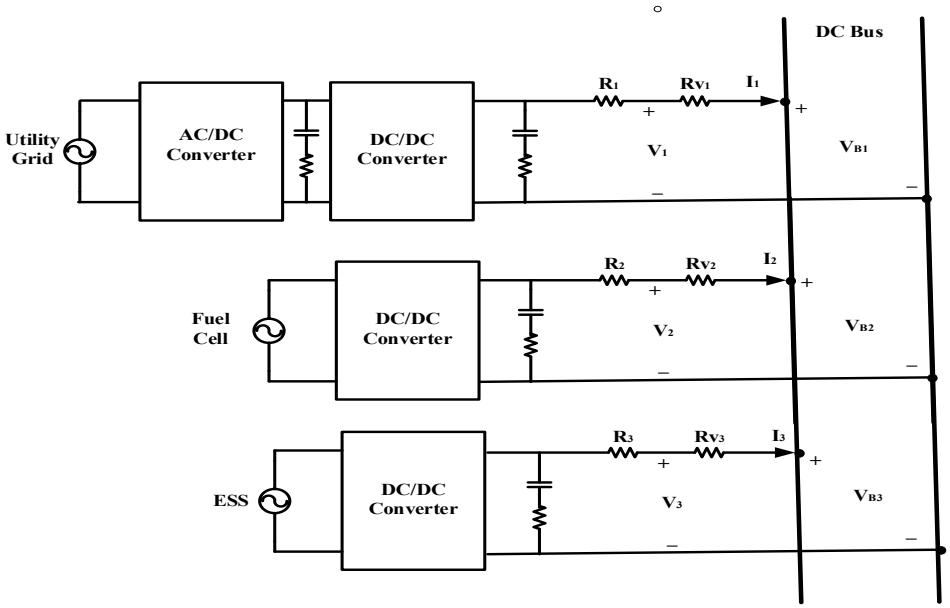


Figure 7: Proposed System of standalone DC microgrid with energy management system

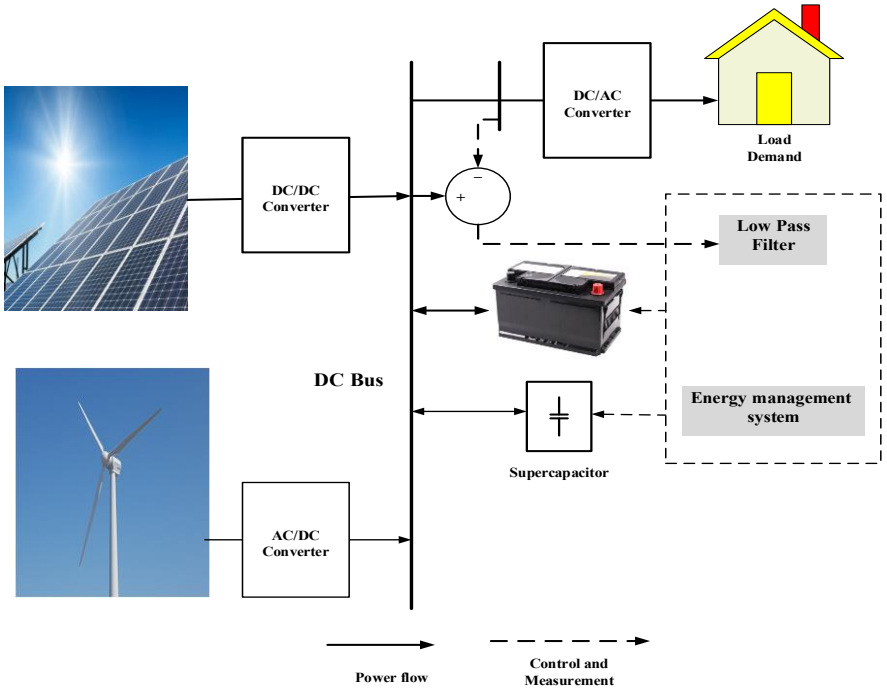


Figure 8: Diagram of conventional droop control of DC microgrid with energy management system

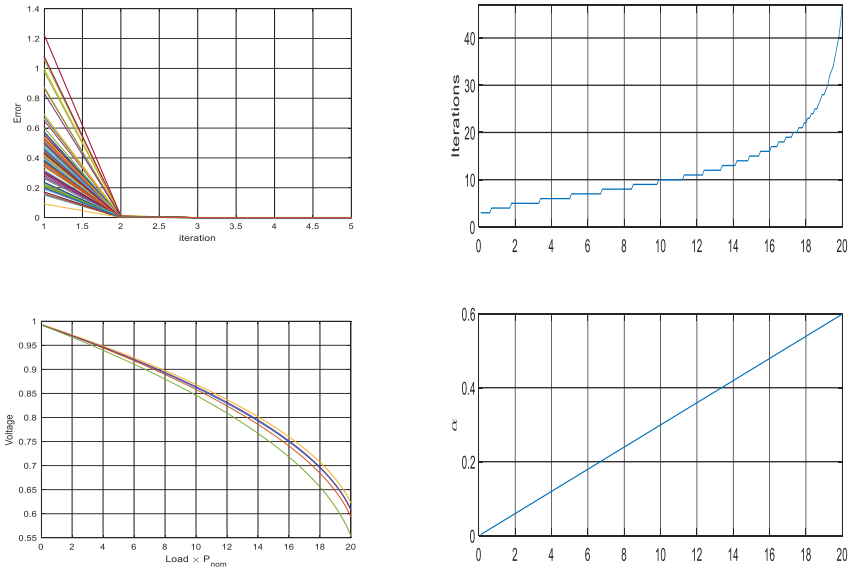
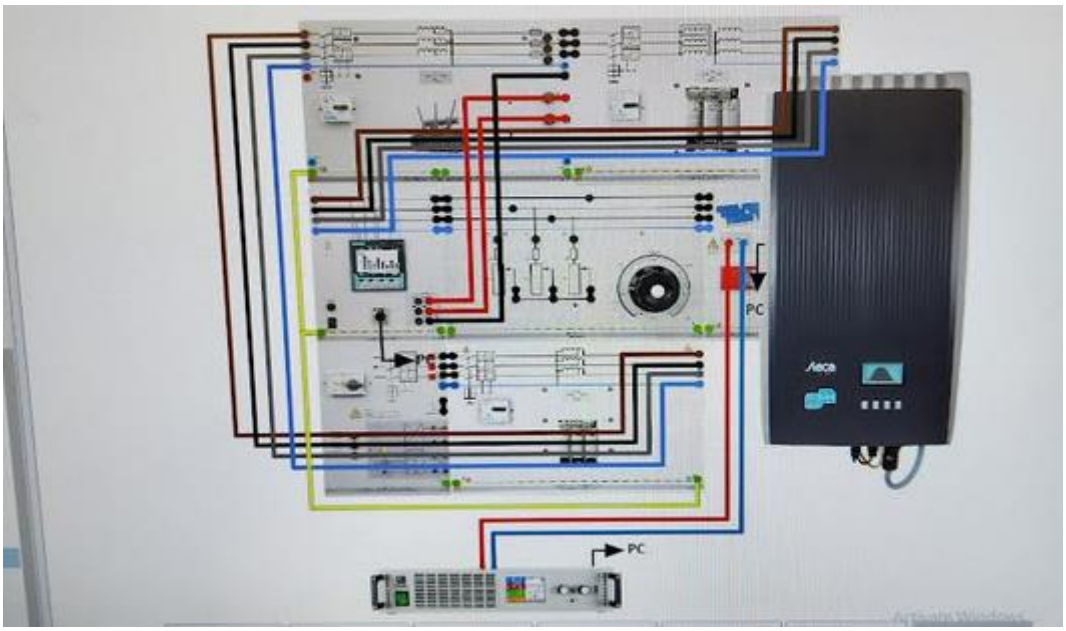


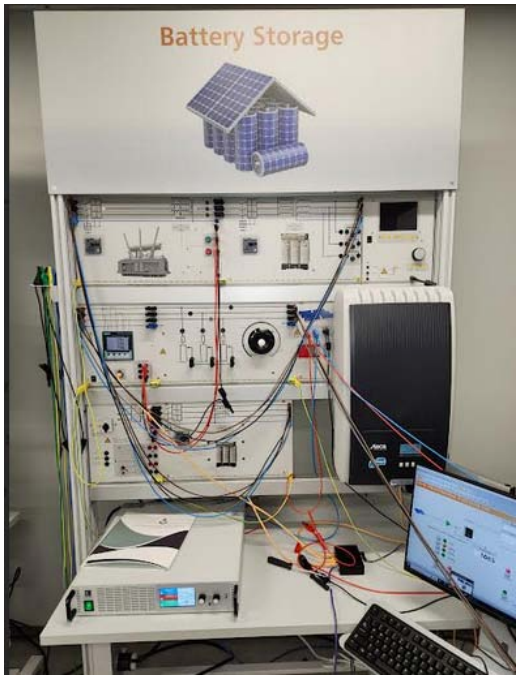
Figure 9: Time response of the proposed control schematic using MATLAB/Simulink

VI. RESULTS AND DISCUSSION

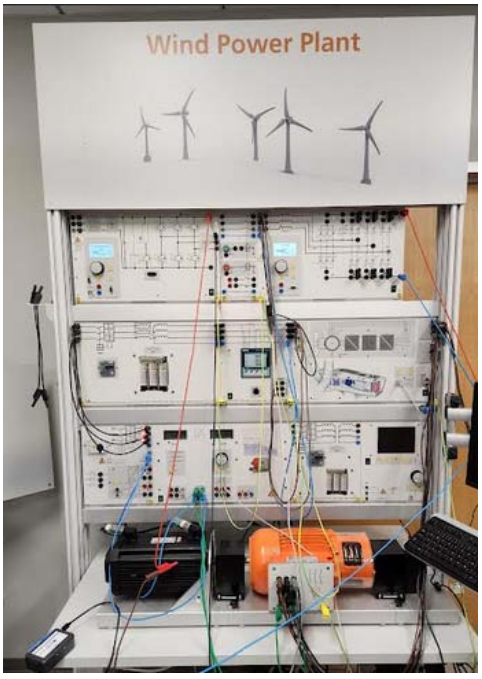
The SCADA system provided Hardware Implementation and setup. I conducted the physical building of the system as I was learning how to work it.



(a) Wire Setup Diagram



(b) Complete Solar Setup



(c) Complete Wind Turbine Setup

Figure 10: Renewable sources used in this work

6.1 Results

The first step was to dial both the solar and the wind modules to 120 volts to tie both systems together. This in fact produced a challenge as the system was easy to overload. If the system overloads everything in each module had to be powered down and restarted. Achieving this feat was used with trial-and-error method. After a few weeks was able to get each module to read out 120 volts and combine both modules to the micro grid. I tested this theory by measuring the voltage at each point across the combined system. Another challenge faced was keeping the current stable as it was not reading out at certain positions across the system due to the wind turbine inverter timing out or overload, nonetheless once everything was stable and working the reads across the system output the desired results. After accomplishing

120 volts across the system, I tested the system by add a different load source. For the rectifier found that using the setting BC6 at 97 degrees was the perfect position to keep the system from overloading.

For the inverter the setting used was DC PWM (1QH) at 30%.

Wind Turbine

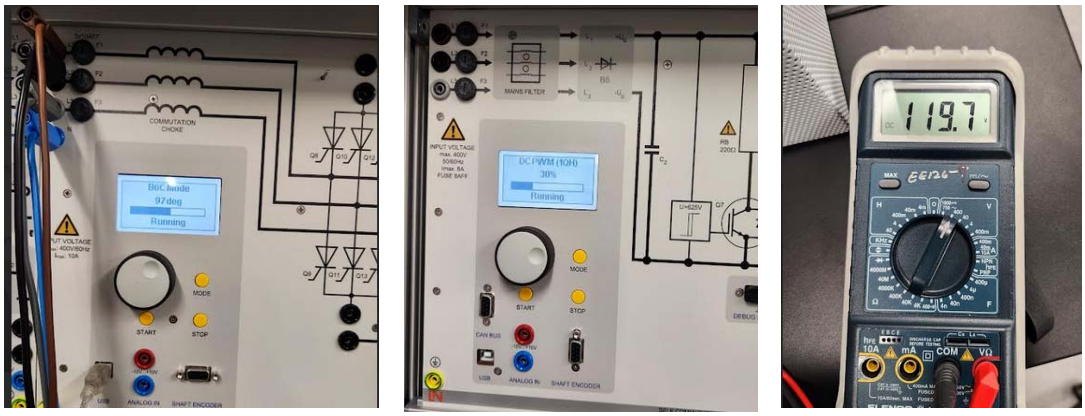
$$Vdc = \frac{3\,VSline\,Max\;\;[1 + cos\,(60 + \alpha)]}{\pi}$$

$$= \frac{3 * 300\sqrt{2}\;\;[1 + \;cos\,(60 + 73)]}{\pi}$$

$$= 900\sqrt{2}\;[1 + \;cos\,(60 + 73)]$$

$$Vdc = \;128.835\;Volt$$

The flow charts below were the results gathered from de-rating the solar cell system using SCADA. Tables 2-5 shows the Power and voltage measurements at DERATING 0, 30% , 60% and 100%.

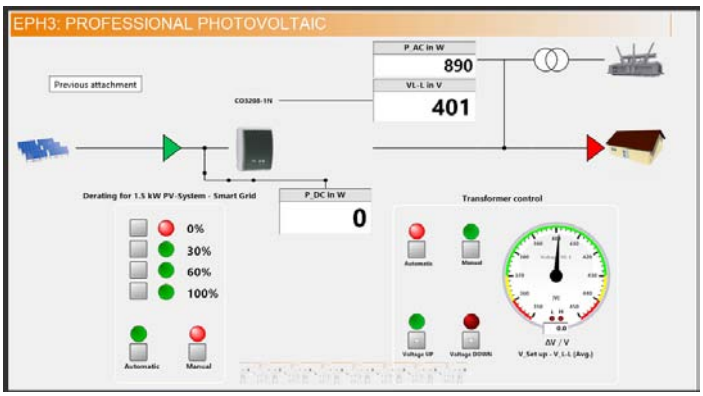


(a) (b) (c)

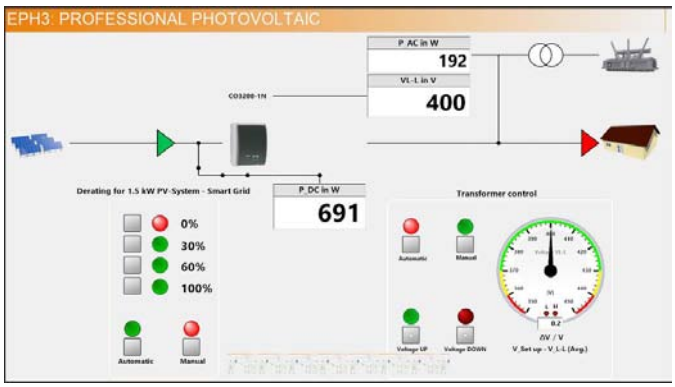
Figure 11: System output (a) Rectifier (b) Inverter (c) System Voltage

Table 2: Power and voltage measurements at DERATING 0%

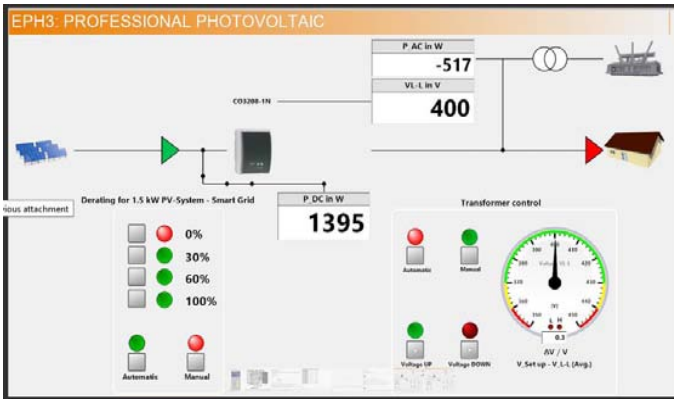
Shading	PDC (W)	P_AC in (W)	VL-L
0%	0	890	401
10%	121	766	400
20%	265	623	400
30%	408	478	400
40%	550	336	400
50%	691	193	400
60%	833	44	399
70%	973	-95	400
80%	1115	-235	400
90%	1255	-383	399
100	1395	-516	401



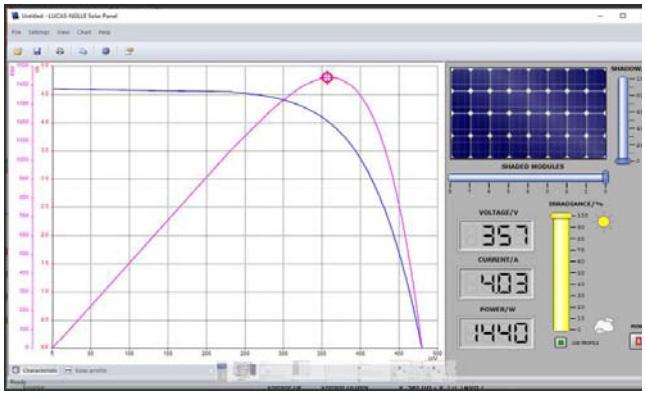
(a) Derating 0% and 0% shading



(b) Derating at 0% and 50% shading



(c) Derating at 0% and 100% shading



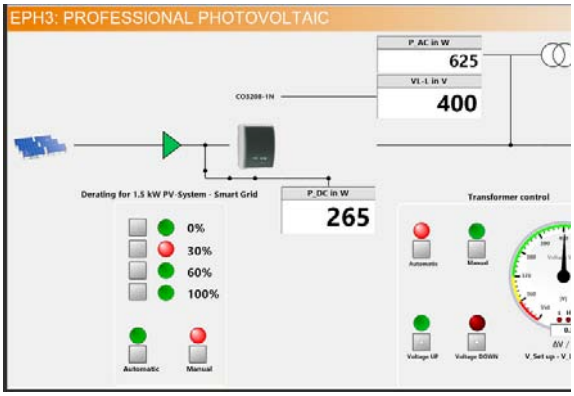
(d) Derating at 0% and 100% shading

Figure 12: SCADA results at 0%, 50% and 100% shading

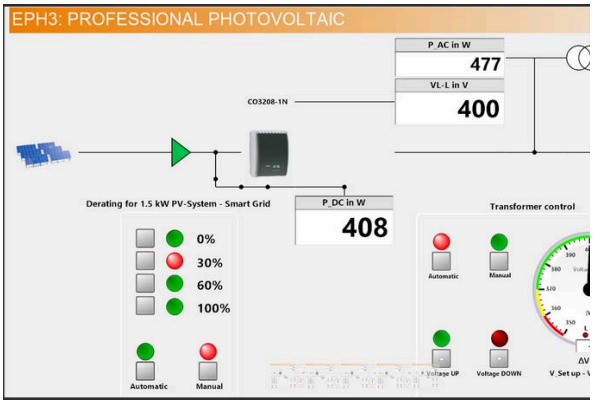
Table 3: Power and voltage measurements at DERATING 30%

SHADING	PDC (W)	P_AC in (W)	VL-L
0%	0	886	400
10%	265	767	400
20%	265	625	400
30%	408	477	400
40%	550	339	401
50%	691	187	399
60%	832	48	400
70%	974	-90	401
80%	1097	-215	401
90%	1097	-211	401
100%	1096	-222	399

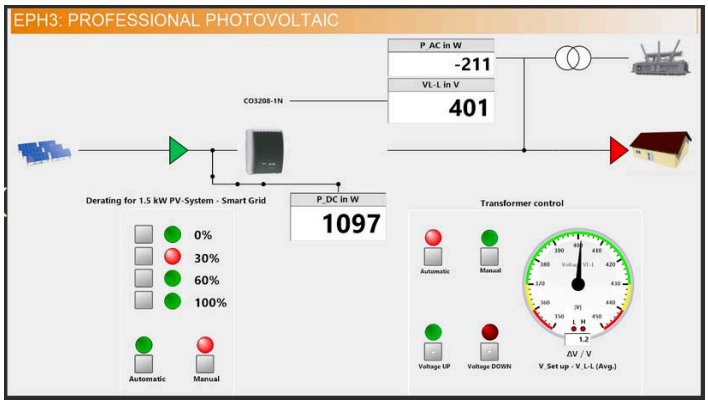
(c) Derating at 30% and 90% shading



(a) Derating at 30% and 20% shading



(b) Derating at 30% and 30% shading



(c) Derating at 30% and 90% shading

Figure 13: SCADA results at 20%, 30% and 90% shading

Table 4: Power and voltage measurements at DERATING 60%

Shading	PDC(W)	P_AC in (W)	VL-L
0%	0	891	400
10%	120	767	401
20%	265	624	402
30%	408	475	400
40%	548	338	401
50%	552	339	402
60%	552	336	401
70%	552	335	399
80%	552	340	399
90%	552	337	400
100%	552	334	400

Table 5: Power and voltage measurements at DERATING 100%

Shading	PDC(W)	P_AC in (W)	VL-L
0%	8	887	400
10%	8	887	401
20%	8	888	401

30%	8	890	401
40%	8	889	401
50%	8	886	401
60%	8	883	400
70%	8	884	400
80%	8	883	400
90%	8	886	400
100%	8	876	399

6.2 Discussion

As I progressed near the end of this work, I contemplated how I could make the system better in real time. Though the equipment appears to be an alpha model I would change a few things on how the system works. The first thing that I noticed early on was that there was only one dingle drive. This drive is the key to the software portion of the work. It is used to verify authenticity and allows for the system to verify that active components and equipment are registered. (I.e. the power quality meter, the converter). Another flaw that I saw was that the wiring cords were too short. I had to connect multiple cords to extend the length to make some of the connections. This caused safety trip issues if you were not paying attention. However, these small setbacks created a learning opportunity and allowed for more understanding in the process.

Due to the popularity of the power lab and SCADA system, I had the pleasure to demonstrate what I had built and worked on over the course of the study. Students from other universities such as Texas A&M were impressed and excited to see firsthand what Prairie View offered its students. Professional employees from multiple companies also came and experienced the power lab. This system is state of the art due to not many universities or companies have the technology. Since the power lab is large in cost the benefits and hands-on learning is worth it. It gives the students a look into what is reality but on a small scale.

VII. CONCLUSION

Renewable energy sources can be cost effective and efficient in it's make up. It highlights how it can be a reliable source of energy for years to come. Depending on the setup it can be used it can and off the grid and give the same results as if it was connected to the grid. These energy sources can be used full-time, part-time, or can support the grid. The scale in this work was small in size but can also be retrofitted for a scale. This work took me approximately 4 months to complete. Starting with learning how the system works to physically connecting the units. Ironing out all the kinks and setbacks using trial and error and help from my mentor allowed me to gain the knowledge necessary to finish the work.

Finally, I found this work to be challenging but fun. Knowing that I would be ahead of the curve once I stepped into the real world where these types of renewable energy sources are being used in everyday life, no matter the scale or setup.

REFERENCES

1. F. Gao, S. Bozhko, A. Costabeber, C. Patel, P. Wheeler, and C. I. Hill, "Comparative stability analysis of droop control approaches in voltage-source-converter-based DC microgrids," IEEE Transactions on Power Electronics, vol. 32, no. 3, pp. 2395–2415, Mar. 2016.
2. P. Prabhakaran, Y. Goyal, and V. Agarwal, "Novel nonlinear droop control techniques to overcome the load sharing and voltage regulation issues in DC microgrid," IEEE Transactions on Power Electronics, vol. 33, no. 5, pp. 4477–4487, May 2017.
3. E. Espina et al., "Distributed control strategies for microgrids: An overview," IEEE Access, vol. 8, pp. 193412–193448, 2020.
4. C. Papadimitriou, E. Zountouridou, and N. Hatziargyriou, "Review of hierarchical control in DC microgrids," Electric Power Systems Research, vol. 122, pp. 159–167, May 2015.

5. P. Lin et al., "A semi-consensus strategy toward multi-functional hybrid energy storage system in DC microgrids," IEEE Transactions on Energy Conversion, vol. 35, no. 1, pp. 336–346, Mar. 2019.
6. M. S. Alam, F. S. Al-Ismail, and M. A. Abido, "Power management and state of charge restoration of direct current microgrid with improved voltage-shifting controller," Journal of Energy Storage, vol. 44, pp. 103253, Nov. 2021.
7. Z. Peng et al., "Droop control strategy incorporating coupling compensation and virtual impedance for microgrid application," IEEE Transactions on Energy Conversion, vol. 34, no. 1, pp. 277–291, Mar. 2019.
8. B. Hredzak, V. G. Agelidis, and M. Jang, "A model predictive control system for a hybrid battery-ultracapacitor power source," IEEE Transactions on Power Electronics, vol. 29, no. 3, pp. 1469–1479, Mar. 2013.
9. Q. Xu et al., "A decentralized power management strategy for hybrid energy storage system with autonomous bus voltage restoration and state-of-charge recovery," IEEE Transactions on Industrial Electronics, vol. 64, no. 9, pp. 7098–7108, Sep. 2017.
10. R. Zhang, B. Hredzak, and T. Morstyn, "Distributed control with virtual capacitance for the voltage restorations, state of charge balancing, and load allocations of heterogeneous energy storages in a DC datacenter microgrid," IEEE Transactions on Energy Conversion, vol. 34, no. 3, pp. 1296–1308, Sep. 2018.

This page is intentionally left blank

Understanding the World Through Systems: History, Classification and Future Directions

Kochetov N. V.

ANNOTATION

The article is devoted to the understanding of the world as an object consisting of systems, as separate "bricks", despite their diversity.

Non-living systems develop according to the laws of natural sciences: physics, chemistry, astronomy. Their development is not targeted, while there is an increase in disorder (entropy).

Living systems strive to survive and expand, while they transform everything around them to improve their existence. Disorder (entropy), from the point of view of living systems (in relation to the achievement of a goal), decreases.

A general classification of the systems of the surrounding world is given. The evolution of living systems and the features of their development are explained.

Keywords: system, non-living systems, living systems, social systems, evolution, development.

Classification: LCC Code: Q295

Language: English

Great Britain Journals Press

LJP Copyright ID: 392913
Print ISSN: 2631-8474
Online ISSN: 2631-8482

London Journal of Engineering Research

Volume 25 | Issue 1 | Compilation 1.0

Understanding the World Through Systems: History, Classification and Future Directions

Kochetov N. V.

ANNOTATION

The article is devoted to the understanding of the world as an object consisting of systems, as separate "bricks", despite their diversity.

Non-living systems develop according to the laws of natural sciences: physics, chemistry, astronomy. Their development is not targeted, while there is an increase in disorder (entropy).

Living systems strive to survive and expand, while they transform everything around them to improve their existence. Disorder (entropy), from the point of view of living systems (in relation to the achievement of a goal), decreases.

A general classification of the systems of the surrounding world is given. The evolution of living systems and the features of their development are explained.

Keywords: system, non-living systems, living systems, social systems, evolution, development.

I. INTRODUCTION

In recent years, the concept of "system" has been increasingly used. As a result, the concept of a system is found in almost all fields of knowledge, which has given rise to many definitions. It is almost impossible to give an accurate comprehensive definition of the system. Any of them will be relative.

Therefore, it is better to consider the historical process of the formation of the concept of "system".

The system (from the Greek. Σύστημα – a combination, a whole made up of parts, a connection) is a set of elements that are in relations and connections with each other, which forms a certain integrity, unity. [1]

Having undergone a long historical evolution, the concept *of system* has become one of the key philosophical, methodological and special scientific concepts since the middle of the 20th century. In modern scientific and technical knowledge, the development of problems related to the study and design of systems of various kinds [2] is carried out within the framework of the systems approach [3], the general theory of systems [4], various special theories [5], in cybernetics [6], and systems engineering [7], system analysis, etc. [8]

II. HISTORY OF DEVELOPMENT

In ancient Greek philosophy and science (Euclid [9], Plato [10], Aristotle [11]) the idea of the systematic nature of knowledge (the axiomatic construction of logic and geometry) was developed. constructions of scientific systematics. 17th and 18th centuries, which strove for a natural interpretation of the systematic nature of the world (for example, the classification of Carl Linnaeus) [14]. In the philosophy and science of modern times, the concept of system was used in the study of scientific knowledge; at the same time, the range of proposed solutions was very wide, from the denial of the systemic nature of scientific and theoretical knowledge (Etienne Condillac) [15] to the first attempts at a philosophical substantiation of the logical-deductive nature of knowledge systems.

And to this day, in the absence of a precise definition of the concept of a system, researchers widely use it at an intuitive level. [16]: "Most researchers intuitively realize that there is still a real commonality in this variety of directions, which should follow from a common

understanding of the system. However, the reality is that there is still no common understanding of the system."

It was also widely used by the classics, although the term itself is not in their works [17, 18]. In the 19th and 20th centuries, there was no such detailed terminology as there is now, some concepts underwent changes and were clarified. But the concept of system was in the air in the works of scientists, penetrating the content of the works. The concept of complex systems appeared: **"An** electron is as inexhaustible as an atom**"** [18].

A great contribution to the theory of systems was made by the Russian philosopher A.A. Bogdanov, in his work "Tectonomics", republished in the 1880s [19]. The work concerned organizational structures.

Later, the "General Theory of Systems" was formulated [20], in which systems are divided into closed and open (developing). The division is relative, since all systems are under the influence of entropy in time. Inanimate systems tend to increase disorder. Living systems strive to create conditions for habitation, for which they develop surrounding systems for this purpose.
There are several modern definitions of the system.

Academic definition of the concept of a system by Academician P.K. Anokhin, who studied living systems.

A system is a complex of selectively involved elements that interact to achieve a given useful result, which is accepted by the main system-forming factor [21].

Disputes about the *classification of systems* continue [3, 22, 23]. We can propose another one based on the evolution of systems. First of all, it is advisable to divide systems into *non-living* and *living*.

III. NON-LIVING SYSTEMS

Non-living systems are characterized by the fact that they develop uncontrollably by anything Albert Einstein once expressed it as follows: "Nature hides its secrets due to its inherent height, and not by tricks" [24, 25]. Non-living systems change over time. In comparison with living systems, the time of change of non-living systems occurs either quickly (changes at the molecular level) or very slowly (at the planetary level). the change of day and night, seasonal changes (winter-summer), tides, movement of air masses, water as a result of sea flows.

That is, the time of changes in inanimate nature has the widest range. For now, we can say that this range is unlimited, as is space. It is supposedly described by the "big bang theory", and this area remains open for study.

Any system, as a group of interconnected elements, can remain intact under certain conditions. For example, an atom of a substance. There are stable states (in our usual conditions), and there are states that disintegrate and can exist in a short period of time (radioactive substances). Most of all in the universe is elemental hydrogen. Why is there so little of it on Earth in its free form? Where does so much oxygen come from on Earth, which, when combined with hydrogen, formed a lot of water? The force of gravity does not keep hydrogen on the Earth's surface. Other gases are simply squeezed out from the surface of the Earth. But some of it combined with oxygen and formed water. Oxygen is a very active element, so there are a lot of oxides on Earth, in addition to water.

What are the favorable conditions on Earth? First of all, high stability. There is a magnetic field, an atmosphere, which protect from the adverse effects of the external environment. The Earth's rotation makes it possible to maintain the temperature on its surface within 200-300 degrees Kelvin [26, 27]. The pressure of the atmosphere is such that many substances are in a gaseous state [28].

The presence of winds allows you to move large volumes of gases and liquids, which can be a building material during chemical reactions. For example, carbon dioxide, calcium and magnesium salts dissolved in water.

Water occupies an exceptional place for favorable conditions: on Earth it can be in several states of aggregation: gaseous, liquid, solid. Fluid mobility provided an environment for the development of life. The evaporation of water led to the formation of clouds that carried large masses of water in the form of rain over the Earth's surface. This created conditions for life on the surface as well. Clouds are an important stabilizer of temperature on the Earth's surface.

A characteristic feature of the development of non-living systems is that they are not targeted. They develop according to the laws of physics, chemistry, the laws of motion of celestial bodies, and the development of the Galaxy. The action of the laws of the development of inanimate nature is chaotic. Usually, the development of non-living systems leads to disorder (an increase in entropy).

These laws are objective, that is, they were before, are and will be after the development of living systems. But living systems depend on the behavior of non-living systems. Therefore, the study of non-living systems makes it possible to use these laws in the interests of humanity. By combining inanimate systems, mankind achieves important target results that cannot be achieved in inanimate nature (technical systems: engines, artificial materials).

Non-living systems are the "building material" on which living systems arise and develop. Conditions contribute to the origin or introduction of living systems from outside. And then evolution is possible (if the development of non-living systems does not interfere with this.

IV. LIVING SYSTEMS

Living systems are more studied systems from the point of view of science. There are several fields that study them (biology, botany, zoology, anthropology, etc.). The time intervals of changes in such systems are more definite, although this range is quite wide (the periods of division of bacteria and viruses are quite short). The change of generations of plants and animals (including humans) is slower. The evolution of life on Earth is a slower process (so far we do not know of other living systems). But they are also very short compared to the "big bang" process.

So far, we cannot explain the emergence of living systems (life). But scientists are concerned about the question of what will happen as a result of the development of living systems [29]. It is clear that evolution was impulsive. At some moments, there were stops and rollbacks due to the influence of the external environment (changes in non-living systems). For example, global catastrophes that led to the formation of chalk deposits, fossil fuels (gas, oil, coal). And later local catastrophes that led to the disappearance of entire living systems (the extinction of mammoths).

Closer examples of the development of living systems are the evolution of man. Darwin put forward the theory of evolution. Did man evolve from apes or are they species that evolved in parallel? Apparently, a common ancestor howled, but it disappeared as a result of one of the disasters. Neanderthals, Cro-Magnons, Australopithecus - what place do they occupy in the process of human evolution?

As we can see, living systems that are closer to us are also inexhaustible for study. Perhaps the simplest forms of living systems take place in the Universe, but their development and evolution are possible only under certain favorable conditions (very rare). So far, no other Civilizations have been discovered, but does this mean that the Earth is the only non-living system suitable for the development of living systems in the infinite Universe?

Non-living systems develop according to the objective laws of chemistry, physics, and the laws of the development of galactic systems. Living systems cannot influence these laws, but they can use them to produce useful results. This also applies to living systems that are in a state of symbiosis with other living systems (man and botanical, zoological, bacteriological systems, objects of genetic engineering).

V. ARTIFICIAL LIVING SYSTEMS

Artificial systems (systems related to life) are systems that have arisen as a result of the evolution of life for its development (symbiosis, the formation of the "food pyramid").

Such systems have goals because they serve the interests of living systems (humanity). Humans have not created artificial life, but they have been successful in the selection of living systems that are useful to them.

Centuries of work to improve these living systems led to the development of agriculture, primarily crop production and animal husbandry. The potential of the seas is being actively developed. The selection of many plants, the breeding of productive breeds of livestock, the development of the potential of the sea led to the fact that large volumes of useful plants and animals were grown.

The accumulated information is reflected in the form of arrays of knowledge about living useful systems [30, 31]. This allows you to expand the areas of animal breeding, increase sown areas, and develop the potential of the seas.

Thanks to the successful development of agriculture, the population is steadily growing [32, 33]. However, this process will come to a halt: in the developed countries of Europe, there is almost no natural population growth. The quality of life has changed: the period of starting a family is preceded by a period of training, education, career building, etc.

Next, let us consider the social systems that are closest to this issue.

VI. SOCIAL SYSTEMS

Social systems are one of the specific types of systems. On the one hand, they are not living systems in their pure form, like a single individual. On the other hand, these are artificial systems that form relationships between people.

In the animal form, we also observe the rudiments of such systems: flocks of birds, shoals of fish, herds of herbivores, flocks of wild animals. This allows you to get a greater competitiveness of a particular type of living world.

But to the greatest extent, such behavioral systems were received in the development of mankind. At first, these were separate tribes. Then separate state structures began to form: Ancient Egypt, Babylon, the civilizations of China and the original states of the American region. This led to the stratification of the population of the state (the ruling elite, ordinary members of society).

As they developed and grew, states began to interact with each other (trade, confront, compete). This led to the emergence of a monetary system, jurisdiction, law enforcement agencies, armies, etc. The economy began to develop: crafts, construction, manufacture of ships, weapons, household items. That is, the structure of the state developed, became more complex, and became multi-invested [34].

Different segments of the population specialized in certain types of activities. Such universal trends in the development of society as health care, sports, education, art, science, and culture have appeared. Each of which is a separate, independent system.

For example, the international trade system, presented in the form of the Observatory of Economic Complexity (EC) [35], which is designed to increase the efficiency of economic ties between businesses.

VII. ARTIFICIAL NON-LIVING SYSTEMS

Non-living systems are the "bricks" on which living and artificial systems develop. These "bricks" are inexhaustible. But you need to know how to make something out of them, that is, you need TECHNOLOGIES. Examples of the influence of technology on the development of artificial systems: people have learned to smelt iron, produce aluminum, make artificial diamonds from carbon material, polymerize artificial materials from natural gas.

The most studied are technical systems, which are non-living artificial systems. As a result of the

research, it was possible to obtain complex nested systems. Examples include steam engines and turbines, internal combustion engines, electric motors, and electrical power generators.

What is the manifestation of nesting of systems? Consider the internal combustion engine. The main elements of this system will be the crank mechanism, ignition system, fuel mixture preparation system, exhaust system, control system, and a certain type of fuel. Each element performs a specific function. But in order to perform this function, the element in question is itself a system of a lower level, and so on.

If we consider a higher level in relation to the internal combustion engine, then here we can give an example of systems in which internal combustion engines are used: vehicles (cars, diesel locomotives, motor ships), power machines (mobile power plants, construction and technological production equipment).

At the same time, the engine, as an element of the system, will be supplemented by other systems that will determine the purpose of the system of a higher level. For example, a car that turns on the engine will be supplemented with a propeller (wheels), steering mechanism, gearbox (gearbox), accommodation for the driver and passengers (body), etc. Such development of artificial systems reduces disorder (entropy).

As we can see, each element of the artificial system will have a pronounced target character. In this way, artificial systems differ from non-living systems, which develop according to the objective laws of physics, chemistry, celestial mechanics, and the theory of probability. The development of inanimate nature tends to increase entropy.

Examples of such systems are social and state institutions, distribution systems (money), and technical systems.

It is these systems that continue to develop rapidly. Since artificial systems are created for the interests of people, they should not develop in such a way as to harm the existence of humanity. This should be the basis for the development of artificial systems.

The development of living non-artificial systems also has a purposeful character: self-preservation and expansion. But humanity uses much more opportunities for this, thanks to the acquired knowledge, science, and built infrastructure. Although initially the external environment is the same for all living systems.

VIII. CONCLUSIONS AND PROSPECTS

1. The concept of "system" is widespread and has several definitions in various fields of knowledge. Almost none of them fully reveals this concept.
2. It is important to distinguish between three types of systems: inanimate, living, systems for achieving certain goals of living systems (artificial).
   Non-living systems are objective in nature, they develop according to objective laws (the laws of physics, chemistry, and the development of galactic systems). The result of their action is due to the laws of probability theory, which is why the result is chaotic (tandem). The development of such systems tends to reduce order (increase entropy).
3. In the last century, artificial systems begin to develop rapidly and tend to accelerate this development. Artificial systems can be non-living (technical), living (based on living systems: new varieties of plants, breeds of livestock, bacterial strains, vaccines), combined (in recent years, research in this area has been actively conducted).
4. Living systems adapt to existence in the conditions of non-living systems (either die or follow the path of evolution). For successful evolution, artificial systems can appear for the purposes of living systems (artificial). Such systems are created for their intended purpose. They are based on a combination of several non-living systems, on living systems, or a combination of them. Such systems lead to ordering (reducing entropy).
5. Living systems have goals: self-preservation and expansion. At the same time, their development is aimed at overcoming the

conditions created by the development of inanimate systems. On the other hand, in order to expand living systems, they begin to compete with each other. Here the process of destruction, squeezing out or subjugation is observed. These processes can be intertwined. As a result, symbiosis can occur.

Living and artificial systems function within the framework of non-living systems and depend on the development of the latter. Therefore, in order to preserve life, it is important to study the laws of development of non-living systems in order to prevent catastrophes that may arise as a result of the development of these non-living systems.

6. Humanity is at the top of the evolutionary pyramid. It is a living system based on non-living, living, and artificial systems. Naturally, mankind is characterized by progressive development, the desire to expand the environment of its existence. Just like all living things, competition based on knowledge and science is inherent.
7. The desire for self-survival is no less strong, like any other living system. Further development of mankind should follow the path of eliminating the dangers that may come from non-living systems (global catastrophes, galactic collisions, radiation, poisoning, etc.) and living systems (diseases, toxic strains of bacteria and viruses, Varangian bacteria from other terrestrial environments and space systems).

In the conditions of earthly competition, it is in this direction that it is necessary to evolve.

SOURCES

1. Philosophical Encyclopedic Dictionary. Editors: L.F. Ilyichev, P.F. Fedoseev, S.M. Kovalev, V.G. Panov – Moscow: Sov. Encyclopedia, 1983. – 840 p.
2. Prigogine, I. Stengers Order from Chaos. New Dialogue of Man with Nature Moscow: Progress, 1986.- 432 p.
3. Meyen S.V., Shreider Y.A. 1976. Methodological aspects of the theory of classification. Questions of Philosophy. 12: 67-79.
4. Bertalanffy, L. von (1969). Research on General Systems Theory. Moscow: Progress.
5. Jolly, R. (2015). Systems thinking for business: Capitalize on structures hidden in plain sight. Systems Solutions Press.
6. Viner N. Cybernetics, or Control and Communication in Animal and Machine. /Lane. I.V. Solovyov and G.N. Povarov / Ed. by G.N. Povarov. – 2nd edition. Moscow: Nauka; Main Editorial Board of Publications for Foreign Countries, 1983. – 344 p.
7. Good G.H., Macol R.E. System Engineering. Introduction to Large Systems Design. Moscow, Soviet Radio Publ., 1962. 383 p.
8. Volkova V. N., Denisov A. A. Systems Theory and System Analysis: A Textbook for Academic Bachelor's Degree. — 2nd. — Moscow: Yurayt, 2014. — 616 p.
9. Nachala M.: Nauka, 1983. — P. 159. 400 p.
10. Plato. Collected Works in Four Volumes / General Edition by A. F. Losev and V. F. Asmus. St. Petersburg: St. Petersburg University Publishing House: 2007. — T. 2. 626 p.
11. Metaphysics / Aristotle; [transl. from ancient Greek, introduction, article and comment by A. V. Markov]. Moscow: RIPOL Classic, 2018. 384 p.
12. Konnikov I. A. Materializm Spinoza [Spinoza's Materialism]. — Moscow: Nauka, 1971. — 268 p.
13. Leibniz Gottfried Wilhelm (Leibniz Gottfried Wilhelm) // Physics: Biographical reference book / Ed. by A. I. Akhiezer. — Plato Publishing House. Collected Works in Four Volumes / General Edition by A. F. Losev and V. F. Asmus. St. Petersburg: St. Petersburg University Publishing House: 2007. — T. 2. 626 p
14. Система органического мира / А. Л. Тахтаджян // Сафлор — Соан. — М.: Советская энциклопедия, 1976. — (Большая советская энциклопедия: [в 30 т.] / гл. ред. А. М. Прохоров; 1969—1978, т. 23).
15. *Г. Л. Зельманова*. Кондильяк, Этьен Бонно де // Большая советская энциклопедия: [в 30 т.] / гл. ред. А. М. Прохоров. — 3-е изд. — М: Советская энциклопедия, 1969—1978.

16. Agoshkova E.B., Akhlibininsky B.V. Evolution of the Concept of the System. Questions of Philosophy. - 1998. - №7. Pp. 170-179.
17. Friedrich Engels. Dialectics of Nature. - Moscow: Politizdat, 1975.-359 p.
18. V. I. Lenin; Institute of Marxism-Leninism at the Central Committee of the CPSU. - 5th ed. Moscow: Gos. Izd-vo polit. lit., 1968. — T. 18. Materialism and empirio-criticism. — S. 7—384.
19. Bogdanov, A. A. Tektology: (Universal Organizational Science): in 2 books. / A. A. Bogdanov. Moscow, Ekonomika Publ., 1989.
20. Ludwig von Bertalanffy General System Theory. New York: George Braziller: 1968. – 289 p.
21. Anokhin P. K. Ahead of the reflection of reality. 1962. № 6. Pp. 97—109.
22. Lyubishchev A.A. Problemy systematiki [Problems of systematics]. Problems of evolution. Vol.1. Novosibirsk: Nauka: 1968. P.7-29.
23. Karpov, V.E., Karpova, I.P. On the Question of the Principles of System Classification. Moscow: Finance and Statistics, 1989. – 350 p.
24. Quotes by A. Einstein. [Electronic resource] / Mode of access: LiveJournal.https://quadr-ro.livejournal.com/41651.html. - Date of access: 01.02.2025.
25. Albert Einstein, Leopold Infeld. Evolution of Physics. Development of Ideas from Initial Concepts to the Theory of Relativity and Quanta. Moscow: "NAUKA", 1965.- 328 p.
26. Pogosyan Kh. P., Turketti Z. L. Atmosphere of the Earth: A Manual for Teachers. Moscow, Prosveshchenie Publ., 1970. 320 p.
27. Atmospheres of the Solar System. LiveJournal is a network community LiveJournal [Elektronnyi resurs] / Rezhim dostupa: https://kiri2ll.livejournal.com/498-851.html. - Date of access: 01.02.2025.ë
28. Zemlya: kharakteristiki, stroenie i osobennosti planeti [Electronic resource] / Mode of access: https://dzen.ru/a/Y8GAX-llzhiBcXc2Y
29. Kochetov, N.V. Once Again on the Evolution of Life on Earth. Human Capital as a Factor in the Innovative Development of Society: Collection of Articles of the International Scientific and Practical Conference, Yekaterinburg, May 2, 2022. Ufa: OMEGA SCIENCE, 2022. – 216 p. P. 123-125.
30. Agricultural Encyclopedia. In 6 volumes. Ed. by V. V. Matskevich, P. P. Lobanov Soviet Encyclopedia Publishing House: 1969 — 1975.
31. Register of Medicines of Russia. Encyclopedia of Medicines. Issue 19 Author Development. G. L. Vyshkovsky; Hl. Ed. by Y. F. Krylov Publishing House RLS, RLS-Media: 2011. 1368 p. (In Russian)
32. UN Growth of the World Population [Elektronnyi resurs] / Rezhim dostupa: https://www.un.org/ru/global-issues/population. - Date of access: 01.02.2025.
33. Anthropogenesis.RU: Russian scientific and educational portal dedicated to human evolution. [Electronic resource] / Mode of access: https://antropogenez.ru/- Date of access: 01.02.2025.
34. Bohr, M.Z. History of the World Economy: Lecture Notes. Moscow, Delo i servis Publ., 2000. – 496 p.
35. OES – Observatory of Economic Complexity. OEC - Observatory of Economic Complexity [Electronic resource], Boston, MA, 2024, - Available at: https://oec.world/en – Accessed 28.01.2025.

This page is intentionally left blank

Design, Construction and Performance Comparison of Fuzzy Logic Controller and PID Controller for Two-Wheel Balance Robot

Mehrdad Esmaeilipour & Mohammad Hossein Zalzar

Islamic Azad University

ABSTRACT

In this article, the design, construction and control of a two-wheeled balancing robot was carried out with two PID controllers and fuzzy logic. This type of robot has advantages such as lower energy consumption and the ability to be used in small spaces with the ability to navigate in the entire workspace and rotate in place compared to other wheeled robots. Therefore, the need for research on the knowledge of building a two-wheeled balancing robot and controlling its balance optimally is felt to enter fields such as medical robots (assistant and military and robotics of humanoid robots), etc. For the design and modeling of the robot, SolidWorks software was used, and for programming, Arduino software was used along with the Arduino microcontroller board. In this research, the robot system is nonlinear and is considered as a black box, and two linear PID and nonlinear fuzzy controllers have been used. After conducting investigations and obtaining numerous results for both controllers in the results section, the performance of the fuzzy controller was observed to be much more optimal than the PID controller.

Keywords: fuzzy controller, PID controller, two-wheeled balance robot

Classification: DDC Code: 629.892

Language: English

Great Britain Journals Press

LJP Copyright ID: 392914
Print ISSN: 2631-8474
Online ISSN: 2631-8482

London Journal of Engineering Research

Volume 25 | Issue 1 | Compilation 1.0

Design, Construction and Performance Comparison of Fuzzy Logic Controller and PID Controller for Two-Wheel Balance Robot

Mehrdad Esmaeilipour[α] & Mohammad Hossein Zalzar[σ]

ABSTRACT

In this article, the design, construction and control of a two-wheeled balancing robot was carried out with two PID controllers and fuzzy logic. This type of robot has advantages such as lower energy consumption and the ability to be used in small spaces with the ability to navigate in the entire workspace and rotate in place compared to other wheeled robots. Therefore, the need for research on the knowledge of building a two-wheeled balancing robot and controlling its balance optimally is felt to enter fields such as medical robots (assistant and military and robotics of humanoid robots), etc. For the design and modeling of the robot, SolidWorks software was used, and for programming, Arduino software was used along with the Arduino microcontroller board. In this research, the robot system is nonlinear and is considered as a black box, and two linear PID and nonlinear fuzzy controllers have been used. After conducting investigations and obtaining numerous results for both controllers in the results section, the performance of the fuzzy controller was observed to be much more optimal than the PID controller.

The PID coefficients have been obtained experimentally, and the membership functions of the fuzzy controller have also been adjusted experimentally, and the Mamdani method and trapezoidal functions have been used. All the results have been obtained from the Arduino software and with a serial port connection between the computer and the Arduino microcontroller, all of which are presented in the results section.

Keywords: fuzzy controller, PID controller, two-wheeled balance robot.

Author α: Department of Electrical, Islamic Azad University, Bushehr Branch, Bushehr, Iran.
σ: Department of Electrical, University of Persian Gulf, Bushehr, Iran.

I. INTRODUCTION

Modern control is based on the time domain analysis of differential equation systems. Modern control has simplified system design because this theory works based on a model of the real control system. However, the stability of the system is sensitive to the model error, that is, when the controller designed based on the model is applied to the real system, the system may not be stable.

In order to prevent such a situation from occurring, the range of possible errors is first determined during design, and then the control system is designed so that it remains stable if the error is within this range. The design method based on this principle is called robust control. Both frequency response and time domain approaches are used in this theory, and this theory has a lot of mathematical complexity [1].

Two-wheeled balancing robots that act like inverted pendulums are naturally unbalanced, nonlinear, and loaded [2] and consist of two moving wheels, each located on one side of its body and driven by direct current actuators. As a model of an inverted pendulum, the robot's balance is achieved by controlling the rotation of its wheels. Since the introduction of the two-wheeled balancing robot [3], this type of robot has attracted great interest due to its many application opportunities in the field of design, controller, and signal processing of distributed control systems. Simple design and use in tight spaces, resulting in a small footprint, are also other reasons for the popularity of this type of

robot. However, the inherent tendency of the robot to be unbalanced indicates the need for dynamic control of the system.

Even with a higher height, they can accelerate quickly without falling over. On the other hand, not having more than two wheels means that they can use larger wheels to cross uneven surfaces [4]. Given the characteristics and features mentioned, this type of robot is used in fields such as agriculture, transportation, etc. [5, 6]. Also, due to its self-control, flexibility, and small size, this type of robot has high usability in dangerous and small workspaces [7]. In some studies, they are used for educational purposes [8].

The study of two-wheeled balancing robots emphasizes more on the robot balance control technology. The first researcher in this field is Professor Yamafuji, who works at the University of Electrical Communications in Japan. In 1987, he was working on an invention called a (parallel) two-wheeled robot that incorporated two-wheeled robot technology [9]. Soon after, Sony Electric developed a two-wheeled self-balancing robot called the Feelsro and demonstrated it at the June 2002 Robot World Cup. The world's first humanoid two-wheeled self-balancing robot was developed by ATR Research Laboratory in Japan. Any instability and instantaneous movement of the robot's upper body must be compensated immediately and the balance maintained [10]. Various techniques have been proposed for controlling two-wheeled robots, which require a good understanding of the mathematical model, which is represented by Lagrange's equations [3, 11, 12], Newton-Euler equations [13], Gibbs-Eppel equations of motion [14, 15], or the Ken method [10]. In general, such a device is affected by external disturbances. Also, the lack of a dynamic model, errors in parameter estimation, and noise related to the measurement of input parameters through sensors are other influencing factors that need to be considered [16]. To In order to effectively control the two-wheeled balancing robot, linear and nonlinear control strategies have been used. Among the linear controllers, PID controller [11, 12, 17-19], LQR [13] and LQG [15] have been used. These linear controllers offer a limited range of performance due to linear processing, thus limiting the performance of the robot. As a result, the use of nonlinear controller techniques can overcome these limitations. In a study conducted by Slavo et al. in 2014 on the control of two-wheeled balancing robot with LQR controller. One of the reasons cited for using the controller is the lack of an accurate mathematical model for the robot [20].

Zhang et al. [20] proposes a fuzzy fractional-order PID (FFOPID) controller for the motion control of a TWSBR system in an inclined environment. The control goal of TWSBR is to realize the wheel position control and to stabilize the non-vertical direction of intermediate body (IB). Finally, we compare the control effect of the proposed FFOPID controller with that of the integer-order PID controller, the fuzzy PID (FPID) controller, and the fractional-order PID (FOPID) controller when TWSBR moving on the inclined plane. The simulation results show that the FFOPID controller has better control performance and anti-interference ability.

Mudeng et al. [21] apply a Proportional Integral Derivative (PID) controller as a control system in a self-balancing robot with a working principle is similar to an inverted pendulum. An Inertial Measurement Unit (IMU) sensor is utilized to detect the angular acceleration and angular velocity of the self-balancing robot. The phase design is constructed by planning the robot dimension, mechanical system, and an electronic system. Particularly, this study performs mathematical modeling of the robot system to obtain the transfer function. In addition, we simulate the PID parameter with multiplication between the basic parameter and several fixed constants. The simulation results indicate that the robot can maintain its balance and remains perpendicularly stable for balancing itself.

Kumar et al. [22] investigates a novel coupling-based mixed interval type-2 fuzzy logic controller (MIT2FLC) for trajectory tracking problems of highly nonlinear and complex robot manipulator plants. For checking the performance, the MIT2FLC approach is compared with its type-1 fuzzy counterparts, namely mixed type-1 fuzzy logic controller (MT1FLC), type-1

fuzzy logic controller (T1FLC) and PID controllers. Robustness analysis of the proposed controller is investigated for external disturbances, varying system parameters, and random noise.

II. ROBOT DESIGN

In this section, items such as design and selection, structure dimensions and materials used in the construction of the robot are described in detail.

2.1 Selecting the Appropriate Design

The first step in designing a balancing robot is to select the appropriate structure according to the purpose of its controller. For this purpose, in this research, a platform-type two-wheeled balancing robot has been built in order to apply controllers and study the efficiency of each of them on the balance of the robot. The prototype was designed in the 3D design engineering software SolidWorks 45. The remaining steps, including the selection of electronic parts, construction materials, construction methods, etc., are described step by step.

2.2 Robot Dimensions

Since the robot was built solely to study control methods, an attempt was made to design a robot with small dimensions and, as a result, low cost, so that the robot operators have the ability to apply appropriate torque to maintain the balance of the robot. This robot was designed with a simple floor plan to place the robot's electronic components, so that the dimensions and spacing of the floors and the selection of components for each floor were carried out taking into account the principle that the center of mass of the robot should be in its upper part and the aim is to control the imbalance of the two-wheeled robot.

2.3 Robot Shape

In order to achieve optimal performance of the robot, a solid and symmetrical framework has been used in its structural design in SolidWorks software. This platform has the ability to increase floors and place various sensors and electronic components, as well as cameras for image processing. Therefore, this robot will have the ability to increase capabilities in all fields, including automatic navigation and recognition of the surrounding environment, which in this research is only concerned with controlling the balance of the robot. The final model of the two-wheeled balancing robot that was carried out in SolidWorks software is shown in Fig. 1.

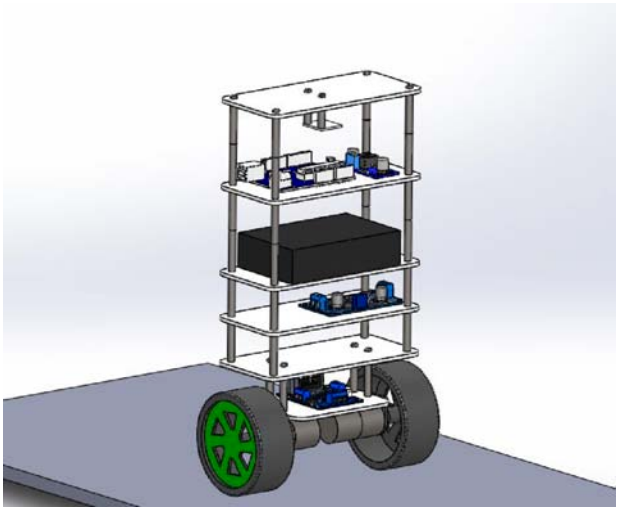


Fig. 1: Designed Structure of Two-Wheeled Balancing Robot in Solidworks Software

2.4 Material Selection

Based on the ability to cut with a laser cutting machine and easy drilling to place robot parts, Plexiglass 46 material was used to make it. The advantages of this material include low density, good resistance to thickness, and accurate cutting

based on the output map from SolidWorks software for use in two-wheeled balancing robot.

2.5 Selecting Electronic and Mechanical Parts of the Robot

To build a high-performance robot, selecting the right parts that have the best performance in a set is one of the most important parts of building a robot. The selection of electronic and mechanical parts of the robot should be based on the defined performance for that part, and if the performance is similar, the best part can be selected based on their strengths and weaknesses. Ultimately, excellent choices can create a high-performance robot. Of course, cost is also an important factor in selecting parts because a cheap part can partially meet the performance of an expensive part.

2.5.1 Selecting the Type of Actuator

In selecting the right actuator for the two-wheeled balance robot, DC motors with a gear box and a Hall effect encoder have been used to build the robot. High-speed DC motors, if used with a gearbox, provide high torque with a small volume to the robot. High torque with high reaction speed is needed by two-wheeled balancing robots to maintain their balance. On the other hand, to adjust the speed and find out the speed of the tires and calculate the linear speed of the robot, it is necessary to use an encoder on the output shaft axis of the robot so that the robot can move at the desired speed. The initial position and distance traveled can also be one of the encoder outputs, which in this research we only need to maintain balance and the distance, speed and initial position are not design considerations. So, the geared high-speed DC motor is the best option to be in the built robot set, but to continue the research and practical development of the robot, geared DC motors with Hall effect encoders have been used to build the robot.

2.6 Construction

After designing the three-dimensional model of the two-wheeled balancing robot and also choosing the material, we used plexiglass material here for ease of cutting and complete compliance with the output design from the design software.

To make the platform prototype, various parts of the robot are cut. Due to the choice of plexiglass, the need for high precision in cutting parts, and the elimination of machining operations, laser cutting has been used in cutting various parts of the robot. The laser cutting machine is very important in carrying out specific cutting orders because this machine is used with the most up-to-date systems for laser operations. Laser cutting machines are among the most popular machines and are used in most cases for precise cutting. After cutting the parts and assembling them all with electronic and mechanical parts such as tires and robot shafts, the finally built model is shown in Fig. 2, which are displayed side by side to match the output design from the design software.

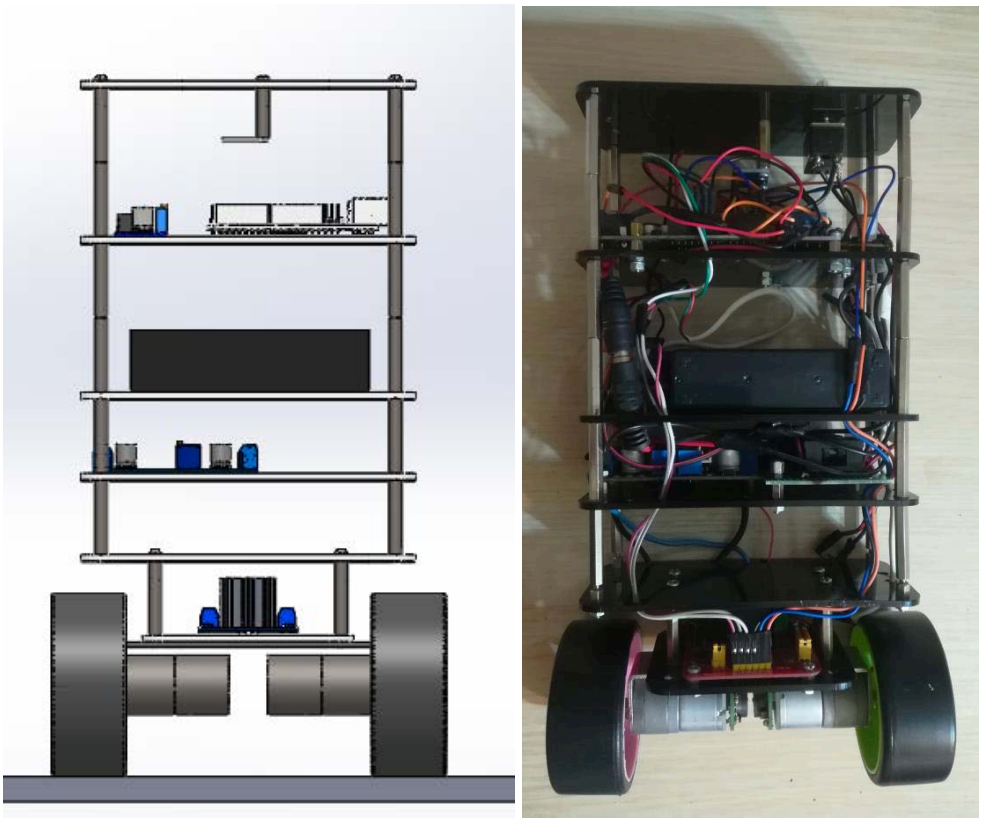


Fig. 2: Example of a Two-Wheeled Balancing Robot Platform

2.7 Control

Two control methods have been used to control the balance of the two-wheeled robot to compare the efficiency of these methods in controlling the robot and the results obtained show which method shows the desired performance in which conditions.

The methods used are one PID method assuming the linearity of the problem and the other fuzzy logic considering the nonlinearity of the problem. The principles that we have considered in controlling the robot are as follows:

- The two-wheeled balancing robot system is considered as a black box without considering the equations governing the balance of the robot.
- The surface intended for robot navigation is an uneven surface.
- Obtaining PID coefficients and the range of input and output values in fuzzy logic has been done experimentally
- The method used in fuzzy logic is Mamdani method

2.7.1 PID Controller

In this method, considering the availability of the built platform of the two-wheeled balancing robot, we obtained PID values by trial and error, then by entering the impact type input, we plotted the robot's balance graph with the output data from the MPU 6050 sensor and communicating with the computer via the serial port in the Arduino software. The steps for obtaining the optimal PID coefficients experimentally are as follows: first, we started with the proportional coefficient (P), as we know, the effect of this coefficient is that it increases the speed of the system and reduces the steady-state error to some extent, but does not make it zero. We know that to maintain the balance of the built two-wheeled balancing robot, the error must tend to zero, so the speed coefficient P alone will not be effective and other coefficients will need to be adjusted. Using a large coefficient. It increases the tendency to maintain balance, so the torque applied by the motors for even a small impact will be large and at high speed to create imbalance in the robot, and this in itself is a factor in creating imbalance in the robot.

The integral coefficient or I also makes the steady-state error zero, but it adds a large amount of unwanted oscillations to the transient response, and this in turn causes imbalance in the robot itself in situ. The derivative coefficient also weakens the oscillations of the transient response, so in addition to using the integral coefficient, we will need to use the derivative coefficient.

2.7.2 Fuzzy Controller

The Mamdani method has been used to implement the fuzzy logic controller. In this method, according to Figs. 3 to 5, the membership functions must be defined at first. We have considered two inputs as inputs for fuzzy control; One is the angle value at any moment and the other is the change in angle at any moment, which Fig. 3 shows three membership functions, and one is the negative angle (ND) which is a trapezoidal membership function from negative infinity to -2, and the other is the angle near zero (ZD) which is again defined as a trapezoidal function from 5 to -5. There is a range between these two functions, and if the angle is in that part, that is, between -5 and -2, depending on the membership level, from a value between zero and one, it may be 0.7 of a negative angle and 0.3 of a zero angle, which is the same logic as the logic of human reasoning, and uncertainty is involved here, and its difference from zero and one or binary logic is evident here. The membership function of the positive angle (PD) is also considered from zero to infinity. In fact, for the angle input at any moment, we have considered three membership functions that are overlapping functions. The more or less the overlap, the different results of the fuzzy logic. This case, Fig. 3, provides the best result for the defined fuzzy logic controller through trial and error for the first input.

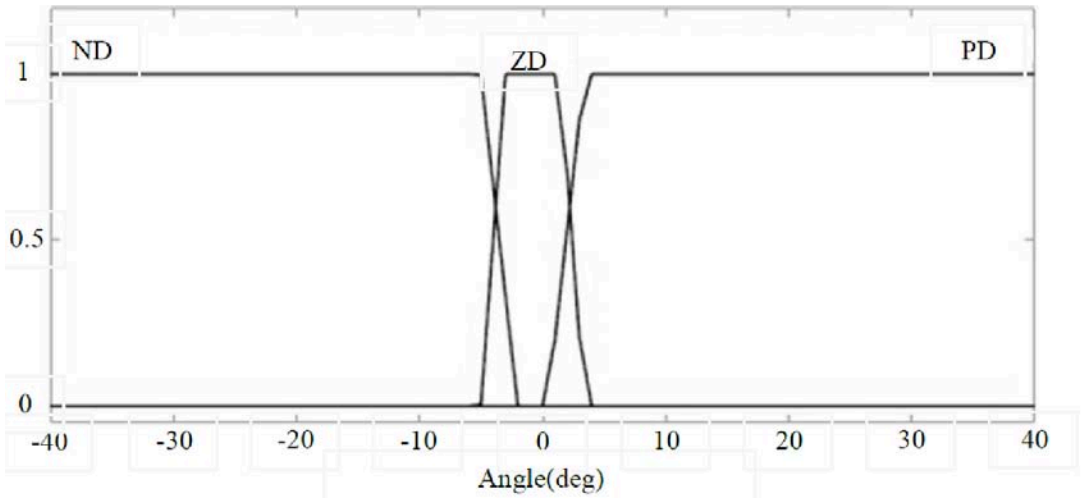


Fig. 3: Narrator Input Membership Functions

Fig. 4 show the amount of angle change at any moment, which is measured by the MPU 6050 sensor in a fraction of a second and with the smallest angle change and is used by the fuzzy logic controller. Here, the size of the upper side of the trapezoid is considered to be much smaller, which means that with this small value, the robot does not show any reaction. If this value is taken large, the robot oscillations will be greater.

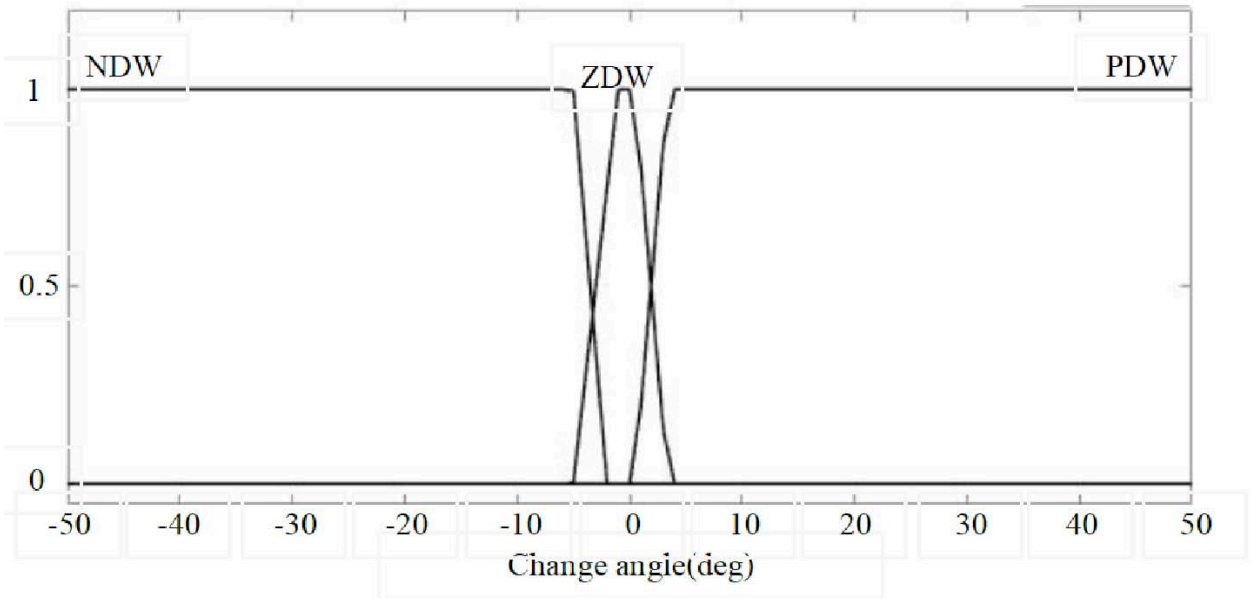


Fig. 4: Membership Functions of the Angle Change at any Moment

Fig. 5 shows the output membership functions to be applied to the actuators. These functions are trapezoidal in type and the values of the horizontal axis indicate the output of the fuzzy logic controller due to the reasoning that is written from the rules. These values are the same values that must be sent to the actuator driver so that the drivers can send the required speed and torque to the actuators with the PWM wave and the robot can maintain its balance. It is worth mentioning that the program written for the MPU 6050 sensor uses the I2C interface and is taken by the Kalman filter of the sensor noise and calculations are performed at high speed so that the robot can react quickly. Here, three membership functions are defined: the first is negative speed, which means the direction of the speed is counterclockwise, and the second is zero speed, which if we look carefully is considered to be completely zero in a very small range, and in the rest of the zero-speed membership function, the membership rate is raised. For positive speed, the function is again trapezoidal and its overlap with the zero-speed membership function is symmetrical with the overlap of the negative speed membership function and the zero function.

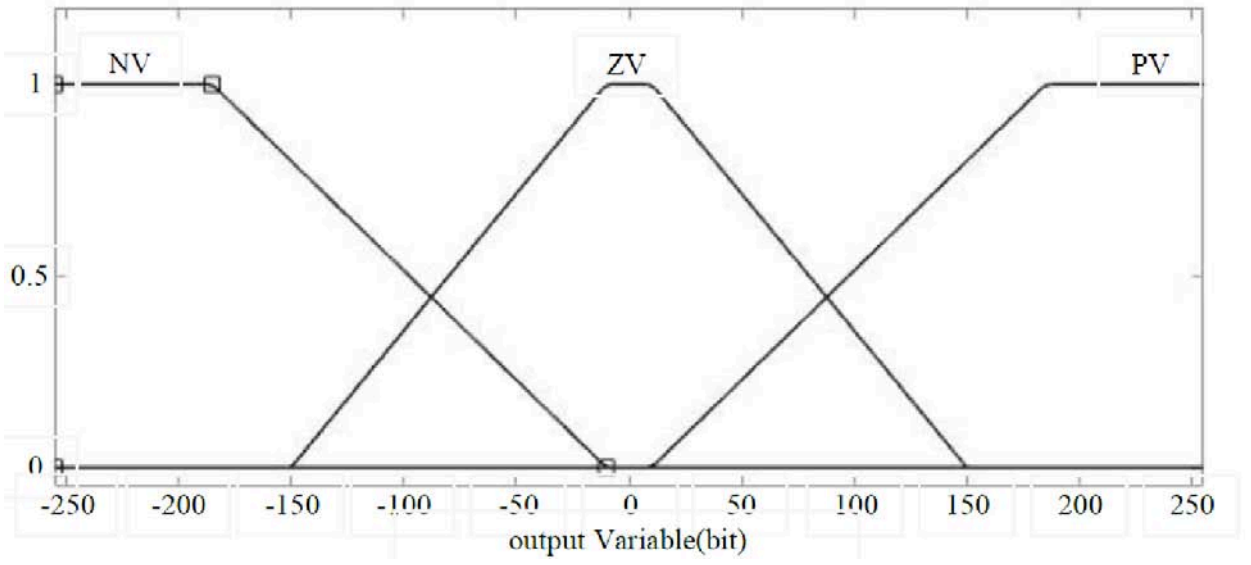


Fig. 5: Velocity Membership Function

III. RESULTS

In this section, the performance comparison of two PID controllers and fuzzy logic is used for the robot that was built. This robot, as described in the details of the construction in the third chapter, has the ability to apply two controllers through the Arduino software on the Arduino microcontroller. To adjust the coefficients and values of the controllers manually and experimentally, by trial and error and observing the equilibrium output diagram of the angle-type robot, we tried.

3.1 Results of the PID Controller

In the first stage, we use only the P coefficient for the PID controller and the rest of the coefficients are given zero, which is completely related to the physics of the system and we obtain the coefficients empirically and we have no problem with the complexity of physics.

Fig. 6 shows the fact that the robot actually falls and using the P coefficient alone cannot control the balance of the robot.

In the second stage, by increasing the proportional coefficient, we tried to test the balance response of the robot with respect to the value of this coefficient, the result of which is plotted in Fig. 7.

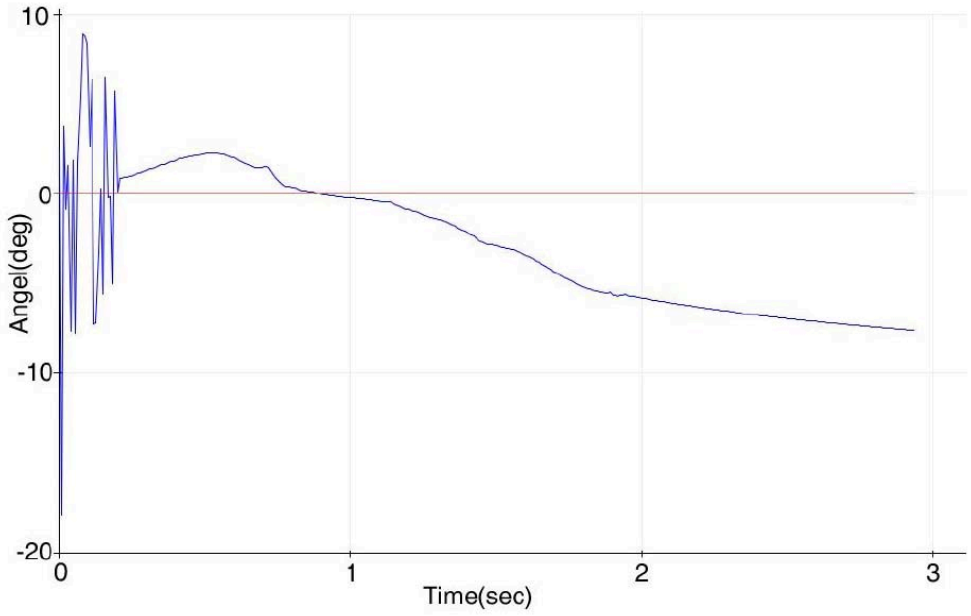


Fig. 6: Control Diagram of the Robot in the First Stage

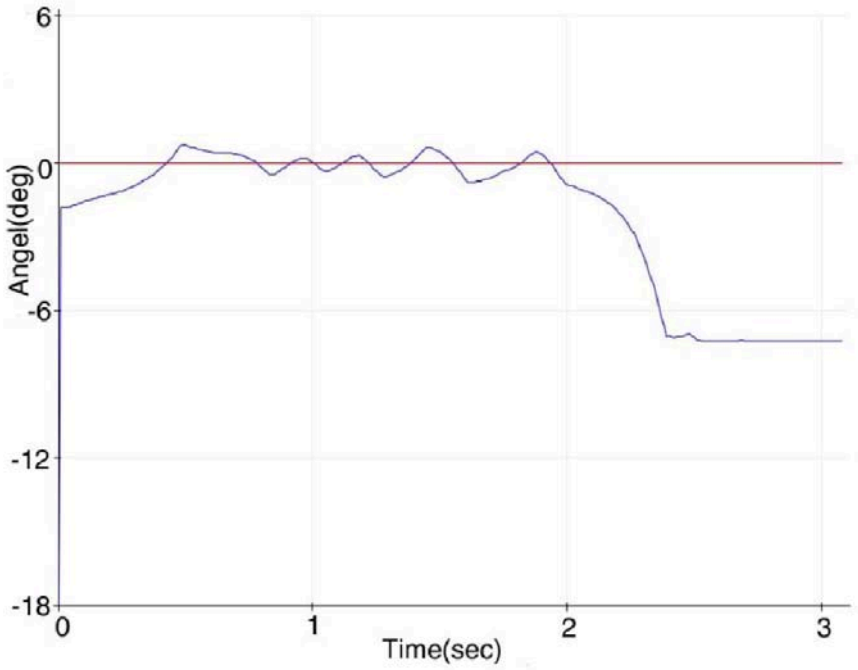


Fig. 7: Robot Control Diagram in the Second Stage

In Fig. 7, it can be seen that the presence of the P coefficient without other coefficients and simply increasing the P coefficient value to 100 causes the robot to oscillate sharply and fall. Although it has an initial balance of up to 2 seconds, we eventually have the robot fall.

Considering that in the controller, the proportionality of the P coefficient alone causes a large error and, as a result, a large signal to the input, therefore, in Fig. 7, an equilibrium is initially created, which, as the error increases or the difference between the angle value and the angle, a large signal reference is created and applied to the wheels, and therefore the robot falls.

Therefore, we must also use other coefficients, and for now, we will suffice with this 2-second equilibrium for the P coefficient value and observe the effect of other coefficients on the robot balance control.

In the third step, we use the integrator coefficient or I with a value of 60 for the controller and set the coefficient value to 100 and the derivative coefficient or D to zero to measure the effect of the integrator coefficient along with the proportional coefficient on the output graph of the MPU 6050 sensor.

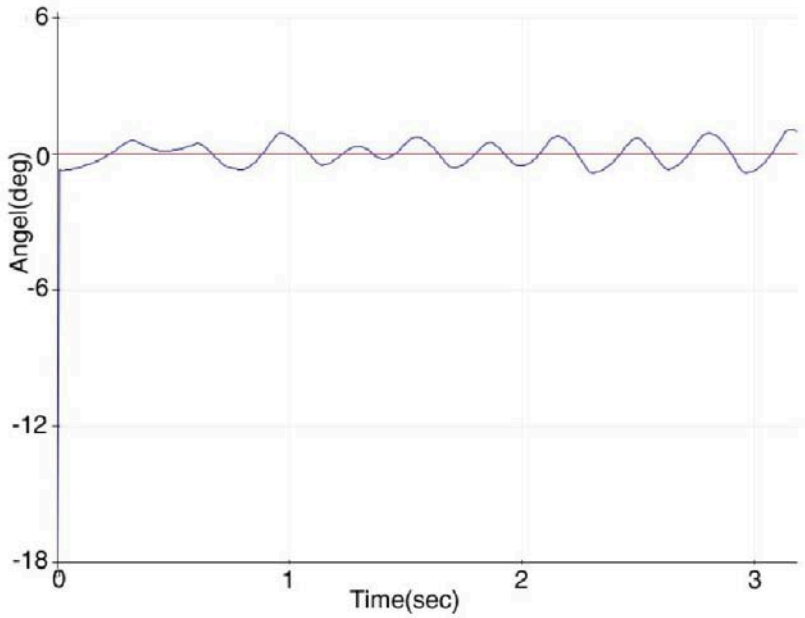


Fig. 8: Robot Control Diagram in the Third Stage

Fig. 8 shows the robot's strong oscillation again around the balance point and the possibility of the robot falling with the slightest impact. In this stage, along with the proportional coefficient, the integral coefficient was also used. The desired result, which is the robot's balance around the zero point, was not observed and severe oscillations were applied to the robot and the controller could not maintain the robot's balance. In the fourth stage, we use the value 100 for the proportional coefficient and 400 for the integral coefficient. Fig. 9 shows the robot's balance diagram around the balance point. It is true that the integral coefficient creates the robot's balance around the robot's balance point with a permanent oscillation, but in a two-wheeled robot, the goal of balance is to settle down and the robot's balance angle tends to zero, which is not possible with the proportional-integral coefficient according to the results obtained.

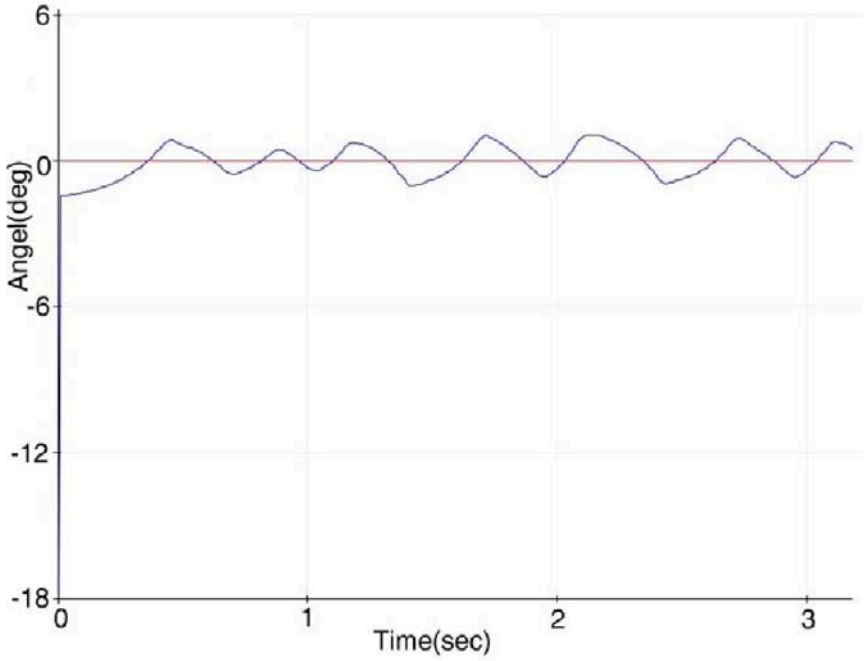


Fig. 9: Robot Control Diagram in the Fourth Stage

In this stage, by adding a derivative coefficient to the proportional-integral controller, we seek to reduce the robot's oscillations and the robot's ability to respond to impact input to maintain balance.

For fast response and large torque against impact, we increased the proportional coefficient to 120 and used the same integral coefficient of 400 so that oscillations do not increase, and we applied a derivative coefficient of 1. The equilibrium diagram in Fig. 10 is drawn for the robot with these control values, which shows the reduction of oscillations and the robot's tendency to maintain balance.

In the sixth stage, by only increasing the derivative coefficient to 3.5 and keeping the other values constant, we see in Fig. 11 that the robot has found a smaller amplitude of oscillations around the equilibrium point. But it still has not found complete equilibrium around the equilibrium point, meaning it has oscillations in place. We used all the coefficients in the stage, but we still need to modify and change the coefficient whose high value can cause the robot to oscillate.

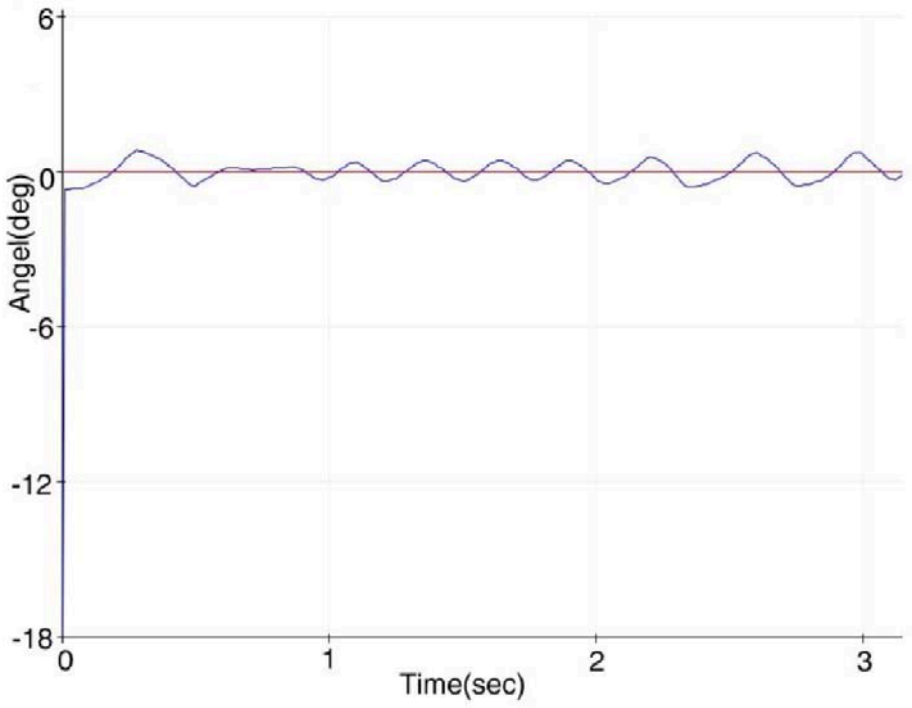


Fig. 10: Control Diagram of the Robot in the Fifth Stage

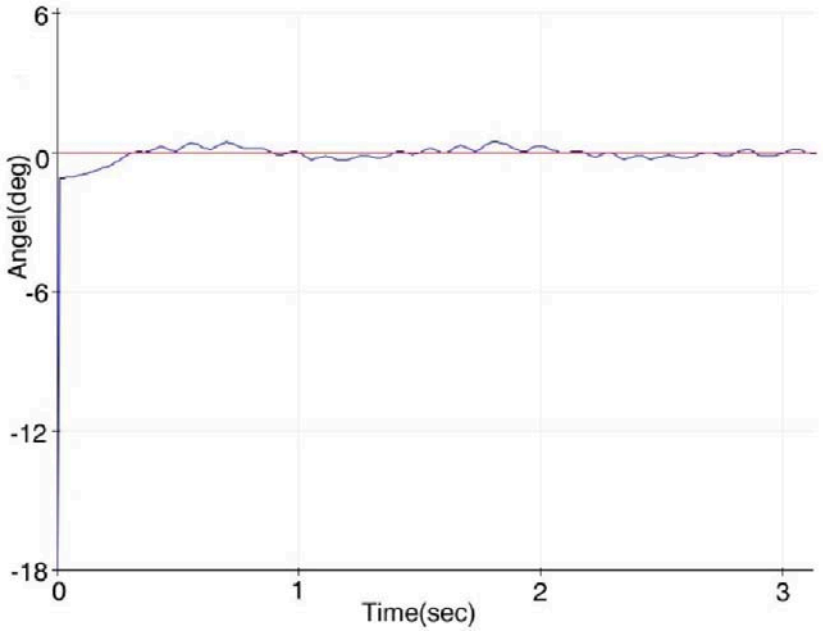


Fig. 11: Robot Control Diagram in the Sixth Stage

In the last stage, by reducing the value of the integral coefficient to 3.5, the amplitude of oscillations is greatly reduced and according to Fig. 12, the robot even reaches equilibrium. At this stage, the remaining coefficients have the same value as before.

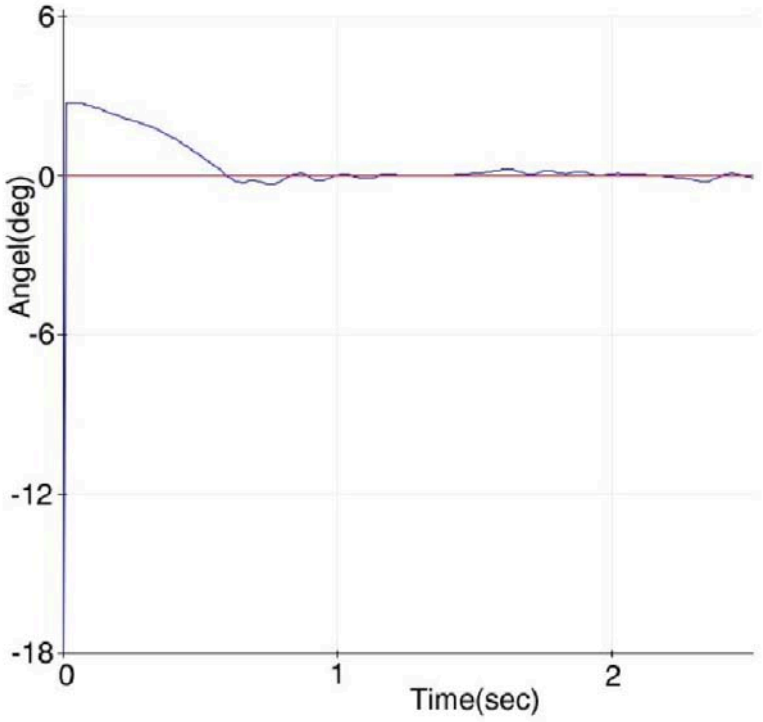


Fig. 12: Robot Control Diagram in the Seventh Stage

Using the trial-and-error method and repeated iterations, we finally reached a desirable result for the PID coefficients, the results of which can be seen in Fig. 13, and brought better stability and stability to the two-wheeled balancing robot.

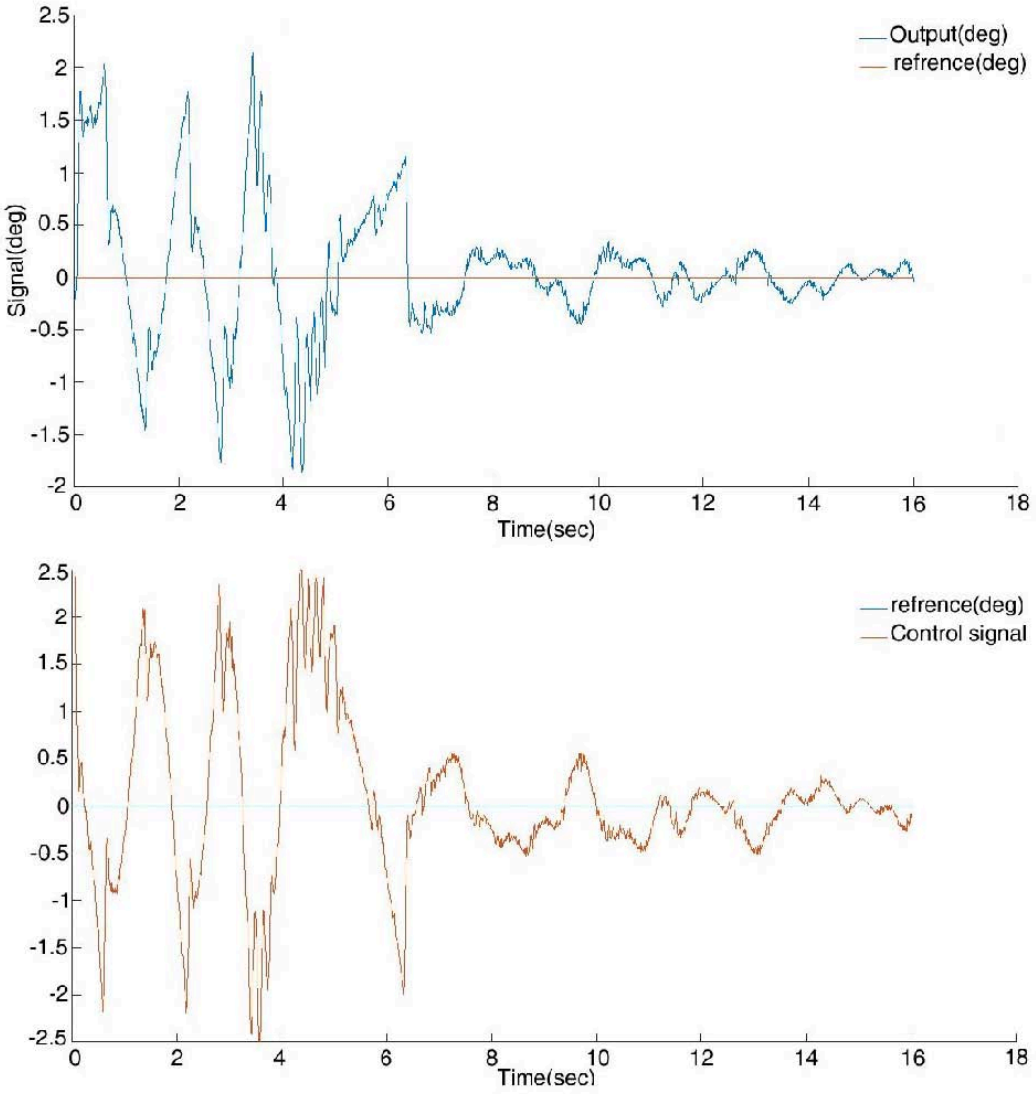


Fig. 13: Robot Control Diagram in the Best Result Obtained for PID

In Fig. 13, the output is the final output and the balance angle of the robot, and the reference diagram is the balance line. The control signal is the PWM signal sent from the Arduino microcontroller to the motors. As can be seen, the control signal is in the opposite direction of the control signal at any moment, and this process will continue until equilibrium is reached.

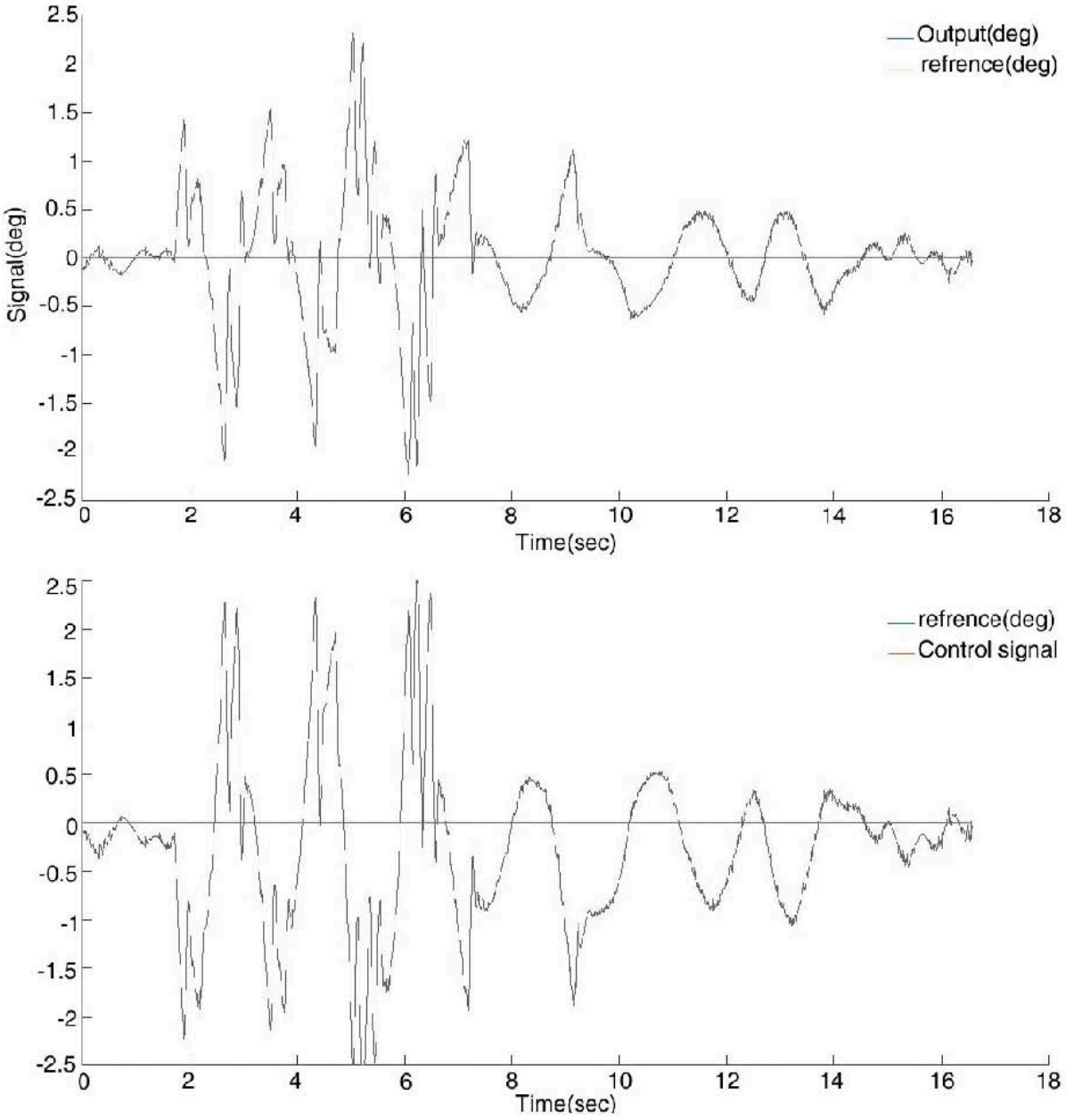


Fig. 14: Balance Diagram of PID Controller on Two-Wheeled Balancing Robot with External Disturbance

Fig. 14 shows the reaction of two-wheeled balancing robot to impact with PID controller and the coefficients obtained in Fig. 13. As can be seen from the diagram, by applying impact to the robot, the robot takes about 12 seconds to reach equilibrium, but it does not overturn and regains its balance due to the movement and torque of the motors opposite the angle of the robot.

According to the output diagrams from the MPU 6050 sensor of the robot with PID controller, the result obtained is that we need all three coefficients to control the robot, and all three coefficients must be obtained by trial and error or by other methods that can optimize the PID coefficients. The experimental method is a very tedious method and may not achieve the desired result, and it is better to use the PID method together with other methods. On the other hand, the type of motors used, the gearbox backlash, and the design of the structure can be other factors that complicate the control of the robot. On the other hand, this method is not resistant to impacts and uneven surfaces, despite not

reducing the control error to below one degree, but it has the ability to control and respond quickly to impacts.

3.2 Result of the Fuzzy Controller

By applying a fuzzy logic controller that has a nonlinear nature, we have tried to compare the results of the linear PID controller with the nonlinear fuzzy controller. In this controller, we used the Mamdani method with trapezoidal functions and obtained the values of the functions by trial and error according to the PID controller. Here, we have presented the graph of the best results for the fuzzy controller.

Fig. 15 is the result of the fuzzy control on the balance of the two-wheeled balance robot. As can be seen, the robot reached equilibrium within two seconds. This controller, with its nonlinear nature, has been able to balance the nonlinear system of the two-wheeled balance robot well.

Fig. 15 is the initial balance of a two-wheeled balancing robot on which the fuzzy control algorithm has been applied.

3.3 Comparison of Results

In the research, we want to make a comparison regarding the performance of two PID and fuzzy controllers applied to the robot because two controllers on the robot cannot perform the control action at the same time.

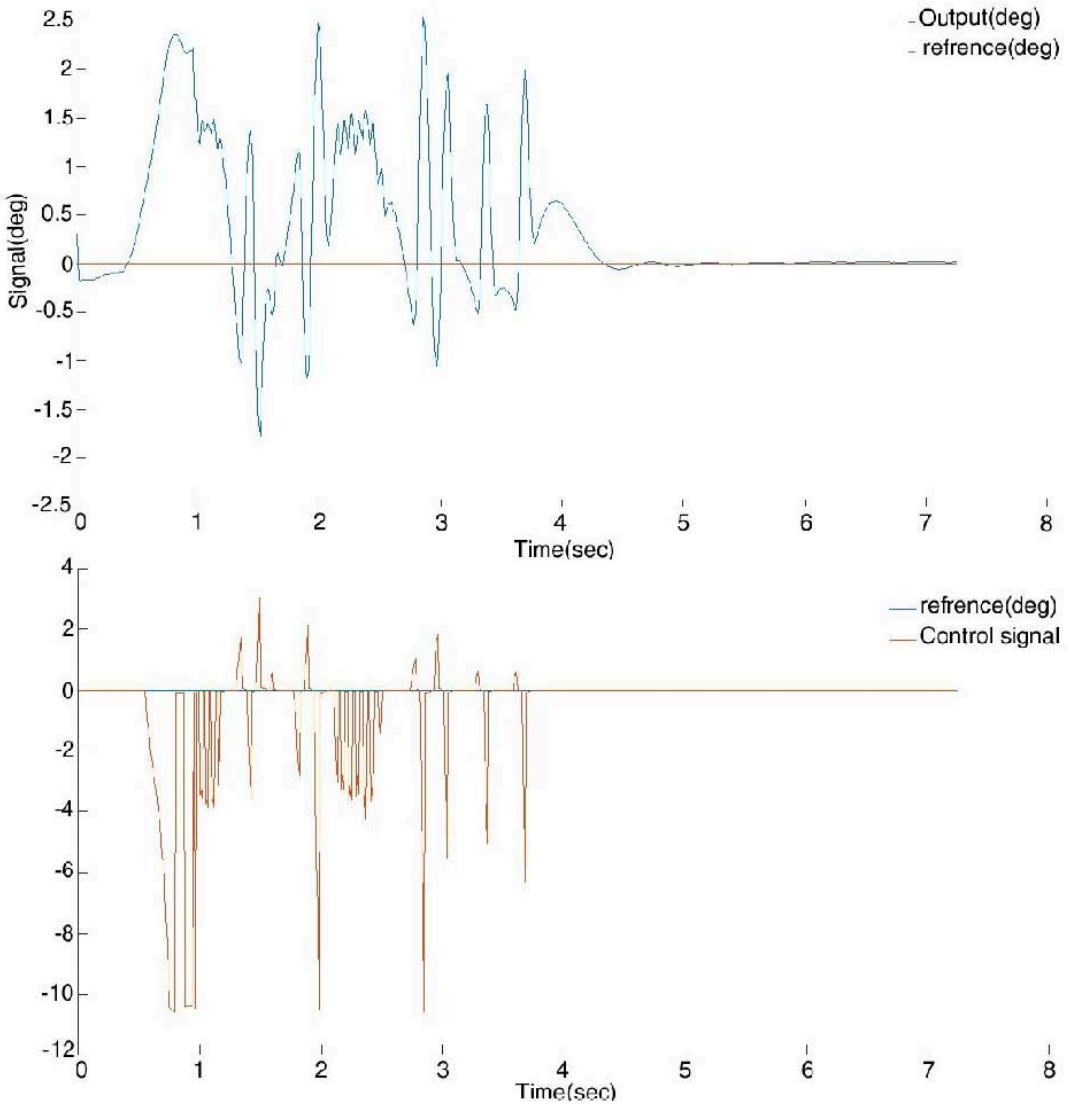


Fig. 15: Fuzzy Control Balance Hairpin on a Two-Wheeled Balancing Robot

In Fig. 16, we have brought both balance diagrams of the robot from Figs. 13 and 14 together so that the comparison can be done easily and accurately and the result of the comparison between the two fuzzy and PID controllers can be easily seen. Therefore, it can be seen that although the PID controller has been able to maintain the balance of the robot, this balance is not complete and sometimes the robot goes out of balance and the robot is oscillating and vibrating around the balance angle and even compared to the fuzzy controller whose result is shown in Fig. 14, it has not reached balance quickly. This is because the fuzzy controller reaches equilibrium much faster and stands completely still and without oscillation at the equilibrium point. This is because the fuzzy controller uses human logic (but) and (if) in the control discussion and has inherent flexibility in controlling nonlinear systems.

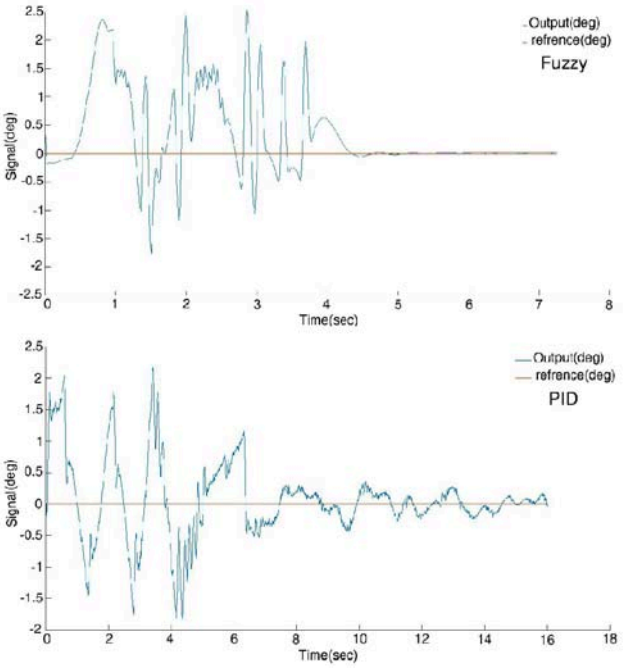


Fig. 16: Comparison Diagram of Balance of Two-Wheeled Robot with Fuzzy and PID Controllers and in Initial Start Mode

IV. CONCLUSION

The two-wheeled balancing robot system is such that it is involved in many factors to achieve the balance of the robot, including the unevenness of the surface below, the tilt of the robot, the friction surface between the wheels and the surface below the wheels. All of these factors cause a mismatch between the output and input applied to the wheels (the same torque applied by the motors.

Therefore, despite the nonlinear nature of the system in this study, we have considered the system in a linear range. However, due to the nonlinear nature of the two-wheeled balancing robot system, as can be seen, the fuzzy logic controller shows much better performance than the PID controller in the results section and the balance of the robot is maintained without oscillation. On the other hand, the adjustment of values in the fuzzy controller is done much faster than the PID controller in this study, and this is one of the other advantages of the controller. It is fuzzy. The fuzzy controller, with its nature of being close to human logic, shows high flexibility in behavior and decision-making, just as humans are very successful in maintaining their balance when walking, standing, or receiving an impact, etc. Because it does not operate in a zero-one logic in decision-making and has a logic similar to fuzzy logic, in fact, fuzzy logic reflects the logic of human behavior when making decisions, and this research shows this issue well by comparing the results that only three fixed coefficients in PID have less ability than fuzzy logic in controlling the balance of a two-wheeled robot.

A two-wheeled balancing robot uses less energy than other wheeled robots and has advantages such as high maneuverability in a small space due to rotation, the ability to increase capabilities such as recognizing the surrounding environment and other unique capabilities. The need for study in the field of balance control is still felt. Therefore, this research focuses more on the balance of the robot, and the research results show that the balance of the robot can be well maintained by applying optimal controllers.

These types of robots can function as humanoid robots by upgrading the robot structure and adding processing and images to identify the surrounding environment and various sensors and sound recognition algorithms, and can be used as assistants for the disabled, or by changing the shape of the tires and using powerful actuators, they can also be used in uneven environments. One way to increase the ability to maintain balance in two-wheeled robots is to use learning algorithms along with a controller to increase the flexibility of the robot and the ability of the robot to encounter balance-disrupting factors each time.

Data Availability Statement: The data that support the findings of this study are available on request from the corresponding author.

Funding Statement: This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflict of Interest Disclosure: The authors declare that they have no conflict of interest.

Ethics approval statement: Not applicable.

Patient consent statement: Not applicable.

Permission to reproduce material from other sources: Not applicable.

Clinical trial registration: Not applicable.

REFERENCES

1. Ogata K, Yang Y (2002) Modern control engineering. Vol. 4. Prentice hall India.
2. Bature AA, Buyamin S, Ahmad MN, Muhammad M (2014) A comparison of controllers for balancing two wheeled inverted pendulum robot. International Journal of Mechanical & Mechatronics Engineering 14(3):62-68.
3. Grasser F, D'Arrigo A, Colombi S, Rufer AC (2002) JOE: a mobile, inverted pendulum. IEEE Transactions on Industrial Electronics 49(1):107-114.
4. Chan RPM, Stol KA, Halkyard CR (2013) Review of modelling and control of two-wheeled robots. Annual Reviews in Control 37(1):89-103.
5. Azimi MM, Koofigar HR (2013) Model predictive control for a two wheeled self balancing robot. In: Robotics and Mechatronics (ICRoM), 2013 First RSI/ISM International Conference on, IEEE.
6. Brisilla R, Sankaranarayanan V (2015) Nonlinear control of mobile inverted pendulum. Robotics and Autonomous Systems 70:145-155.
7. Sun L, Gan J (2010) Researching of two-wheeled self-balancing robot base on LQR combined with PID. In: Intelligent Systems and Applications (ISA), 2010 2nd International Workshop on, IEEE.
8. Solis J, Nakadate R, Yoshimura Y, et al (2009) Development of the two-wheeled inverted pendulum type mobile robot WV-2R for educational purposes. In: Intelligent Robots and Systems, IEEE/RSJ International Conference on, IEEE.
9. Yamafuji W, Kawamura R (1989) Attitude control of coaxial two-wheeled car [J]. Journal of Robots 17(4):74-79.
10. Kamida S (2002) Evaluation of self-discipline robot interacting with human [J]. Journal of Robots 20(3):1-9.
11. Jamil O, Jamil M, Ayaz Y, Khubab Ahmad K (2014) Modeling, control of a two-wheeled self-balancing robot. In: Robotics and Emerging Allied Technologies in Engineering (iCREATE), 2014 International Conference on, IEEE.
12. Peng K, Ruan X, Zuo G (2012) Dynamic model and balancing control for two-wheeled self-balancing mobile robot on the slopes. In: Intelligent Control and Automation (WCICA), 2012 10th World Congress on, IEEE.

13. Akesson J, Blomdell A, Braun R (2006) Design and control of yaip—An inverted pendulum on two wheels robot. In: Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, 2006 IEEE, IEEE.
14. Sun F, Yu Z, Yang H (2014) A design for two-wheeled self-balancing robot based on Kalman filter and LQR. In: Mechatronics and Control (ICMC), 2014 International Conference on, IEEE.
15. Raffo GV, Ortega MG, Madero V, Rubio FR (2015) Two-wheeled self-balanced pendulum workspace improvement via underactuated robust nonlinear control. Control Engineering Practice 44:231-242.
16. Grasser F, D'Arrigo A, Colombi S, Rufer A (2002) JOE: a mobile, inverted pendulum. IEEE Transactions on Industrial Electronics 49(1):107-114.
17. Ren T-J, Chen T-C, Chen C-JJCEP (2008) Motion control for a two-wheeled vehicle using a self-tuning PID controller. Control Engineering Practice 16(3):365-375.
18. Xu C, Li M, Pan F (2011) The system design and LQR control of a two-wheels self-balancing mobile robot. In: Electrical and Control Engineering (ICECE), 2011 International Conference on, IEEE.
19. Slavov T, Kralev J, Petkov PJCAS (2014) Linear-Quadratic Control of a Two-Wheeled Robot. Comptes rendus de l'Acade'mie bulgare des Sciences 67(8):1153-1158.
20. Zhang J, Zhao T, Guo B, Dian S (2021) Fuzzy fractional-order PID control for two-wheeled self-balancing robots on inclined road surface. Systems Science & Amp; Control Engineering 10(1):289–299. https://doi.org/10.1080/21642583.2021.2001768.
21. Mudeng V, Hassanah B, Priyanto YT, Saputra O (2020) Design and Simulation of Two-Wheeled Balancing Mobile Robot with PID Controller. International Journal of Sustainable Transportation Technology 3(1): 12-19. https://doi.org/10.31427/IJSTT.2020.3.1.3
22. Kumar A, Raj R, Kumar A, Verma B (2023) Design of a novel mixed interval type-2 fuzzy logic controller for 2-DOF robot manipulator with payload. Engineering Applications of Artificial Intelligence 123. https://doi.org/10.1016/j.engappai.2023.106329

Stamp Tests of Loess Soil Samples in a Tray

Dr. Aigul Budikova

Kyzylorda University

ABSTRACT

For construction conditions, when it is necessary to determine the expected amount of subsidence of small-width structures on high-capacity loess bases having a soaking area larger than the area of the base of their foundation due to prolonged moistening by filtration flow, the compression compression model for determining relative subsidence becomes unacceptable.

In addition, when determining the relative subsidence based on the results of compression tests, it is assumed that the soil of the loess base is deformed only in the vertical direction during moistening, while lateral deformations of the soil from the action of horizontal compacting stresses are not taken into account. However, as the results of comparing the calculations of many researchers of the expected subsidence with field data show, compression tests underestimate the additional deformations of structures that occur when their loess bases are moistened.

Keywords: NA

Classification: DDC Code: 624.151

Language: English

Great Britain Journals Press

LJP Copyright ID: 392915
Print ISSN: 2631-8474
Online ISSN: 2631-8482

London Journal of Engineering Research

Volume 25 | Issue 1 | Compilation 1.0

Stamp Tests of Loess Soil Samples in a Tray

ШТАМПОВЫЕ ИСПЫТАНИЯ ОБРАЗЦОВ ЛЁССОВЫХ ГРУНТОВ В ЛОТКЕ

Dr. Aigul Budikova

АННОТАЦИЯ

Для условий строительства, когда необходимо определить ожидаемую величину просадки сооружений небольшой ширины на лёссовых основаниях большой мощности, имеющих область замачивания больше площади подошвы их фундамента вследствие длительного увлажнения фильтрационным потоком, модель компрессионного сжатия для определения относительной просадки становится неприемлемой.

Кроме того, при определении относительной просадки по результатам компрессионных испытаний предполагается, что грунт лёссового основания при увлажнении деформируется только в вертикальном направлении, боковые деформации грунта от действия горизонтальных составляющих уплотняющих напряжений при этом не принимаются во внимание. Однако, как показывают результаты сравнения расчетов многих исследователей ожидаемой просадки с натурными данными, компрессионные испытания дают заниженные величины дополнительных деформаций сооружений, возникающих при увлажнении их лёссовых оснований.

ABSTRACT

For construction conditions, when it is necessary to determine the expected amount of subsidence of small-width structures on high-capacity loess bases having a soaking area larger than the area of the base of their foundation due to prolonged moistening by filtration flow, the compression compression model for determining relative subsidence becomes unacceptable.

In addition, when determining the relative subsidence based on the results of compression tests, it is assumed that the soil of the loess base is deformed only in the vertical direction during moistening, while lateral deformations of the soil from the action of horizontal compacting stresses are not taken into account. However, as the results of comparing the calculations of many researchers of the expected subsidence with field data show, compression tests underestimate the additional deformations of structures that occur when their loess bases are moistened.

Автор: Молдашевна, кандидат технических наук, ассоцированный профессор, Кызылординский университет имени Коркыт Ата, город Кызылорда, Республика Казахстан.

Author: Budikova Aigul Moldashevna, candidate of technical sciences, associate professor of the Department «Architecture and construction production», Kyzylorda University named after Korkyt Ata, Kyzylorda city, Republic of Kazakhstan.

I. INTRODUCTION

Вертикальная просадочная деформация по рекомендациям СП (свод правил) и ВСН (ведомственные и строительные нормы) подсчитывается по модели компрессионного сжатия (без возможности бокового расширения) небольшой мощности слоя грунта под подошвой фундамента сооружения. При этом предполагается, что замачивание происходит только под подошвой фундамента, а мощность лёссового основания невелика, так как в компрессионном приборе отношение высоты образца к его диаметру h/d= =3,7/8,75=0,42см. Для условий строительства, когда необходимо определить ожидаемую величину просадки сооружений небольшой ширины на лёссовых основаниях большой мощности, имеющих область замачивания

London Journal of Engineering Research

больше площади подошвы их фундамента вследствие длительного увлажнения фильтрационным потоком, модель компрессионного сжатия для определения относительной просадки становится неприемлемой. Кроме того, при определении относительной просадки по результатам компрессионных испытаний предполагается, что грунт лёссового основания при увлажнении деформируется только в вертикальном направлении, боковые деформации грунта от действия горизонтальных составляющих уплотняющих напряжений при этом не принимаются во внимание. Однако, как показывают результаты сравнения расчетов многих исследователей ожидаемой просадки с натурными данными, компрессионные испытания дают заниженные величины дополнительных деформаций сооружений, возникающих при увлажнении их лёссовых оснований [2].

Результаты исследований просадочных деформаций многих ученых показали, что их характер зависит от ширины подошвы фундамента сооружения. Под распластанными сооружениями просадка происходит без бокового расширения и в этих условиях компрессионные испытания дают результаты, сопоставимые с натурой. Но под узкими фундаментами деформация просадки происходит с боковым выпором грунта. Как показали опыты, проведенные Н. Н. Фроловым [2], фактическая просадка сооружений (в г. Никополе) оказалась в 2-2,8 раза больше, чем расчетная с учетом данных компрессионных испытаний. Причину таких несовпадений автор объясняет наличием бокового выпора грунта, не учитываемым при расчете. Следует, однако, отметить, что сравнение величин вертикальных и горизонтальных перемещений, по данным С.Г.Юрченко [1], показывает, что боковые перемещения значительно меньше вертикальных и составляют около 15% от их величины. Расхождение между фактическими и расчетными величинами просадки автор объясняет тем, что просадка по своей природе носят явно пластический характер, а не является деформацией уплотнения, как принято в нормативных документах, и, следовательно, компрессионные приборы не способны моделировать работу грунта в фазе пластических деформаций [4].

Для приближения модели лабораторного исследования просадочных свойств лёссового грунта к реальным условиям их работы под неширокими гидротехническими сооружениями гидромелиоративных систем и для получения качественной картины деформированного состояния образцов лёссового грунта после их замачивания под нагрузкой нами были проделаны штамповые испытания такого грунта в лотке. Поскольку в реальном грунтовом основании под сооружением (в зависимости от соотношения сторон подошвы фундамента и сжимаемой толщи грунта), кроме компрессионного, может быть и другое деформационное состояние с возможностью бокового расширения грунта и областью замачивания за пределами контура фундамента, необходимо было размеры основания под штампом, имитирующим подошву фундамента строительных сооружений, принять в несколько раз больше размеров штампа как в ширину, так и в глубину для того, чтобы не было влияния стенок и дна лотка на возможное расширение грунта под действием нагрузки и увлажнения [2].

Так как в лотке для штампового испытания использовались образцы натурного грунта естественной структуры и напряжения σ_z, передаваемые штампом на основание, равны реальным, то и его относительная деформация ε_{sl} будет соответствовать реальной.
Нами были проделаны два вида испытания образцов лёссового грунта в лотке:

1. Имитация плоской модели сооружения.
2. Имитация осесимметричной модели сооружения.

В первом случае лоток имел размеры bxlxh=10x55x60см; штамп имел размеры 10x10см, ширина лотка была равна ширине штампа, имитирующего ширину подошвы

сооружения, а длина лотка была принята в пять раз больше ширины штампа. Во втором случае лоток имел размеры bxlxh=30x42x45см и для передачи нагрузки на грунт использовался круглый штамп диаметром d=7см.

Высота образцов лёссового суглинка была равна 35см, ниже лотки были заполнены плотным мелким песком. Размеры лотков (а, следовательно, и монолита грунта) назначены таковыми, чтобы они соответствовали возможным контурам замачивания грунта основания от имитируемого относительно неширокого гидросооружения.

Одна сторона лотков была выполнена из прозрачного оргстекла. В лоток помещался образец натурного грунта в виде монолита в естественном по ориентации состоянии. На лицевой стороне монолита, примыкающей к оргстеклу, прежде чем помещать его в лоток, наносилась несмываемой краской сетка размерами 2x2см. Во избежание застенной фильтрации передняя поверхность монолита тщательно выравнивалась для наиболее плотного прилегания к лицевой стороне ящика. Прозрачная стенка необходима для возможности визуального фиксирования изменения контура промачивания грунтового основания под штампом, а также изменение его деформативного состояния после приложения к нему через штамп постепенно увеличивающейся ступенями нагрузки. Лоток размещался на металлическом столе с нагрузочной системой, позволяющей передавать на образец грунта нагрузку так называемым «мертвым грузом» ступенями, начиная с p=0,0065МПа, через жесткий штамп толщиной t=1см. Штамп с нагрузочной системой соединялся жестко. Для замеров перемещения лёссового грунта от нагрузки и при замачивании применялись прогибомеры ЛИСИ. Общая картина - деформированный грунт - снималась на кальку и фотографировалась. На рисунках 2.2, 2.3, 2.4, 2.5 и 2.6 показаны стенд и испытуемый грунт в нем до и после нагружения и замачивания [1].

Рисунок 1: Вид установки и монолит лёссового грунта до нагружения и замачивания

а)

б)

Рисунок 2: а) картина деформирования лёссового грунта при замачивании после нагружения p=0,1МПа, w_{sl}=16% (плоская модель); б) вид плоской модели сверху

а) – при нагрузке p=0,02МПа

б) – при нагрузке p=0,08МПа

Рисунок 3: Вид осесимметричной модели сверху

После суток консолидации грунта под нагрузкой и снятия начального отсчета по индикатору вокруг штампа заливалось строго дозированное количество воды. Количество воды определялось с учетом изменения пористости образцов грунта после замачивания. Для этого взвешивали монолит грунта природной влажности, помещаемый в лоток, m_{go}=22770г и определяли массу частиц: $m_s=\rho_s v_s$=2,73x7804,5=21306,2, г.

В таблице 1 показано количество воды, которое доливалось в лоток для плоской модели при замачивании образца до нужной влажности:

Таблица 1: количество воды долитой в лоток для плоской модели при замачивании образца

Влажность замоченного грунта, w_{sl}, %	Масса долитой воды, m_w, кг	
	Плоская модель	Осесимметричная модель
16	1,945	2,55
21	3,010	3,98
26	4,075	5,98

При этом производился отсчет по индикатору для определения величины просадки. С продвижением контура увлажнения под штампом вглубь монолита по координатной сетке определялась мощность замоченного слоя. Результаты эксперимента при разных значениях давления и влажности приведены в таблице 2.

Таблица 2: Результаты лабораторного испытания образца в лотке при разных давлениях и влажности

Влажность w, %	Просадка при разных давлениях S_{sl}, см,		
	p=0МПа	p=0,05МПа	p=0,1МПа
8	0	0,3	0,5
16	0,7	1,4	1,8
21	1,8	2,7	3,1
23	2,5	3,7	4,9
26	4,2	6,1	7,9

Кроме того, проводились определения величин относительной просадки при испытаниях образцов лёссового суглинка, замоченных до различных влажностей, при испытаниях их в плоской и в осесимметричной моделях. Результаты экспериментов приведены в таблице 3.

Таблица 3: Величины относительной просадки, полученные при испытаниях лёссового суглинка на различных моделях

w, %	Относительная просадка ε_{sl} при испытаниях грунта в плоской модели и осесимметричной									
	p=0МПа		p=0,04МПа		p=0,05МПа		p=0,1МПа		p=0,2МПа	
	плос-кая	осе-сим	плос-кая	осе-сим	плос-кая	осе-сим	плос-кая	осе-сим	плос-кая	осе-сим
8	0	0	0,55	0,75	1,9	2,3	2,5	3,4	3,1	5,3
16	0,3	0,4	1,6	1,9	2,9	3,4	3,7	5	4,3	7,5
21	0,5	0,61	2,5	2,7	3,7	4,5	5,2	6,8	8,7	9,4
23	0,71	0,71	3,9	3,6	4,5	4,5	7,6	7,6	10,5	11,7
26	0,92	0,92	4,2	4,7	4,9	5,1	8,5	9,2	12,9	13,5

Как показала раскопка грунта в лотке после завершения эксперимента, влажность грунта сразу под подошвой штампа и чуть ниже от него была больше влажности ниже лежащих слоев грунта. Как видно из рисунки 1,2,3 области пластической деформации возникают прежде всего сразу под штампом и под краями его и далее по мере продвижения фронта увлажнения область деформированного грунта распространяется вниз на глубину, равную примерно 2,5в и на 1,5в в каждую сторону от штампа, где в- ширина штампа.

Как показал анализ результатов экспериментов, существует предельное

значение влажности, после достижения которой величина деформации грунта резко возрастает и происходит пластическое течение грунта, хотя нагрузка при этом не увеличивалась и оставалась постоянной. По достижении этой влажности штамп резко опускается вниз и вокруг него образуются бугры выпирания. Причем такая же картина наблюдается как при испытании лёссового грунта в плоском лотке, так и при осесимметричном испытании (рисунка 3(б). При этом, как показали эксперименты, картина разрушения при достижении критической влажности w=19% в осесимметричном лотке наступает при нагрузке p=0,07МПа, а при испытании в плоском лотке значение критической влажности мало меняется (w=16%), а критическая нагрузка увеличивается до значения p=0,1МПа. Дальнейшее увеличение влажности вызывает непрерывное возрастание деформации пластического течения, т.е. в отличие от непросадочных грунтов, где потеря прочности наступает при достижении второй критической нагрузки, в лёссовых грунтах разрушение наступает не от увеличения нагрузки, а от достижения влажности грунта критического значения [1].

На наш взгляд, такое поведение грунта объясняется тем, что по достижении критической влажности грунт переходит в текучее состояние и его несущая способность резко уменьшается от существенного уменьшения прочностных характеристик грунта - угла внутреннего трения ϕ и удельного сцепления с. А разные значения критических нагрузок при различных схемах лотка можно объяснить тем, что при осесимметричной модели в лотке есть возможность образования бугров выпирания не только слева и справа от штампа, как в плоском лотке, но и спереди и сзади от него, что и убыстряет наступление момента разрушения.

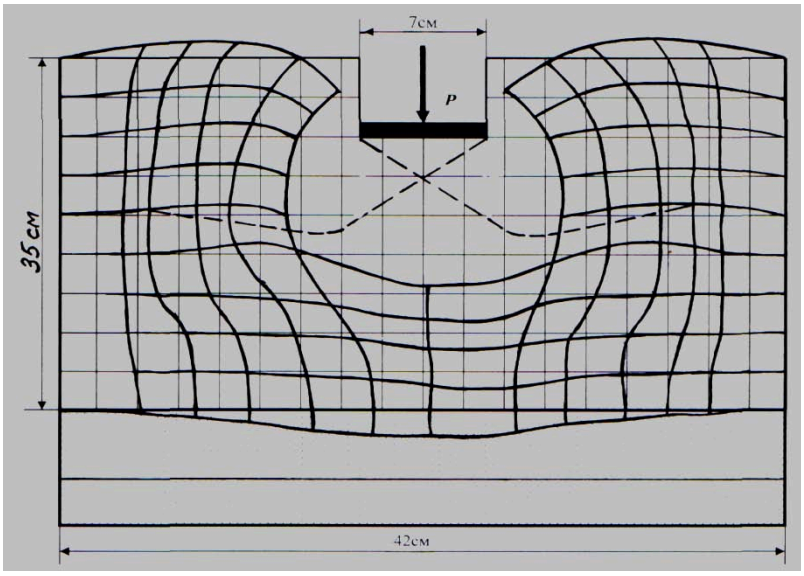


Рисунок 4: Деформация лёссового основания при нагрузке p=0,07МПа и w_{sl}=19% в осесимметричной модели

Необходимо отметить, что как показали эксперименты, при увлажнении лёссового грунта существенно уменьшается величина удельного сцепления, падая практически до нуля, угол внутреннего трения при этом уменьшается незначительно, что объясняется природой лёссовых грунтов, где прочность в основном обеспечивается сцеплением между частицами грунта, а трение между ними играет незначительную роль в связи с их большой пористостью и малой природной плотностью [3].

II. СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Юрченко С.Г. Методика расчета ожидаемой совместной просадочной деформации гидротехнических сооружений и их лёссовых оснований с учетом области замачивания//Вестник Томского государственного архитектурно- строительного универ-ситета // Томск, 2008, №2.
2. Фролов Н.Н. Повышение надежности расчетов гидросооружений на лёссовых основаниях по просадочным деформациям//Мелиорация и водное хозяйство // М., 2007, №2.
3. Будикова А.М., Камалова Н.Н. Research of engineering-geolo-gical characteristics off orest soilsof the city of kyzylorda. Вестник Национальной Академии Наук Республики Казахстан (WoS), 1(2021), ISSN 2518-1467 (Online). Bulletinofnationalacademyof sciences of the republic of kazakhstan ISSN 1991-3494 Volume 1, Number389(2021)
4. Будикова А.М., Отепберген Н.О. Определение характеристик замоченного лессового грунта, научно-технический журнал Проблемы науки, №06(30), 2018г. Свидетельство федеральной службы по надзору в сфере связи, информационных технологий и массовых коммуникаций (роскомнадзор) пи № фс 77-62929. международные стандартные серийные номера: issn print 2413-2101, issn online 2542-078x. http://scienceproblems.ru/images/PDF/2018/30/suprematizm-i-informatika.pdf.