Parametric System Design of Belt Conveyor Legs Based on SolidWorks

Sulan Han* and Xun Liu

North China University Water Resources and Electric Power, 450045, Zhengzhou, China

*Corresponding author

***Abstract*—The DTII (K) belt conveyor is a general series of products, and the machine frame legs can be classified as type I legs and type II legs. Using Visual Basic 6.0 as the development platform, taking secondary development on the SolidWorks software, the dimension driving method and the programming design method are used to achieve the parametric design of legs. Then, by using SolidWorks API function and traversal feature design method, the automatic assembly of middle frame is completed.**

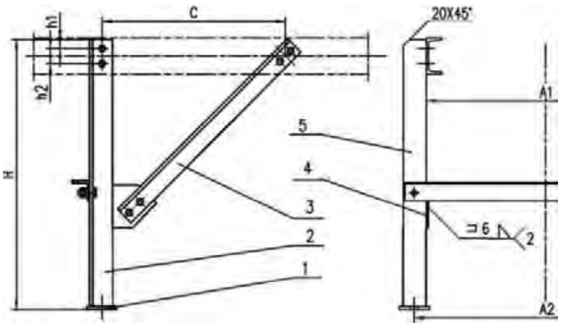
Belt conveyor is a kind of mechanical equipment that transports materials in a continuous manner, and is widely used in metallurgy, port, mining and other industries. The DTII (K) belt conveyor is a general-purpose series of products. The legs are the main support components for the belt conveyor frame, based on different bearing capacity and terrain, the legs are classified as type I and type II, and these types can be divided further into standard height and non-standard height patterns [1,2]. At present, the common design method is manual design on the CAD platform [3], which has a large design workload and non intuitive design process.

With the development of computer three-dimensional (3D) modelling technique, in the design process of mechanical products, 3D model expression gradually takes the place of two-dimensional graphics, and parametric design gradually replaces manual design. Therefore, in order to meet the needs of modern design, a set of parametric design system based on SolidWorks is developed to achieve design parameterization and assembly automation, improve design efficiency.

Keyword—SolidWorks; legs; parametric design; automatic assembly

I. Structural Analysis of Legs

The frame legs are mainly used to support the bearing parts such as middle frame and idler roller, and are commonly composed of five components: base board, supporting legs, diagonal brace, cross brace, and yoke plate. These components are bolted or welded to form a gantry structure, and the supporting legs are generally perpendicular to the middle frame of machine frame. The main difference between type I legs and type II legs is the presence or absence of diagonal brace. Take type II legs as an example, the structure is shown in Figure. 1 [2].



1- Base board; 2- Supporting legs; 3-Diagonal brace; 4- Yoke plate; 5- Cross brace

FIGURE I. STRUCTURE DIAGRAM OF TYPE II LEGS

There are many parameters of legs, including: Band width B, Width A1, Height H, Spacing between diagonal brace and supporting legs C, Installation starting point of bolt hole h1, Installation hole spacing h2 and Foundation bolt center distance A2. Components and parameters of the same type of legs have fixed specifications, and parameters vary with bandwidth B, as shown in Table 1.

TABLE I. LEGS PARAMETERS TABLE

B	H	A1	A2	h1	h2	C
500	765	800	800	25	40	560
800	740	1150	1230	25	40	560
1000	860	1350	1430	30	50	605
1200	850	1600	1680	30	50	605
1600	780	2060	2160	40	70	580
2000	1030	2470	2600	45	80	755

II. Legs System Design

The system uses Visual Basic's graphical objects to establish a human-computer interaction interface, multiple parameters are added by the Items::Add method, achieve dynamic update of data, then creating the DLL file using development toolkit API SDK, to generate the plug-in, to achieve seamless integration with SolidWorks [4].

According to function, the system can be divided into 4 modules: interactive interface module, data management module, parametric design module and automatic assembly module.

The interactive interface module is designed as the system entrance, through which the relevant parameters are select and input. The data management module is mainly used to store the legs parameters and assembly information. The parametric design module is the core module of the system, which can generate legs model that satisfies with the requirements of obtained parameters information. The automatic assembly module normally collects legs based on legs type and pattern and assembles middle frame and idler roller.

III. Key Technologies of Parametric Design of Legs

This system is designed on the basis of the secondary development of SolidWorks. It adopts the dimension driving method. On the basis of analyzing different types and patterns of legs, the corresponding templates are drawn. After obtaining parameters through the system interaction interface, the parameters are assigned to the dimensional variables and the model reconstruction is driven, thereby realizing the overall parameterized design. The process of program design is: creating SldWorks object and connecting with SolidWorks, calling SolidWorks API functions to connect with database, opening template documents, assigning parameters and updating model.

(1) The Sldworks object is the most advanced object in the SolidWorks API, and it provides a direct or indirect interface to all objects in the SolidWorks API, which controls the basic operations of SolidWorks. At the time of programming, the CreatObject method can be used to get the Sldworks object and open a series of template files. Taking the design process of type II legs as an example, the main programming code is as follows:

```
Set swApp = Create Object ("Sldworks.Application")

Set Part = swApp.Open Doc4(strFile Name, 1, 0, "", longstatus)

Set Part = swApp. ActivateDoc ("Type-II-legs")
```

(2) Calling design parameters from the database can make use of the API functions provided by ODBC to connect the system and Access under the VB development environment, so as to realize the operation of the system to the database. In the program, the ADO control mainly used to achieve the link between the program and the database. The main program code is as follows:

```
cn.open"Driver={Microsoft Access Driver (*.mdb)};DBQ="&App.Path& "\Legs.mdb"

rs.Open "Design parameter",cn,adOpen Keyset,adLock Optimistic
```

(3) After acquiring parameters, with the dimension driving method, the Parameter object is used to link the dimension variables in the template and then they are reassigned through the Values attribute of system. However, the parametric design of the legs is designed for the components under the assembly environment. There may be dimension variables with the same name in different components. At this time, if the dimension variables are obtained by name, the operation will be disordered. Therefore, the size name in the program should express the parent-child relationship of the dimension and the part. Taking legs design as an example, the program is as follows:

```
ZTAssembly. Parameter("D4@draft4@SupportingAngleIr_on1").System Value = h11 / 1000

ZTAssembly.Parameter("D5@draft4@SupportingAngleIr_on1").System Value = h12 / 1000
```

The type I legs and the type II legs are similar in structure, and the difference is whether there is a diagonal brace. Thus, the component type can be converted through compressing or decompressing the brace. The compression and decompression of component is achieved by EditSuppress2 function and EditUnsuppress2 function, respectively. Taking compression brace as an example, the main code is as follows:

```
Ztpart.Extension.Select By ID2(ZTstr, "BODYFEATURE", 0, 0, 0, False, 0, Nothing, 0)

Part.EditSuppress2()

Part.ClearSelection2(True)
```

Components such as cross brace, diagonal brace, and yoke plate are series of parts that can be parameterized by changing the corresponding configuration. The configuration change is implemented by the CompConfigProperties4 function.

IV. Parametric Design of Non-standard Legs

In the design of legs, in addition to the standard legs that the legs are perpendicular to the middle frame, there are also non-standard legs. Compared with the standard legs, the supporting legs of the non-standard legs are inclined with the middle frame, between which there is a certain angle α generally ranging from -15° to 30°.For non-standard type I legs, the distance from the top of the middle frame mounting hole h1 and the center distance of the middle frame mounting hole h2 are offset with the angle α. For the non-standard type II legs, the diagonal brace length l also changes with the angle α, whose structure diagram is shown in Figure.2. Based on the standard type II legs, the relationship between the diagonal brace length l and the angle α is analyzed [5], and the formula (1) is obtained.

$$l = \frac{(H-H_1)\sin(90^\circ+\alpha)}{\sin(45^\circ-\alpha)} \qquad (1)$$

H-Height; H_1 - Mounting height of yoke plate; L-Diagonal brace length; A-Inclination angle.

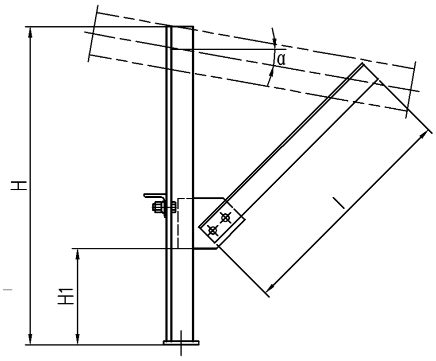


FIGURE II. NON-STANDARD TYPE II LEGS

According to the results of structural analysis, in parametric design, the parameter values are first calculated according to the design conditions, and then modified using the Feature::Parameter object. The main programming are as follows:

Lxie1=(H-Hlianban)*Math.Sin(1.5707964+Val(TextBox2.Text)*Pi/180)/Math.Sin(0.7853982-Val(TextBox2.Text)*Pi/180)

ZTAssembly. Parameter("D1@BossStretch1@DiagonalB_race1").SystemValue=Lxie1/1000

V. KEY TECHNOLOGIES OF AUTOMATIC ASSEMBLY DESIGN FOR MIDDLE FRAME

For automatic assembly, currently the common methods are: (1) assembly modeling method based on spatial position change of parts. This method is to determine the position of the part in space according to the constraints existing between the parts. Through calling the 4×4 matrix of the space displacement and rotation provided by SolidWorks, the part is placed in the specified spatial position. (2) feature-based assembly modeling methods. This method is to automatically select the features of the assembly base component and the assembly target component, then call the matching function in SolidWorks to constrain the freedom of the component in the assembly environment, and finally find complete positioning [6]. This module adopts the secondary method of feature-based assembly modeling for assembly, and the assembly process is shown in Figure. 3.

Programming key technique:

(1) Adding a component. When adding a component, first open the part you want to insert to make sure it is in memory, and then insert it into the assembly environment. The opening and inserting of parts are realized by OpenDoc4 and AddComponent4 functions respectively. The main code is as follows:

Set Part=swApp.OpenDoc6("App.Path+"\"+"Leg2.SLDP_RT",1,0,"",longstatus, longwarnings)

boolstatus=swModel.Add Component("App.Path+"\"+"Le_g2.SLDPRT",0,0,0)

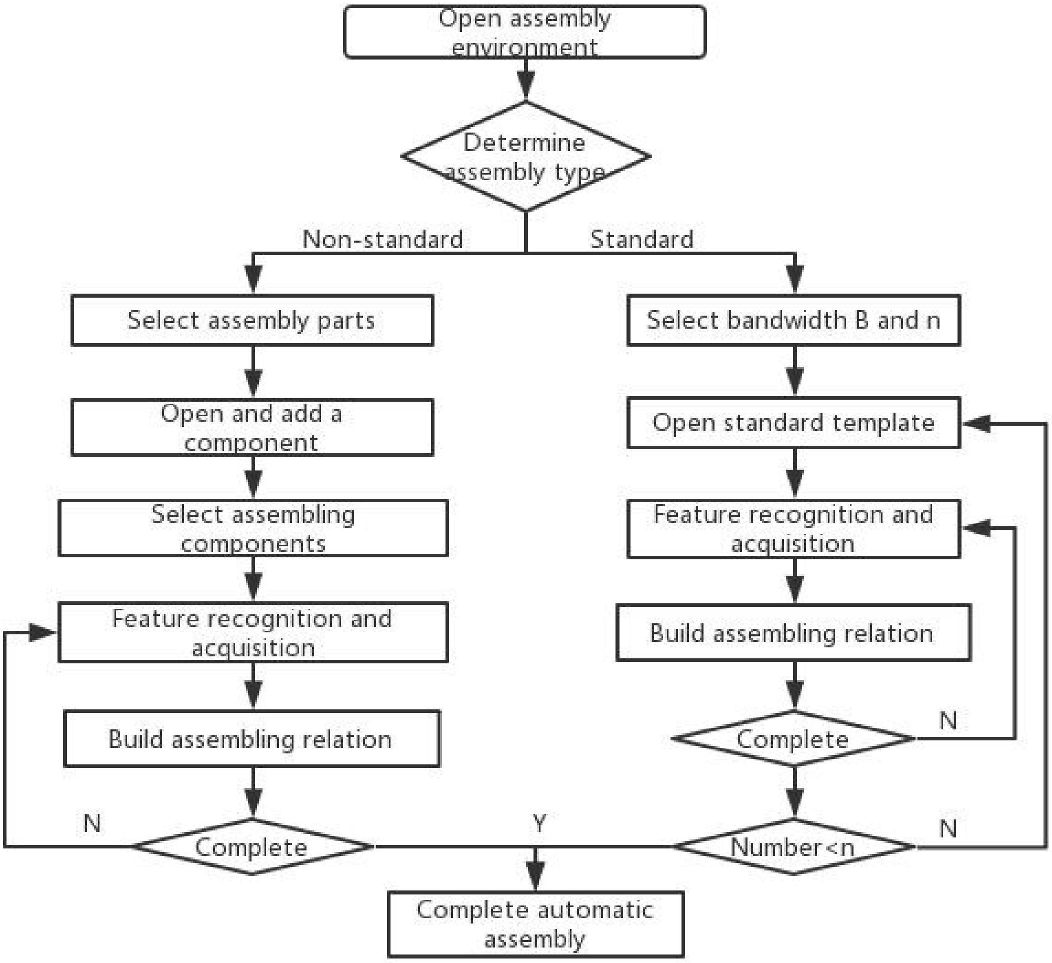

Open assembly environment
Determine assembly type
Non-standard
Standard
Select assembly parts
Open and add a component
Select assembling components
Feature recognition and acquisition
Build assembling relation
N
Complete
Y
Select bandwidth B and n
Open standard template
Feature recognition and acquisition
Build assembling relation
Complete
N
Number<n
N
Complete automatic assembly

FIGURE III. FLOW CHART OF AUTOMATIC ASSEMBLY

(2) Calling in the assembly object. When the module calls in the assembly object, the assembly features are usually identified and acquired according to the feature name. These assembly features mainly refer to the matching reference features created in the template, including the reference plane, the reference axis, etc., which can be implemented by function Select By ID2 in the programming. Since the compressed feature cannot be used as an assembly feature, the feature state needs to be judged before acquiring the assembly feature. The process implemented through traversing feature. The type I legs is obtained as an example, and the main program is as follows:

```
swConf = Model.GetActive Configuration

swRootComp = swConf.GetRoot Component

vChildCompArr = swRootComp.Get Children

For Each Me.vChildComp In vChild CompArr

swChildComp = vChildComp

str1 = swChildComp.Name

Next

Model. Extension. Select ByID2("ZCM" & "@" & CompareStr2 & "@" &AssemStr, "PLANE", 0, 0, 0, True, 1, Nothing, 0)
```

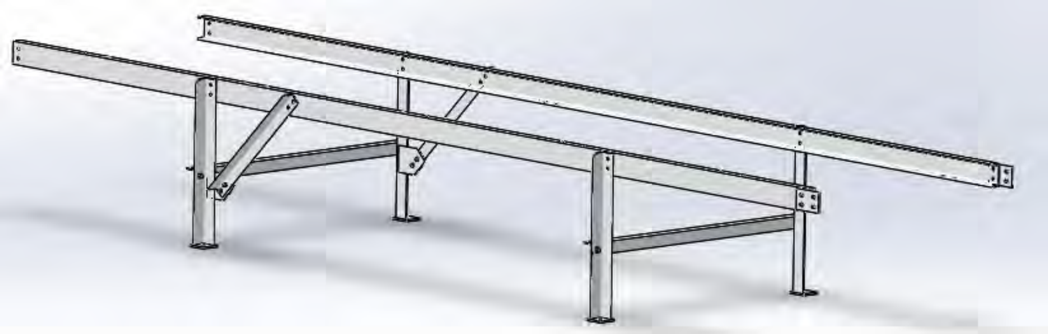


FIGURE IV. ASSEMBLY DIAGRAM OF TYPE II LEGS AND MIDDLE FRAME WITH A BANDWIDTH OF 1000

(3) Defining the coordination. The definition of the coordination is implemented by the Add Mate4 function. Taking the type II legs and middle frame with a bandwidth of 1000 as an example, the main code is as follows:

```
Model. Extension. Select By ID2("QLU" & "@" & CompareStr1 & "@" &AssemStr, "AXIS", 0, 0, 0, True, 1, Nothing, 0)

Model. Extension. Select By ID2("K1" & "@" &swCompStr& "@" &AssemStr, "AXIS", 0, 0, 0, True, 1, Nothing, 0)

JAssembly. Add Mate4(0, 0, False, 0, 0, 0, 0, 0, 0, 0, 0, False, False, longstatus)
```

The resulting assembly is shown in Figure.4.

VI. CONCLUSION

The design system achieves the 3D parametric design of the legs, 3D model of corresponding parts are generated automatically, and the middle frame are assembled automatically, improve the design accuracy, reliability and efficiency, and reduces the cost. It is of great significance to the development of belt conveyor design technology.

REFERENCES

[1] Liu Y, Ma X. Fire detection method of mine belt conveyor based on Artificial Bee Colony algorithm[C]// Chinese Control and Decision Conference. 2017:333-337.

[2] Zhang Shihua, Wei Tie, Belt Conveyor Design Selection Guide of DTII(K) (Volume II) .Jiaozuo Creation on Machinery Manufacturing Co., Ltd. 2013:.48-56.

[3] HAN Sulan. LIU Xun. Parametric System Design of Belt Conveyor Middle Frame Based on SolidWorks, 2017(9):200-203.

[4] Wang Zongyan, Qu Yunqing. SolidWorks mechanical product advanced development technology[M]. Beijing Institute of Technology Press, 2012:70-77.

[5] Gui C, Cui H B, Chen B T, et al. Monitoring system of belt conveyor based on PLC and force control[C]// The, 2016 International Conference on Mechanical Engineering and Control System. 2017:518-527.

[6] Written by SolidWorks Company,Compiler of Shengxinwei Company,SolidWorks API secondary development[M]. Mechanical Industry Press, 2009:125-127.