Implementation of Dynamic Display of Machining Process Based on PyQt

Ren-jie ZHOU¹, Zhi-song ZHU¹,*, Jing LIU² and Long-biao ZHU¹

¹School of Mechanical Engineering, Nantong University, Nantong 226019, China
²Suzhou LEXY Co., Ltd., Suzhou 215010, China

*Corresponding author

Keywords: Python, PyQt, Graphic dynamic display, Keel-processing.

Abstract. The machining process of the keels in the light steel villa are introduced in this paper. The principle of dynamic display based on the characteristics of python is put forward to monitor the machining process more conveniently. The host and slave computer established communication based on Panasonic Mewtocol protocol. In communication, the host computer displayed the machining process synchronously on the main interface with Qt Graphics View framework while machine processed. The host computer displayed and updated the processing state of the keels. The experiment shows that the display process not only meets the requirements, avoids machining errors caused by human factors in time, but also serves as the basis for the study of the quality of the keel.

Introduction

As a new type of green environmental protection building system, steel structure house is developing healthily and rapidly. The structure design of light steel villa has five advantages: no support, variable cross section steel column structure, flexible layout, good seismic performance and light roof. Many countries such as the United States, Japan and Europe, are promoting the application and development of prefabricated steel structure of the middle and low residence actively, forming a production chain and a more mature industrial system¹,²,³. In recent years, middle and low residence develops healthily and rapidly at home, especially light steel villa applications are increasingly widespread.

The skeleton of light steel villa is called keel, whose processing equipment is called keel forming machine. Keel forming machine conveys the steel belt for a certain distance. Then the mold punches while machine stops advancing. Finally, the keel forming machine cuts the steel belt while finishing the keel processing. The keel forming machine comprises a host computer, a slave computer and mechanical part. The three parts cooperate with each other to process the keels. The host computer realizes the function of the work piece selection, history production query and parameters setting of the slave machine. Data transfer through the establishment of communication. The slave computer controls mechanical operation with plc. Mechanical part consists of 8 sets of molds, machine body, feeding device and hydraulic system.

A dynamic display of machining process on host computer needs to be achieved in order to understand the machining process more specifically, quickly and clearly, which is conducive to the control of the processing site and is also advantageous for monitoring whether the slave computer is running normally.

Brief Introduction of Software

The host computer uses software combination of Anaconda2+Eric6+PyQt4 which programs with python language. Python is an interpreted, object-oriented (OOP) scripting language which has dynamic semantics and graceful grammar. Python supports the mainstream existing operating system and can obtain binary code and source files for free. Python is selected as the standard system scripting language by the international free software project plan KDE. Python’s structured or procedural programming can make the program organized into logical blocks to repeat or reuse. PyQt
is a multi-platform to create a GUI application toolkit and can easily develop a high-quality GUI interface\[^{[4,5]}\]. The program mainly uses the QtCore and the QtGui module. The QtCore module contains the core non-GUI functions and the QtGui module contains graphical components and related classes.

The Qt Graphics View framework can be used to display graphics in the program. Qt Graphics View is an item based on MV framework where users can add various models and create different views to display. Qt Graphics View contains three main modules: scene, view and item. Scene is used as a container to manage a large number of primitive objects, managing elements without displaying them. View provides a visual window view of a scene and is responsible for the scene in the graphic display to the user. A scene can have multiple views. Item is the base class in the scene primitives\[^{[5,8]}\], for example: Q Graphics Polygon Item provides a polygon item.

**Dynamic Display Principle**

The dynamic display is to show the machining process of each keel from the beginning to the end and display it clearly and synchronously in the main interface window of the host computer for process monitoring. The main interface of the host computer software is shown in Fig. 3-1. The final effect to be achieved: the main interface displays the unprocessed keels and molds before processing, displays the machining process of every keel synchronously while processing and displays the finished keels and molds after processing. This process is shown in Fig. 3-2 flow chart. Select one keel as an example and elaborate its machining process as follows.

- **Graphic display:** The host computer will need the contour, name and relative position of each unprocessed keel. Display the keels in the main interface window by using Qt Graphics View framework. If there are multiple sets of keels, they are grouped to present. At the same time, the host computer displays the mold outlines and the relative positions in the main interface window according to the relative positions of the molds on the machine.

- **Online communication:** The host and slave computers create communication between python and Panasonic plc with Panasonic Mewtocol protocol to transmit data. The communication process continues from online to offline, ensuring the synchronization and accuracy of dynamic display.

- **Processing:** The main interface window of the host computer sets the outline of the keel in a color switching, flashing state to indicate the keel is processing. In the process of communication, plc drives the steel belt for a certain distance according to the data sent by the host computer and uses the encoder to control forward distance. The host computer reads the value of the encoder to set the strip length in the second window to drive the steel belt to move forward synchronously. When the accumulated distance on the encoder is equal to the data sent by the host computer, the steel belt stops advancing. The length of the steel belt in the second window is the same synchronously. When the molds are drilling, the host computer recognizes limit switches of molds to change the flag bits. The second window sets the molds in different locations with Qt Graphics View framework according to different flag bits, implementing round-trip drilling process.

**Implementation Steps**

4.2, 4.3 and 4.4 realize the graphical display before, during and after the machining process. 4.1 integrates the information of unprocessed keels before the dynamic display.

**Create Action Sequence**

Use Vertex BD Pro 2016 light steel villa design software to design light steel villa. The design will be completed when the keel details are saved as CSV files. The machining process of the keels is abstracted as a plurality of "forward-punching" movements and each action has the difference in the keel number, the name of the keel, the punching molds and the advancing distance. The host computer reads the number, the name, the coordinate, the mold number and the distance of the keel in the CSV file. The distance in CSV files is only relative to the distance between the holes in keels and the origin of the keels. The distance from the origin to the molds and the last moving forward distance
of the steel belt are needed to consider. The actual distance is the sum of the three distances. Each action takes the format of `[['Number', 'Name', 'Mold-Name', 'Distance', 'Finished or not']]]. The distance in each action is chosen as the first element to draw the final mold punching action sequence. The same information of the keel is broken into a plurality of keel action information which is saved as 'List'. 'List' is the most basic data structure in python, and the most commonly used python data types.

Choose one action as an example: `[['2', 'T1B-25', '6', '9.93', '0']]]`. 

Figure 3-1. Main interface window of the host computer.

Figure 3-2. Machining process flow chart.
'2'stands for number 2, indicating the second keel; 'T1B-25'is the name of this keel; '6' represents the 6th mold; '9.93' is the distance required for punching the hole; '0'means that the keel isn’t processed.

When the slave computer receives this action, the steel belt goes forward and stops after 9.93mm. The 6th mold is punched on the keel called T1B-25. If the user selects multiple CSV files, the 'List' is created and executed in the order of selection.

**Graphical Display Before Machining**

GraphicsView1 and GraphicsView2 are used to display the machined keels and molds respectively. The program reads the start and end coordinates and the keel width in the CSV files and use polygons Q Graphics Polygon Item settings to define rectangular function. Display the keel in a rectangular outline and add the outline to the end of rectangular scene. Place the keel name in the middle of a small rectangular box and add it to the middle of the keel rectangle as a whole. Scale keels to keep them finally located in the middle of the interface window though they vary in size. Collect the data of the keel coordinate when the data is imported. Set the whole display proportion according to the difference of the largest horizontal and vertical coordinate in the keel to adjust the proportion of the display. The display style of the unprocessed keel is shown in Fig. 4-1. The red one indicates the keel is unprocessed. If a green keel exists, it means that the keel is finished. In the actual installation, the keel joint is connected by screws which is not displayed in the figures.

```python
self.pen.setColor (QtGui.QColor(255,0,0)):set the keel red. 'self' in this statement is an example of representative class. 'pen' is an elementary graphics object. It is used to draw lines, curves, outlines of rectangles, polygons or other shapes.QtGui.QColor(0,255,0) is color green in PyQt.
```

1 - keel name  2 - vertical keel  3 - inclined keel  4 - horizontal keel

Figure 4-1. Display of processed products.

GraphicsView2 displays each mold and the relative position between mold and the keel forming machine, mold actions, action sequence and machining process. The first to seventh molds are punching molds and the eighth mold is for cutting off the steel belt. Add mold name in polygon. Finally, the results are shown in figure 4-2. The sixth mold is a double face punching mold, so the two sides of the mould are shown in figure 4-2.

```c
V1= [QPointF (x1-50, -y), QPointF (x1+50, -y), QPointF (x1+50, -y+200), QPointF (x1+10, -y+200), QPointF (x1+10, -y+250), QPointF (x1-10, -y+250), QPointF (x1-10, -y+200), QPointF
```

1 - keel name  2 - vertical keel  3 - inclined keel  4 - horizontal keel

Figure 4-1. Display of processed products.
set the mold shape. x1, y is the first mold with respect to the actual distance from the origin of coordinates. QPointF is used to draw lines;

self.poly=QGraphicsPolygonItem(QPolygonF(V1)): abstract the static shape of a mold into a polygon. QPolygonF is used to draw polygons; poly is custom by the program.

self.poly.setBrush(QGui.QColor(0, 255, 0)): fill the keel with color red. setBrush: set a brush to draw shapes of the keel.

self.txtitem.scale(10, 10): adjust the proportion of scene according to the relevant distance set in the software. txtitem means something we want to scale.

1-the sixth mold: punching mold  2-the eighth mold: the cutting off mold

Figure 4-2. Display of machining molds.

Graphical Display in Machining Progress

A "forward-drilling" action will be broken down as the following steps:
1. steel belt moves forward and advances to the corresponding distance to stop;
2. the mold drills.

The corresponding graphical display of step 1 and 2 is shown in GraphicsView2 while the status of keels is updated in GraphicsView1 simultaneously.

Establish Communications. The basis of synchronous display is the communication between the host and slave computer. The host computer communicates with the slave computer, using the Panasonic Mewtocol protocol to send and receive data. Socket is a communication connection at the end of the handle. The host and slave computer communicate through the local socket handle and end-to-end communication after establishing the socket connection. The client creates the local socket.

The global tcpCliSocket: affirm the global variable named tcpCliSocket;
HOST =addr: get the host name of the server;
PORT =int (port): get the port number and convert the type;
ADDR = (HOST, PORT)
TcpCliSocket=socket (AF_INET, SOCK_STREAM)
tcpCliSocket.connect (ADDR): call the socket method and the remote server to connect through sending and receiving method to communicate with server.

The server has the following functions: create the interception object; call the listen function; call the accept method to wait for client connections to arrive; call the sending and receiving method to transfer data to the client. At the end of data transmission, the client and server call the close method, shutting down of the connection actively [9,10].

Graphical Display in Machining Progress. After the establishment of communication, the host computer sends 'List' to the slave computer. plc controls the steel belt moving for a certain distance according to the action, then the corresponding mold punches. When an action from the 'List' is completed, the host computer identifies the corresponding changes in plc register to confirm whether the action has been performed. Then host computer sends the second action and changes the 'Finished or not' bit from 0 to 1. Each keel is cut off by the eighth mold, thus the last mold of each keel must be the eighth mold. If the action is for the eighth mold, the keel changes color upon finishing processing. The program switches color within a short time to mark the next unprocessed one in a flashing state. The rest of the keel remain to update keel state in GraphicsView1.
**Molds Action.** The action of the mold in GraphicsView 2 is synchronized within the whole machining process. During the machining process of the keel forming machine, the limit switches of each mold will change, reflecting the change of the corresponding register in the plc. The host computer identifies the register value to determine whether the mold is at the original point or the moving point. The program sets the original and the moving points of the molds to be mark points. Set a square or round primitive to represent the hole, which is added to the scene at the end of the mold action. Set these primitives at different positions in the scene to achieve the effect of advancing of the steel belt.

```
    self.poly.setPos (0, 350): set the mold in the corresponding position (0, 350) in GraphicsView 2 according to the mark points of molds.
    self.ig.removeFromGroup (k): When the eighth mold moves, the program removes the finished keels from Group and all the elements in GraphicsView 2. ig is a representative object.
```

The dynamic diagram of figure 2-5 is made after the combination of 4.3.1-4.3.3.

![Dynamic Display Diagram](image)

**Graphical Display after Machining Progress.** All the finished keels in GraphicView 1 are set green at the end of the processing. If there are multiple groups of keels, the latest keel will be displayed at the end of processing. All the finished keels are displayed in the methods similar to the unprocessed keels. The difference is that the data of each keel in 'List' have been changed. The mold returns to the initial position and resumes the initial state at last. At the same time, the molds are displayed in the method shown in 4.2.

**Summary**

This program displays dynamic graphics with python during the machining process in the host computer successfully, unifying the theory and the production reality. Graphics update during the communication establishment based on socket. The visual and clear display ensures the production process in a controlled state. The production factors which directly or indirectly affect the quality of the keel analysis can be diagnosed and monitored. Machining errors due to human manipulation can also be found in time to reduce the economic loss.

**Acknowledgement**

This research was financially supported by Nantong Key Laboratory Project and Experimental Teaching Reform of Nantong University.

**References**


