

3D PARAMETRIC MODELING FOR PRODUCT VARIANTS WITH STUDY CASE ON FLATBED CONVEYOR

Stefanus Ongkodjojo¹ ; Hariyanto Gunawan²

Mechanical Engineering Department, Faculty of Industrial Technology,
Petra Christian University^{1,2}

Product Innovation and Development Centre Petra Christian University^{1,2}

Jl. Siwalankerto 121-131, Surabaya 60236

e-mail: stef.ong@peter.petra.ac.id¹; hariyanto@peter.petra.ac.id²

Abstract

3D solid modeling has been widely used in the last decade as a substitution and complementation of the conventional 2D Drafting. As known, 3D working environment offers many benefits compared to the old system. The question is has it been fully utilised? The technique offered on this paper is how to get a full control of the model by managing and driving parameters attached on it. With the aid of SolidWorks 2004 as 3D Solid Modeling Software with the parametric capability, parameters are built by utilising constraints, relations, shared values, equations, and assembly mates features while driving and managing the parameter values are done with the help of design table and configuration. Furthermore, utilisation of parametric modeling is aimed to support product variants as range of products with a base geometric feature can be developed from a single parametric template model by configuring its parameter values. With the study case of flatbed conveyor, the project has made an achievement in driving the full assembly model by using belt width, overall length, and height as the input parameters. The output taken from the project is a real time update of 3D model, 2D drawings, and Bill of Materials which creates instant product variants in accordance of the input parameters.

Keywords: Parametric Modeling, Product Variants, solid modeling

1. Introduction

The success of a company is significantly depends on the fulfillment the market needs with the product produced. Development time and development cost have become major issues in this case. In focusing on the development time, Smith and Reinersten [1], noted that the best way to shorten the development cycle is to minimize the work or process required to develop the product and eliminate all the unnecessary steps. For a customer driven type of product, the product is designed and developed in focusing to the customer requirements. Even though the product already has a design platform, the design should be customizable according to the quotation of orders. The speed of upgrading the design from its platform to a new design of product variants which met all the customer requirements has become one key point to win the market.

At the design stage, many tools have been applied to simplify and reduce the work process; one method is by utilising Computer Aided Design (CAD) technology. The technology has been developed from the early 2 Dimensional platform to 3D based solid modeling. The basic operations of CAD are input-process-output. Input means all the values and commands inserted by the user through the computer interfaces, while process is interpretation of those values and commands by the computer. Process could be in the form of features or entities of drawing. As an output, we would get visible drawing model and stored data. Based on the nature of computer data system, CAD drawings or models are re-accessible and modifiable. Further developments of CAD system were made to improve the performance itself, called the parametric modeling system where models are able to be modified by driving its parameters.

The utilisation of parametric modeling will be described in this paper which would highlight on the benefit of creating product variants using Solidworks 2004 as the 3D solid modeling software. The selected product for the study case was a flatbed conveyor or more often called as multipurpose conveyor as the project was carried out with incorporation of a conveyor manufacturer. The project was primarily aimed to provide a customer driven product delivery system of multipurpose conveyor through the application of parametric 3D design and modeling. Nevertheless, a comprehensive work on the project was also aimed to gain these several achievements:

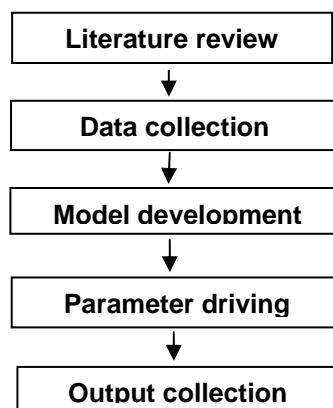


- Reduction of product development time particularly on the design stage
- Production of a user friendly input interface
- Generation of instant model update, technical drawing, and bills of material for product variants

2. Methodology

The project was designed to be conducted according to the following steps;

- Literature review.** The literature review is carried out to obtain a deep source of knowledge regarding the subject being faced and to generate alternatives of solution to the problems. The literature is broadly searched from various sources as: books, journals, e-journals, magazines, and internet
- Data collection.** Data is collected from the company and other sources of the literature in order to be able to generate the model accurately. It mainly consists of geometry, dimensional, functional, and parameters of the product.
- Model development.** Based on the collected data, a 3 D model is built with the selected CAD software and parameters are assigned into the model.
- Parameter driving.** Parameters are managed and driven by an external configuration table on the Microsoft excel base in which the values of control parameters are collected from the user input interface.
- Output collection.** Real time update of 3D model, technical drawing, and bill of material are the final output would be obtained from this project.



3. Literature Review

3.1 Parametric modeling Terminology

'Parametric Modeling' also refers to the same meaning as parametric variational, constraint-based or feature based [2]; where it has the basic knowledge as modeling using parameters and constraints to create relation of geometry and topology on the model. Anderl [3] said that parametric capability of a model is the ability of the geometry of objects to be controlled by a set of parameters. In a parametric modeling, the geometry and the dimensions have a bidirectional relationship where changing in the dimension will give effect on the geometry as well [4].

3.2 Parameter and Constraint Type

The parametric model is controlled by parameters and constraints; those can be categorized as below:

- **Numerical parameter.** It is in the shape of dimensioning variables and scalar values. Distance from point to point, diameter of a circle, length of a profile, radius of a curve, etc are classified as dimensioning variables [5]. These values are linked to the feature or entities they defined, where changing the value of a dimension will change the geometry of the feature. Scalar values define interaction between features, for example number of feature array, scale value, fillet radius, etc.

- **Geometric parameter.** This could be in the shape of a point, a line, a surface, a curve, etc. For example in making a swipe feature, a line profile (could be straight or curve) is needed to define the path of the swipe flow.
- **Geometric constraint.** It defines the relation of geometries or feature which is visible on the modeling screen. Parallelism, tangency, horizontality, concentricity, etc are the examples
- **Engineering constraint.** It is in the shape of equations, equality or non equality [4]. Equality for example is $D1=2*D2$ and non equality suppose hole size \geq bolt size.

3.3 Advantages of Parametric Modeling

A various significant benefits have been obtained from employing parametric modeling in CAD system, as several authors have been noted [3,5,6,7]. Those advantages can be summarized as following:

- Allow interactive changes to any of the design parameter at any time in the evolution of the design model [5,7]
- Provide a flexible solid model in the early stages of product development
- Easy creation of families of parts that saves time and cost [3,5,7]
- Reduce the database storage requirement of families of parts [6]
- Extended performance to tolerance analysis, simulation and kinematics of a model [6]

3.4 Parametric Modeling Route

The process of creating 3D fully parametric product design can be divided into four main stages as illustrated on figure 1 and each step briefly described as follows:

Input Interface

Input interface is the gate to access the model's parameter. This is where the user inserts the values that will drive the modification on the model. According to the link with the parameters, the method can be divided in two:

- **VBA interface.** Visual Basic for Application (VBA) can be used as a tool to create interface by a direct link with the design table as the parameter driver. With the aid of Visual Basic Editor (VBE), macro can be created by building the programme that sets the instruction to modify some certain cells in the design table which correspond with the model's parameter.
- **SolidWorks API.** Application Programming Interface (API) is an interface to access all of the SolidWorks software functionality [8]. It can be run by various programming language such as Visual Basic (VB), Visual Basic for Applications (VBA), VB.NET, C++, C#, or SolidWorks macro files. The access is unlimited and can govern every functions, feature, and operation inside the software including driving the parameters of the model.

Parameter Driver

Items that categorized as parameter driver are functions of the software that set the rules for creating parametric model. Here all relations between values, parameters, and model's entities are created and arranged. The inputs for these drivers are values or definitions while the output links to the property of the model.

Parametric model

As the centre of modeling activity, a model may consist of features, entities, dimensions, properties, sub assemblies, or parts. Every of those operations which have been performed on the model have a unique and typical address. These addresses are the access to control the model parametrically; by assigning them into the parameter driver they will be able to be driven by changing the value or definition on the input interface stage or in the parameter driver directly. i.e. *dimension name @feature*.

Output

The output from Parametric modeling activities can be classified into three types as illustrated on figure 1. Visual 3D model is obtained automatically while the model is being built. Changes on the parameters will result on the visual geometry as it is updated. 2D drawing together with the Bill of materials (BOM) can be generated from the 3D model which has been created before. These outputs on the drawing sections are as fully parametric as the 3D model, where modifications are made on the model these are also updated. Other outputs are classified as digital data for utilities,

as this can be linked with other application that supports the data, for example for creating prototype on the Rapid Prototyping (RP) machine, link to CNC machine, e-drawing, etc.

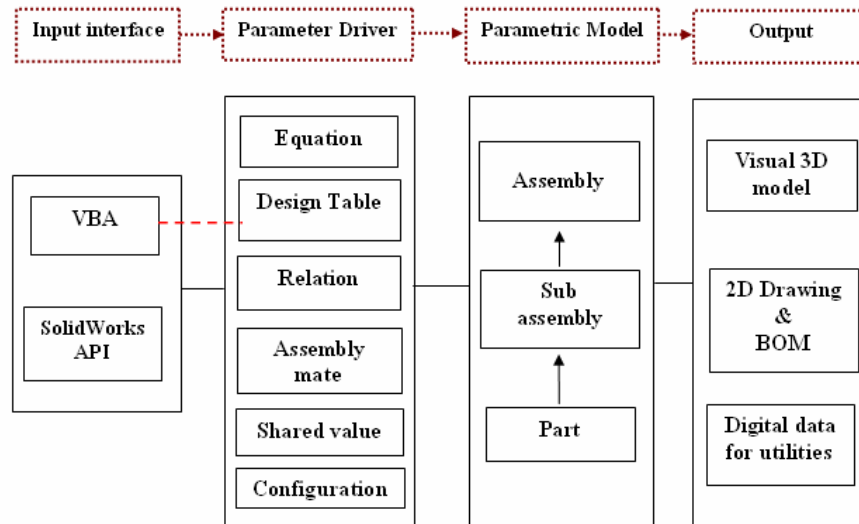


Figure 1. Parametric modeling route

4. Model development

4.1 Building the model

Creating a fully 3D parametric model of flat bed conveyor was started by developing the model itself. Due to the fact that the existing product has not been fully supported with CAD system, most of the components had to be built from scratch. Based on the data collected about the product and the availability of 3D CAD model, the components that build the conveyor can be identified and listed as table I below.

Table I. Component list of Flat bed conveyor

| Component name | Status on assembly | CAD availability |
|-----------------|--------------------|----------------------------------|
| Bed (body) | Part | None |
| End bracket | Part | Available (from company) |
| Belt | Part | None |
| Drive roller | Sub-assembly | None |
| Tracking roller | Sub-assembly | None |
| Leg | Sub-assembly | None |
| Foot adjuster | Sub-assembly | None |
| Fishplate | Part | None |
| Side fishplate | Part | None |
| Bolts & Nuts | Part | Available (SW database download) |

4.2 Building parameters

Parameters have actually been established since the creation of parts began where mostly in the shape of relation as explained on the previous chapter about the type of parameters. But moreover, as the components were assembled together, more parameters and constraints needed to be set up in order to be able to capture the design intent to drive the whole system simultaneously. Building parameters were actualised by these following methods:

Assembly mates

As the components being assembled, assembly mates were used to define the position of each component to each other in the assembly. Conveyor body was used as a position reference for other components by setting the status as fixed while others as float. Assembly mates determine the relative position of each component when a parametric modification and update is brought to the system.

Shared Values

The simplest way to create link of two or more values is using shared values method. Here in the assembly shared values were used to link some dimensions and values between entities as linked dimensions of layout sketch and part features (figure 2), for example dimension of a line sketch with the dimension of sheet metal length on body.

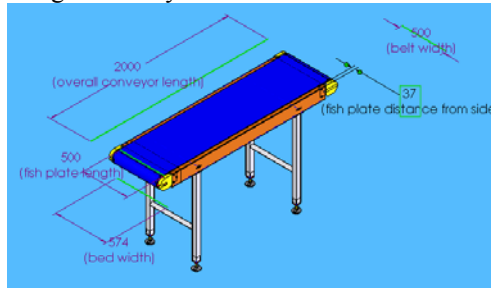


Figure 2. Shared values of layout sketch

Equation

Equation was assigned to parameters value because some simple mathematical solutions were needed in defining the relation. The value of the addressed dimension on the equation was determined on the result of the mathematical evaluation (figure 3)

| Active | Equation | Evaluates To | Add... | |
|-------------------------------------|---|--|--------|---|
| <input checked="" type="checkbox"/> | "D1@Sketch1@box section.Part"="D1@Sketch1@body.Assembly" | '[1] distance of 2 box section = bed width | 574mm | Add... Delete Edit All... Configs... Help OK Cancel |
| <input checked="" type="checkbox"/> | "D1@Sketch1@body.Assembly"="bed width@Sketch2" | '[2] bed width= bedwith on sketch2 | 574mm | |
| <input checked="" type="checkbox"/> | "D4@Sketch1@Tracking Roller Assembly.Assembly"="D1@Sketch1@body.Assembly"/2 | '[3] Tracking roller length = 1/2 bed width | 287mm | |
| <input checked="" type="checkbox"/> | "leg height@Extrude-Thin1@box section.Part"="conveyor height@Sketch1"("D1@Sketch2@bed.Part"+2*"Thickness@Sheet-Metal1@bed.Part")... | '[4] leg height= conveyor height + 2 * thickness | 871mm | |
| <input checked="" type="checkbox"/> | "D2@Sketch1@body.Assembly"="overall conveyor length@Sketch2".270 | '[5] bed length=overall length-2xbracket | 2730mm | |
| <input checked="" type="checkbox"/> | "D11@Sketch1@drive roller assembly.Assembly"="bed width@Sketch2".74/2 | '[6] Roller length = bedwidth on sketch-74 | 250mm | |
| <input checked="" type="checkbox"/> | "D1@Extrude-Thin1@belt stage2.Part"="belt width@Sketch2" | '[7] belt with = belt width on sketch 2 | 500mm | |
| <input checked="" type="checkbox"/> | "distance of leg holes@LPattern1@body.Assembly"="leg distance@LocalPattern1" | '[8] distance of leg flange holes on body = le... | 2130mm | |
| <input checked="" type="checkbox"/> | "number of leg holes@LPattern1@body.Assembly" = "number of legs@LocalPattern1" | '[9] number of holes pattern for leg flange on ... | 2 | |
| <input checked="" type="checkbox"/> | "fish plate distance from side@Sketch2"="bed width@Sketch2"-"fish plate length@Sketch2"/2 | '[10] fish plate distance from side=difference of... | 37mm | |
| <input checked="" type="checkbox"/> | "D1@Distance10"="fish plate distance from side@Sketch2" | '[11] fish plate distance from side on sketch 2 | 37mm | |
| <input checked="" type="checkbox"/> | "D2@Sketch1@fish plate.Part"="fish plate length@Sketch2" | '[12] fish plate length = fishplate length on ske... | 500mm | |
| <input checked="" type="checkbox"/> | "no of fish plate holes in length@LPattern2@fish plate.Part"="fish plate length@Sketch2"/100 | '[13] number of fishplate holes on length direc... | 5 | |
| <input checked="" type="checkbox"/> | "D2@LPattern2@body.Assembly"="no of fish plate holes in length@LPattern2@fish plate.Part" | '[14] number of holes on body=fishplate | 5 | |

Figure 3. Full equation list

4.3 Driving the parameters

All of the parameters that have been built and exist on the model need to be organized in order to be able to be driven simultaneously by one or more values. This was done by assigning a Microsoft excel based design table to the model.

Parameters were inserted into the design table in columns order by double clicking the dimension or value on the model or by copying the exact address to the cell. On the row order configuration was set on default as the changing was meant to be happened on the main model. List of the parameters addressed on design table can be seen on appendix C, while example of design table windows is available on figure 4 below

| | A | B | C | D | E | F | G | H |
|---|---------|-------------------|---------------------------------|-------------------------|--------------------|-----------------------------|-------------------------------|------------|
| 1 | | bed width@Sketch2 | overall conveyor length@Sketch2 | conveyor height@Sketch1 | belt width@Sketch2 | leg distance@LocalLPattern1 | number of legs@LocalLPattern1 | fish plate |
| 2 | Default | 574 | 3000 | 1000 | 500 | 2130 | 2 | |
| 3 | | | | | | | | |
| 4 | | | | | | | | |

Figure 4. Design table windows

Values of the defined parameters can be in number format or text depending on the type of the parameter being addressed. These values were obtained from several sources:

- **Direct input value.** “conveyor height@Sketch1” was filled in with direct value taken from the input cell for conveyor height (figure 5)

| | | | |
|----|------------|----------------|-----------------|
| 11 | | | |
| 12 | Inputs | | |
| 13 | Belt width | Overall length | Conveyor height |
| 14 | 500 | 3000 | 1000 |
| 15 | | | |

Figure 5. Input value on design table

- **Referenced on database value.** The bed width value was based on the requirement of belt width and depending on the availability on the roller length. For that purpose a database of roller length availability was created (figure 6)
- **Result of mathematical function.** Fish plate length was obtained from a mathematical function of the bed width since the steps of fish plate length are in hundreds mm (figure 7)
- **Result of logical function.** Suppression status of fishplate used the logical function in which if the length of the body is less than 2440 mm (sheet metal standard length) no fish plate is required then it is supposed to be suppressed. (figure 8)

| Reference data | | |
|------------------------|----------|-----------------|
| Round up of belt width | RL range | RL availability |
| 500 | 0 | 250 |
| | 251 | 300 |
| | 301 | 350 |
| | 351 | 400 |
| | 401 | 450 |
| | 451 | 500 |
| | 501 | 600 |
| | 601 | 700 |
| | 701 | 800 |
| | 801 | 900 |
| | 901 | 1000 |
| | 1001 | 1100 |
| | 1101 | 1200 |
| | 1201 | 1300 |
| | 1301 | 1400 |
| | 1401 | 1500 |
| | 1501 | Not Available |

Figure 6. Roller length database

| | |
|----------------------------|----------------------------|
| =ROUNDDOWN((B2/100),0)*100 | |
| H | I |
| fish plate length@Sketch2 | number of joints@LocalPart |
| 500 | 1 |

Figure 7. Fish plate mathematical function

| | |
|---|----------------------------|
| =IF((B14-270)>2440,"Resolved","Suppressed") | |
| J | K |
| joint distance@LocalPart | \$STATE@slide fishplate<1> |
| 2440 | Resolved |

Figure 8. Logical function of fish plate suppression status

As all parameters have been linked and driven by design table, modification on the inputs value (belt width, overall length, and conveyor height) will modify all associated parameters according to the rules given and assembly model will be updated automatically when the design table is closed

4.4 User input interface

The aim of the creation of user input interface is to accommodate the modification process of the parametric model by the user. The interface will be the front end (fill-in form) as in figure 9 that controls the model with the bridge of Excel design table, where filled in values on the interface are linked to model's parameter following route as in figure 10. Changes on the interface bring changes to the model at which parameter has already defined in the design table.

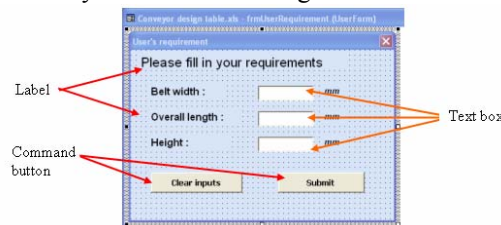


Figure 9. Input interface

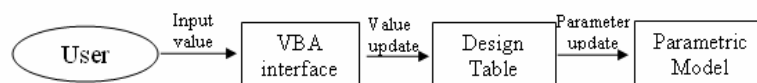


Figure 10. VBA route

5. Output

5.1 Instant update of 3D model

As a result of modification on the parameter's values, the assembled model is updated automatically. This digital data of 3D solid model can be used as a visual representation of the product and other purposes. Few examples of generated conveyor model based on different input requirements;

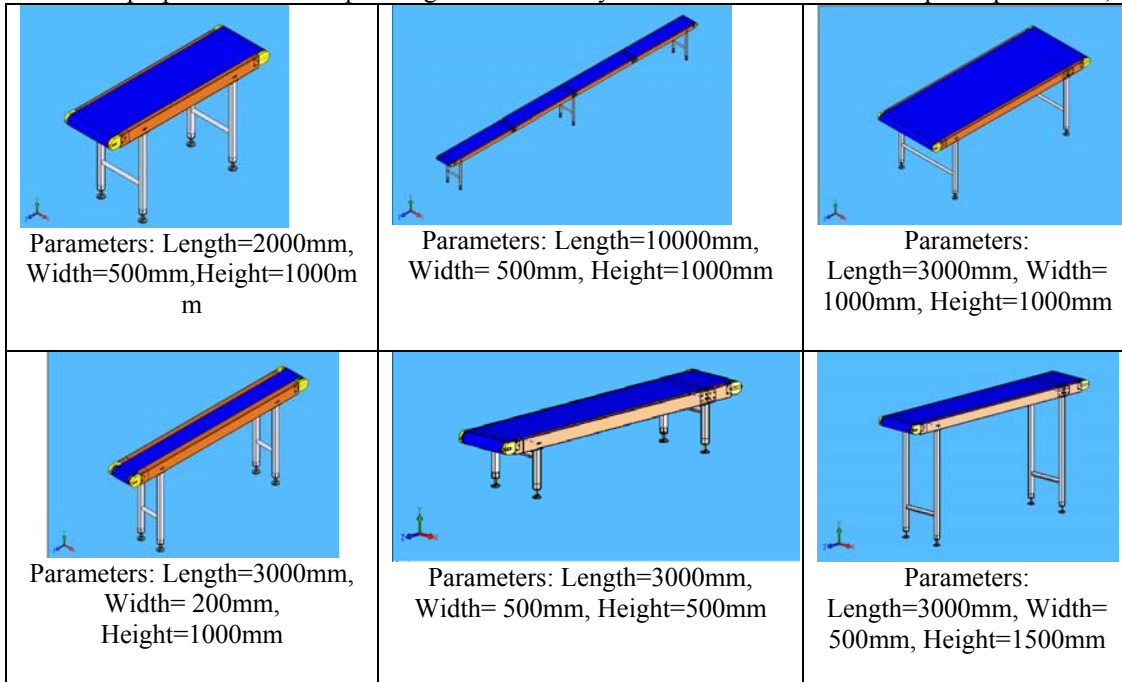


Figure 11. 3D solid model on different parameters setting

5.2 2D Drawing and Bill of Materials (BOM)

2D drawings and BOMs are also fully parametric driven as the model, when an update is brought to the model's parameters the drawings and BOMs are also updated (figure 12). BOM contains information about properties which has been embedded into the model, such as: material, weight, cost, number of parts. Those properties were created through custom properties of the parts or sub-assemblies files. BOMs were created on the drawing with selection on excel base table. The contents were manageable to be modified as requirement. The required properties to be shown on the BOM of conveyor assembly are costs for every component, with refer to below data:

Table II. Component costs data

| Item | Status | Unit Price /cost |
|-----------------|--------------|--------------------------|
| Body | Part | £ 30 / (1220x2440mm) |
| Drive bracket | Part | £ 26 / piece |
| Drive roller | Sub-assembly | £ various |
| Tracking roller | Sub-assembly | £ 10 / piece |
| Leg frame | Sub-assembly | £ 2 / m length 50x50 box |
| Foot adjuster | Sub-assembly | £ 5 / piece |
| Bolts | Part | £ 0.5 / piece |
| Nuts | Part | £ 0.5 /piece |
| Belt | Part | Not available |

For properties which were not able to be directly linked to cost or price, it had to be converted from other properties. For example is to count the body cost against the unit price for 1 piece of 1220x2440 mm sheet metal, the property used to measure was the surface area of the whole body compared to the surface area of 1 piece of sheet metal. This was done by setting the property value linked to surface area which later in the design table the value converted to cost by performing mathematical function.

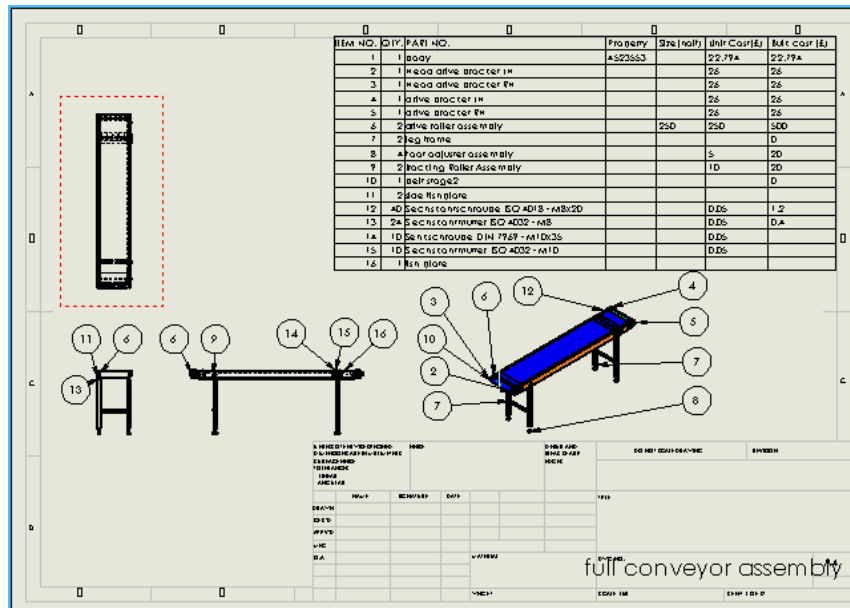


Figure 12. Example of 2D Drawing with BOM

6. Conclusion

With the initial main aim of delivering a customer driven multipurpose conveyor system through the application of parametric 3D design, the project has been carried out and achievements have been made which can be concluded as follows;

- By assigning parameters to a 3D solid model and creating an external configuration table which is linked to the parameters, geometries and entities of the model is able to be driven simultaneously
- The full conveyor model is able to be controlled by by using belt width, overall length, and height as the input parameters
- Product variants can be developed based on a single template model
- Parametric modeling produces output which is also updateable to the modification being brought to the parameters (3D visual model, 2 D technical drawing, and bill of materials)
- An input interface is created to capture the design intent from user and transform it as input parameters for the model.

References

- Tsalidis, S.S and Dentsoras, A.J. (1997) "Application of Design Parameters Space Search for Belt Conveyor Design" *Engineering Application of Artificial Intelligence*. Vol.10 No.6 pp 617-629. UK: Elsevier Science.
- Chua,C.K. and Lye,S.L. (1998) "Parametric Modelling of Drinking Bottles". *Integrated Manufacturing System*, Vol 9/2 pp 99-108.
- Anderl,R. (1994) "Parametrics for Product Modelling". *Parametric and Variational Design*. pp15-26. Ed by J. Hoschek and W. Dankwort. Germany: B.G Teubner Stuttgart
- Anderl,R. and Mendgen,R. (1995) "Modelling with Constraints: Theoretical Foundation and Application". *Computer Aided Design*, Vol. 28. No.3 pp.155-168
- Roche,A. (1994) "STRIM 100: Parametric Modelling for Complex Shapes". *Parametric and Variational Design*.pp53-62. Ed by J. Hoschek and W. Dankwort. Germany: B.G Teubner Stuttgart