

Autocad Programming For Reinforced Concrete Structural Detailing

Akilu Muhammad¹, Ibrahim Ogiri Hassan²

¹Faculty of Civil Engineering, University Teknologi Malaysia, Skudai, Johor Bahru.

²Department of Building Technology, Federal University of Technology, Minna, Nigeria

Correspondence email; amty72@yahoo.com, +6049126263

Abstract— AutoCAD, the popular software for Architectural and Engineering design is increasingly used today to achieve and speed and accuracy in work execution. Complex and repetitive drawings are automated through AutoCAD programming to optimise the design process. In this research, AutoLISP (AutoCad language), is used to programme AutoCAD for automated reinforced concrete structural detailing. Reinforced concrete Open-well staircase is used as a case study. AutoLISP customization features are used to define the coordinates of any point the software uses to draw any element in the detailing. AutoLISP functions were used in defining function of routines, obtaining user input, defining the drawing pattern and changing the system variables were necessary before using the command function to produce the desired drawing. Accuracy and efficiency is achieved in the use of the developed software and user effort is highly minimized and simplified.

Keywords—AutoCAD ; AutoLISP; Structural Detailing; Open-well staircase; Programming.

Introduction

A good drawing they say speaks thousands of words. Structural Detailing is an essential aspect of structural design work; it provides graphical representation of structural design details for ease of understanding of project requirements between design team and construction team in a construction project. As structural detailing work becomes complex, tedious and repetitive, a logical solution that simplifies the detailing work is automation of the detailing process. This is achieved through customization of AutoCAD drawing task in its built-in development environment called Visual LISP. One of the language of AutoCAD programming is AutoLISP and it is a specialized implementation of LISP programming language used in artificial intelligence. Development of AutoLISP programme for AutoCAD is carried out in a text editor, Visual LISP, it is then loaded and run in AutoCAD¹. Interactive programs are developed that interact with user to obtain necessary information needed to achieve a drawing task. This enhances speed and accuracy in design and detailing, design task that takes hours to accomplish can be achieved in a very small fraction of the time. This paper presents a research work carried out on automation of AutoCAD for reinforced concrete structural detailing using AutoLISP programming language. A case study on a programme for interactive open-well staircase drawing is presented. AutoCAD is customized to instantly produce reinforced concrete structural detailing on receiving instructions relating geometry and other parameters. Autolisp in Autocad

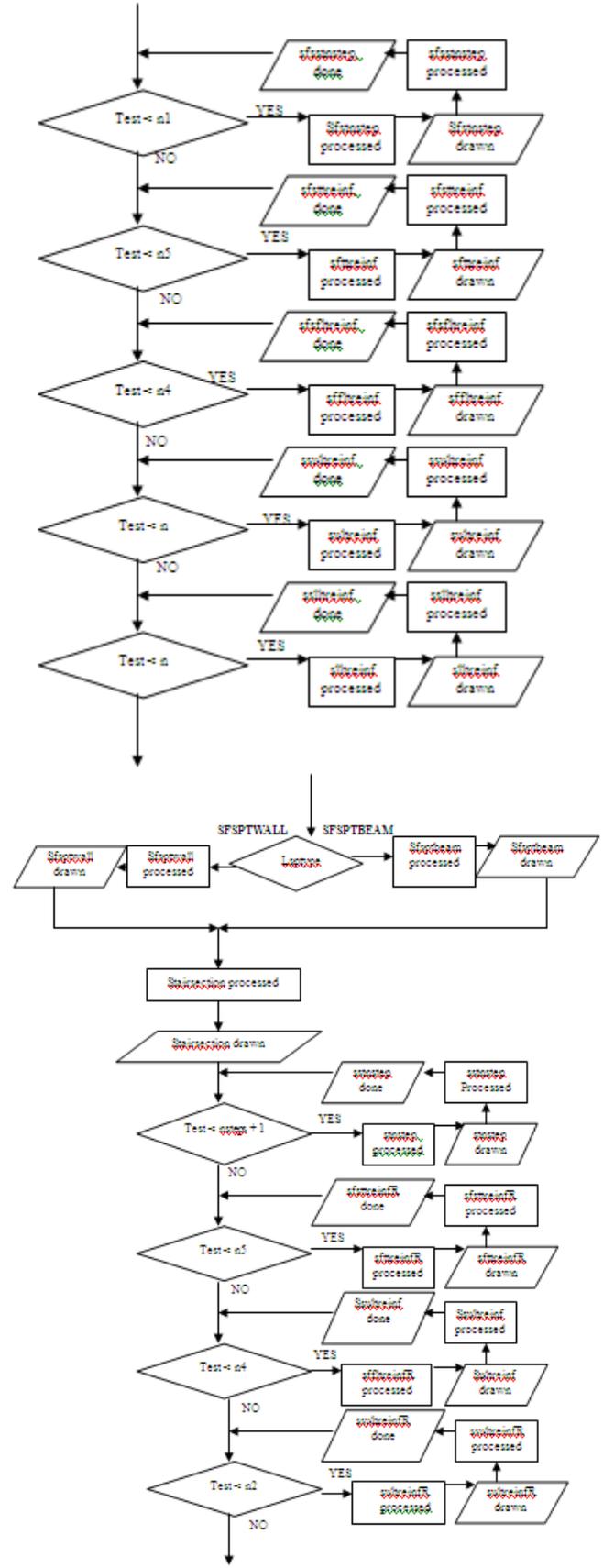
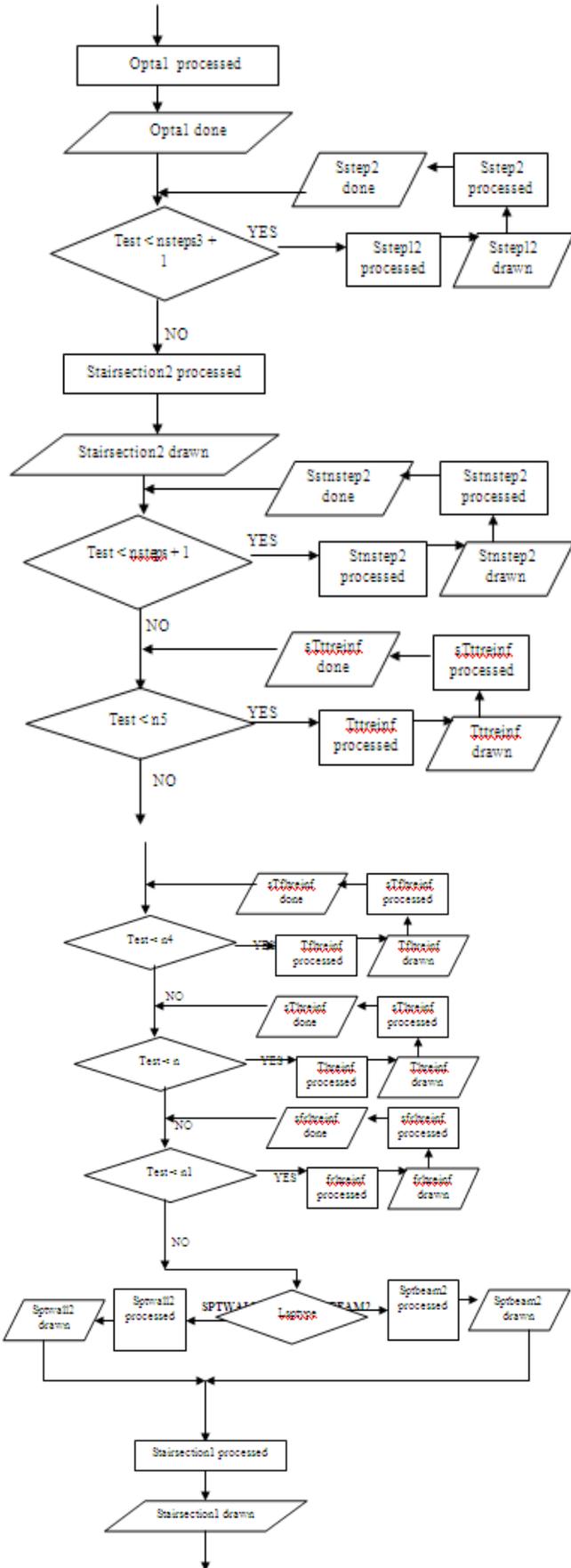
AutoCAD is acknowledged to be the most powerful and widely used technical drawing software that was designed with an open architecture so that you can customize and extend its many features². AutoCAD introduced AutoLISP as an application programming Interface (API) in release 2.1 in the mid-1980s. LISP was chosen as the initial AutoCAD API because it was uniquely suited for the unstructured design process of AutoCAD projects, which involved repeatedly trying different solutions to design problems³. AutoLISP has set the standard for customizing and extending AutoCAD functionality to the extent that one can adapt AutoCAD to ones need by automating repetitive tasks, creating new AutoCAD commands, redefining existing commands, creating new programs or use third party shareware programmes⁴.

III Automation of structural detailing case study of open well staircase AutoLISP customization features were used to develop a programme for automated detailing of open-well staircase.

The steps in program design involve (1) Establishment of the name of the program or function, (2) Obtaining drawing information form user. (3) Processing user information, (4) Production of the desired drawing.

The name of the program or function is first established. This is used in running the program or to instruct the program to carry out specific task. User is then prompted for input relating to geometrical and other parameters which are processed in line with instructions in the source code to produce the drawing.

The Flowchart on figure 1.0 shows that the program will start by seeking user input regarding name, geometry and other parameters of stair to be drawn and starting point of drawing. It will follow instructions in the source code to process and draw stair plan (strplan), then draw first line of plan of step (sstep1) and create of space between the first line of step and the next to be drawn (sstep). A loop command is then issued to make the program perform a test and decide whether the test is less than number of steps plus one. If the answer to the test is yes, sstep1 and sstep will be repeated and another test conducted, the process continues until the number of tests carried out is not less than nsteps plus one. The next (opta) routine offsets the position of pta (the starting point of the drawing) to a point where the first step of the second flight is located. Following the same pattern as above, a similar loop command is given to draw the plan of steps for the second flight (sstep11) and to space the steps (sstep1). The new position of pta is again moved to commencement point of third flight by the routine (opta1) before once again, issuing a similar loop command to draw the plan of steps for third flight (sstep12) and space the steps (sstep2). Stair section for third flight (strsection2) is processed, drawn and dimensioned. Following the same pattern, series of loop commands were issued for the drawing and spacing of section of



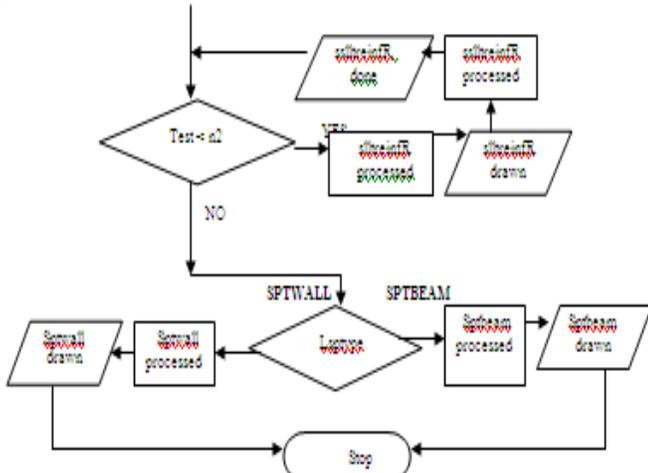


FIG. 1.0: Flowchart for Open-well Staircase Detailing Software

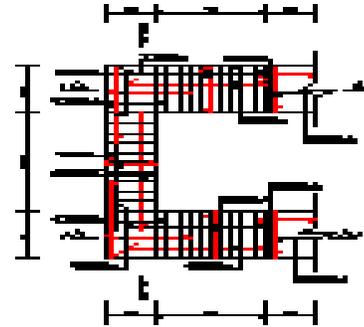
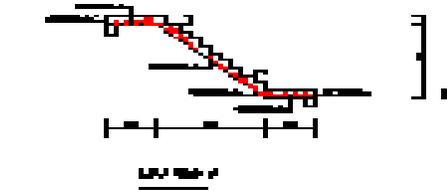
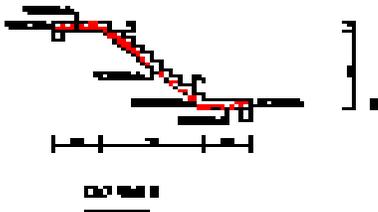
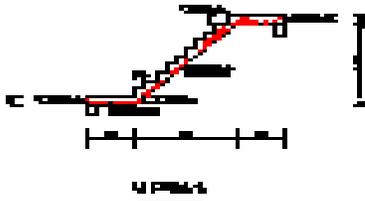


Fig. 2. Open-well Staircase as produced by the programme

REFERENCES

- i. [1] Auto desk (2012) AutoLISP Developer's Guide, AutoCAD 2013, Carlifornia
- ii. [2] E. Fikelstein (2002) *The Bible of AutoCAD. Hungry Minds, New York.*
- iii. [3] AutoDesk, (2002) AutoLISP Developer Documentation,
- iv. AutoCAD 2002 Software, California.
- v. [4] G. Omura, (1990) *The ABC's of AutoLISP.* Syber, Albany.