Computer-Aided Design and Applications

http://www.cadanda.com

Robust Parameterization Schema for CAx Master Models

Courtney L. Berglund[1] and C. Greg Jensen[2]

[1]Brigham Young University, courtney.berglund@gmail.com
[2]Brigham Young University, cjensen@byu.edu

ABSTRACT

Today's engineering companies rely heavily on an engineer's ability to use computers to analyze and optimize designs. With this use of computers in the design process, products undergo multiple design iterations between preliminary concept and final form, which in turn results in Computer Aided Design (CAD) models being passed from one discipline to the next. An industry wide shift towards the use of CAD master models (CADmm) is taking place in attempts to keep consistency within the design process. With this change to CADmms, manufacturing and engineering development companies are attempting to more fully employ the use of parametrics in their initial CADmm so the initial model handed downstream is robust enough to be used throughout the entire design loop. Unfortunately, current parameter definitions are often not robust enough to incorporate all the design changes from the various analyses and manufacturing operations. To address this problem, we present a more robust parametric methodology that broadens the current definition of parametrics as currently employed on CADmm within CAD packages.

Keywords: CAD, parametrics, master model
DOI: 10.3722/cadaps.2008.715-729

1. INTRODUCTION

Today's engineering companies rely heavily on an engineer's ability to use computers to analyze and optimize designs. From a preliminary concept to final form, products undergo multiple analysis and manufacturing iterations. With this increase in the number of design iterations, Computer Aided Design (CAD) models are constantly being passed from one discipline to the next. In attempts to keep consistency within the design process, current trends are showing a shift towards the use of CAD master models (CADmm). Master models are intended to be used in multiple, if not all, phases of the design process. With this change to CADmm, manufacturing and engineering development companies are attempting to more fully employ the use of parametrics in their initial CADmm so the initial model handed downstream is robust enough to be used throughout the entire design loop.

This CADmm shift is however, often thwarted. Commonly, current parameter definitions are not robust enough to incorporate all the design changes from the various analyses and manufacturing operations. As a work around, the various disciplines will manually edit the CADmm for their respective operations, changing the parameterizations to fit their specific objectives and in the process, losing the previous parametric knowledge stored within the CADmm.

To address this problem, a more robust parametric methodology has been developed to broaden the current definition of parametrics as currently employed on CADmm within CAD packages. Extending the definition of parametrics will yield multiple benefits to the current design process. First, there will be greater flexibility added into the parametric modeling process. Designers will be better able to cope with the various uncertainties inherent with the design process and create more robust CADmm. Also, by including greater flexibility into the CADmm from the onset, less time will need to be spent reparameterizing and updating the CADmm and designers can spend more time doing what they do best, creating more designs and products. Secondly, expanding the definition of parametrics will result in greater amounts of design knowledge to be stored within the CADmm to be used and interpreted in downstream operations. Additionally, the process of incorporating this new methodology within commercial CAD packages will allow users to automatically create driving parameters based on factors of scaling, tolerance and exponential process factors as applicable.

Finally, by expanding parametrics in such a way to include scaling, tolerances and exponential process factors, the CADmm concept can be more completely applied and incorporated within the current design process. More complete application of the CADmm concept will greatly reduce confusion and compatibility issues inherent with passing different models to and from disciplines through different development phases. Also, by having this parameter knowledge incorporated into the models from conception, higher fidelity analysis and research can be conducted on CAD representations of how they will actually perform after the various machining and chemical processes are completed on the product.

The remainder of this paper outlines the implementation and verification of this methodology on various test parts within a CAD package.

2. CURRENT PARAMETRIC METHODOLOGY AND RELATED WORK

The following includes a brief literature review regarding the development and implementation of parametrics as currently employed in CAD systems along with an overview of the CADmm concept. Additionally, some of the inherent weaknesses and issues associated with the parametric method as currently employed in industry and academia will be discussed.

2.1 Parametrics

Parametrics have revolutionized the capabilities of CAD systems. They have opened the door for CAD systems to be programmed and allowed for the reuse of CAD models for multiple purposes [1]. The idea of parametrics can be traced back to the onset of CAD systems in the 1960s with the Sketchpad System that incorporated the use and implementation of graphical modification per geometric constraints [2]. From its beginnings, parametrics has evolved to define a model "…in terms of [its] key dimensions and association between these dimensions" [3]. Research in parametric methods continued to increase with the introduction of feature-based design systems such as Minor and Gossard's system at MIT [4], Dixon's system at the University of Massachusetts [5], the First Cut system from Stanford [6], and the QTC system at Purdue [7]. Features, or any entities within the CAD model belonging to a semantic order higher than the geometric one, opened the door to parametric research, as they necessitated the implementation of a methodology to manage the combinatorial relations associated with the resultant models [8].

The proper application of parametrics has the potential to reduce cycle time, improve end designs and allow for more innovation [9]. The onset and greater availability of parametrics has had a huge impact on industry and the design and production process. When parametrics were applied to the design of a raw materials blending yard the design efficiency was improved more than ten times, shortening design cycle time and therefore costs [10]. While the application of parametrics can often be complicated, proper parametric implementation can greatly accelerate the development process by shortening the design and manufacture cycle times, reducing costs and improving product quality [11].

2.2 CADmm Concept

As previously stated, in attempts to keep consistency within the design process, current industry trends are showing a shift towards the use of CADmm. CADmm are intended to be used in multiple, if not all, phases of the design process, with respective clients including but not limited to, the CAD system and the various domain-specific application subsystems. These subsystems could deal with a myriad of issues such as manufacturing process planning, geometric dimensioning and tolerancing, cost estimation, performance evaluation etc [12]. Of particular importance in this area is the process of removing the segregation commonly employed between the design feature CAD model and the CAD model used for manufacture and machining process plans. By fully implementing the CADmm concept, design and manufacturing models can be one CADmm and not separate representations prone to update and consistency errors [13].

2.3 Limitations with Current Parametric Methodologies

While the use of parametrics has resulted in great success and improvement to the design process, there are inherent limits associated with the parametric techniques and methodologies currently used within commercial CAD packages. One example of limits associated with parametrics as it is used today is the lack of inclusion of tolerance information in solid parametric model representations. For current CAD systems, the addition of value based tolerancing data is only available within CAD generated engineering drawings. This tolerance data is included only in the drawing realm of the CAD system and does not appear in the parameter expressions of the CADmm. This lack of information can result in expensive and time intensive iterations downstream in the development process [14]. To deal with the lack of inclusion of tolerance information, designers and analysts will often change and numerically pad the different parameters of the CAD model to in a sense include the tolerance values into the parametric representation of the model. While this method may work as a quick fix to the problem, inevitably it results in a loss of the previous parametric information and data stored within the CAD model. Along with not including tolerance information, parametric models as currently employed can not easily represent physical and in particular, exponential phenomena such as heat transfer or exponential growth and decay [14].

Another example of limits from the current parametric process is the difficulties involved in representing products in the initial design process. Because current parametric methodologies only allow the storage of single value representations to describe model parameters, CAD systems are not able to fully incorporate and seamlessly update for all of the engineering uncertainties inherent with the initial design process [15, 16]. Quantities such as tolerances, feature scaling values and possible exponential growth or decay factors are all examples of factors that, while not dominant in the early design phase, become crucial in later phases of the design process. This becomes particularly problematic when attempting to scale CAD models for design or manufacturing reasons. Current CAD systems have some built-in scaling operations for scaling CAD parts, however, these operations un-parameterize the part, resulting in dumb solids with little or no parametric knowledge stored in the part. The resulting scaled model has no driving parameters, and the un-parameterized solid cannot be edited through the parameters previously created to define the part. Along with resulting in an un-parameterized part, these methods do not allow for partial scaling, such as a scaling scheme applied only to certain part features or parameters. Because of this, models will often be scaled using workarounds similar to the addition of tolerance information, with designers manually changing the parameter values to manually scale models. This method not only results in a loss of data, but is very time intensive and prone to user error.

Various methods for representing these uncertainties such as using a probabilistic-based approach or fuzzy-set based approach have been researched and explored [17]. While these methods have there distinct advantages and disadvantages, this paper presents a methodology to broaden the definition of parametrics expanding the methodology to support and encourage multiple value representations for model parameters.

3. DEVELOPMENT OF PARAMETRIC SCHEME AND METHODOLOGY

Because of the multiple limitations inherent with current parametric practices, a more robust parametric method is needed to more completely realize the CAD master model (CADmm) concept as previously outlined. This new parametric method was developed and tested to verify its robustness compared to current parametric methods. This was accomplished by first defining the robust parameter equation and identifying the appropriate factors included in it. The parametric scheme was then tested to demonstrate its robustness in handling the various parameter changes inherent with the multiple iterations and phases of the design process compared to current parametric methods.

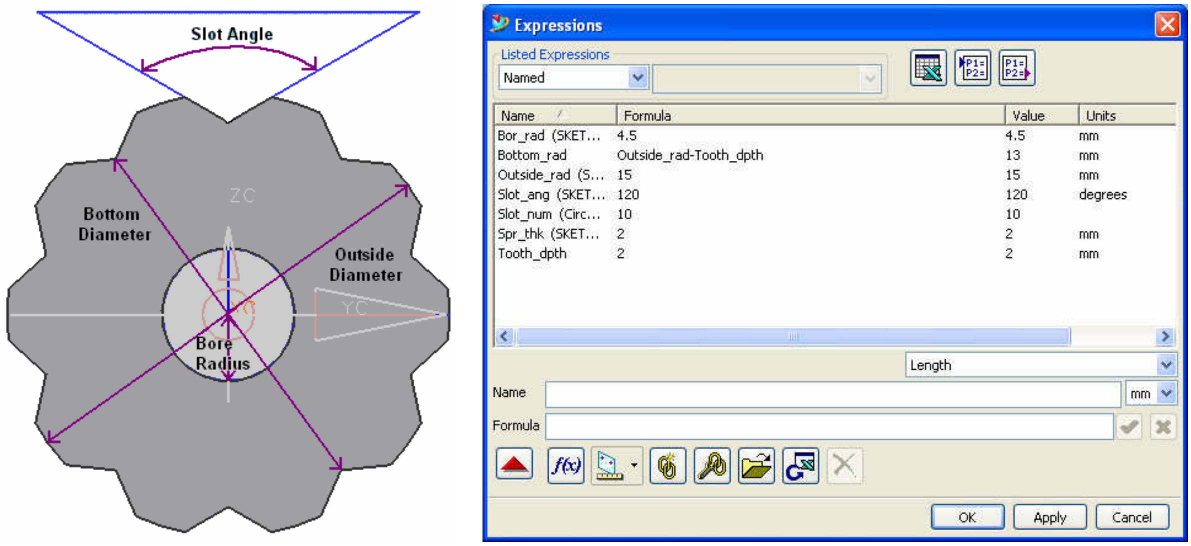
To demonstrate the parameter equation definition, a Robust Parametric Plug-In (RPPI) was developed to be run in conjunction with NX 4.0. Through the use of the RPPI, the robustness of the parameter equation was demonstrated with two different example scenarios, a jet engine turbine case and an automotive internal combustion engine block. The first of these examples has a well documented design history outlining the design and manufacturing steps currently employed by Pratt & Whitney Jet Engine Company (PW). The second example was taken from the current PACE collaborative engineering project at Brigham Young University (BYU PACE). Being such a common part, the engine block also has well documented design and manufacturing processes that were applied in the implementation of the new parametric methodology.

Using this history information, both examples were taken through the various design and manufacturing steps twice, once using a traditional parametric approach and a second time incorporating the new robust parametric method. A comparison between the resulting models per specified metrics, namely, ease of use and time to implement changes was made. The results from this comparison will be covered in more detail in the next section.

The following describe the steps followed in developing and demonstrating the robust parametric schema.

3.1 Robust Parametric Definition Equation

As addressed in the previous section, the current parametric methodology is to use single value representation formulas to represent parameter definition. In the simple example of a sprocket, parameters could be set to define the radius, thickness, radii values etc as outlined in Figure 1.



Name	Formula	Value	Units
Bor_rad (SKET...	4.5	4.5	mm
Bottom_rad	Outside_rad-Tooth_dpth	13	mm
Outside_rad (S...	15	15	mm
Slot_ang (SKET...	120	120	degrees
Slot_num (Circ...	10	10	
Spr_thk (SKET...	2	2	mm
Tooth_dpth	2	2	mm

Fig. 1: Possible parameterization scheme and values for a simple sprocket.

As can be seen from the expression editor screen shot, the various parameters are represented by single values, such as:

$$\text{Hol\_rad} = 4.50\ \text{mm} \tag{3.1}$$

While this single value representation may be adequate for final print parts no longer undergoing parameter revisions, most parts will experience multiple parameter redefinitions. Through these changes, the parametric knowledge surrounding the value change would be lost using the current parametric technique. The following further addresses the development of a more general parametric equation and focuses particularly on the incorporation of the different factors within the robust parametric definition equation.

3.1.1 Scaling

Whether to create an only slightly modified product or a smaller or larger instance of a previous part, scaling of 3D CAD models is an issue many companies deal with. While certain routines exist for uniform and non-uniform scaling of CAD parts within CAD systems, they are not robust enough to incorporate the design and manufacturing needs. Returning to the simple sprocket example, imagine it was to be scaled based on an overall scaling factor of 2.0 for the main body and a factor of 1.2 for the features of the sprocket (ex. hole radius, tooth depth). Current parametric methods would require the part parameters to be updated with new values, either manually or through the import of new values from a spreadsheet or text file. This type of update would result in a loss of the engineering knowledge relating to why the parameter was changed.

Rather than just changing values, the addition of a scaling parameter to the parameter value definition would allow for the flexibility of scaling CAD models without losing specific values concerning the scaling operation. Reorganizing the original user specified parameter to be a part of a driving parameter definition equation to control the original dimension, would accomplish that. Equation 3.2 shows how a parameter definition equation would be organized for a radius dimension. Of importance to note is how the dimension control is transferred from the user specified parameter to a robust equation parameter consisting of the original parameter and a new scaling factor parameter.

$$\mathrm{R\_Radius} = \mathrm{S\_Radius} * \mathrm{Radius} \tag{3.2}$$

Incorporating CADmm scaling by the use of a methodology like this would address the issue of different portions of the model being scaled at different ratios. Because each driving parameter would be reorganized to define the parameter value by means of an equation rather than a single value representation, the resulting scaled models would remain parametric, allowing future changes to be implemented based on the part parameters.

3.1.2 Tolerancing

In addition to modifying models based on scaling parameters, models are often tweaked and changed based on different tolerances associated with the various manufacturing processes completed on the part. In a similar manner to the model scaling update, tolerance values and discrepancies are often entered manually. This again loses the information regarding these changes from the various manufacturing processes. Instead of manually padding the parameter values, if a part radius is specified to be manufactured within a tolerance of 0.01 mm, tolerance values could be added to the parameter definition as outlined in Equation 3.3. As with incorporating the scaling parameter, notice how the dimension control is transferred from the user specified parameter to a robust equation parameter consisting of the original parameter and a tolerance factor parameter.

$$\mathrm{R\_Radius} = \mathrm{Radius} + \mathrm{T\_Radius} \tag{3.3}$$

If the scaled radius parameter in the previous section also had a tolerance value associated with it, the resulting RPDE would be as follows:

$$\mathrm{R\_Radius} = \mathrm{S\_Radius} * \mathrm{Radius} + \mathrm{T\_Radius} \tag{3.4}$$

Of particular importance to note here is the independence of the scaling and tolerance parameters. As a general rule, larger parts have larger tolerance values. However, the relation between the two factors is highly process dependent and often not related based on a consistent factor or amount. For example, doubling the diameter of a part turned on a lathe does not double the process variation. To incorporate this phenomenon into the RPDE, an additional scaling factor can be added to the equation as follows:

$$\mathrm{R\_Radius} = \mathrm{S}_1\mathrm{\_Radius} * \mathrm{Radius} + \mathrm{S}_2\mathrm{\_Radius} * \mathrm{T\_Radius} \tag{3.5}$$

The additional scaling factor allows the designer or manufacturing engineer the freedom of adjusting the CADmm to account for the interactions between the scaling and tolerance values. Alternatively, if the tolerance values do not directly scale with the scaling factor, the designer can independently adjust the tolerance parameter value and set the second scaling factor to the default value of 1.0.

3.1.3 Exponential Processes

Yet another phenomenon affecting dimensions and CAD model parameterizations is that of processes involving exponential growth or decay. There are many situations where the increase or decrease of a variable over a fixed time interval will be proportional to the magnitude of the variable at the beginning of that time interval. Such examples include chemical dipping and etching processes that vary based on the amount of treatment time. These processes react based on various functions, which when included in the previous radius dimension example, xan be denoted by *f(t)* in the resulting RPDE:

$$R\_Radius = (S_1\_Radius * Radius + S_2\_Radius * T\_Radius) * f(t) \tag{3.6}$$

Including exponential processes within the RPDE by means of process functions allows for increased flexibility in the parametric method. This function could represent a myriad of processes, including, but not limited to, power series, logarithmic processes or polynomial operations. For this research, an example process was used to demonstrate the robustness of the new parametric methodology. The exponential process used was a growth and decay process that occurs in nature as well as in many manufacturing processes. This process responds according to the following equation:

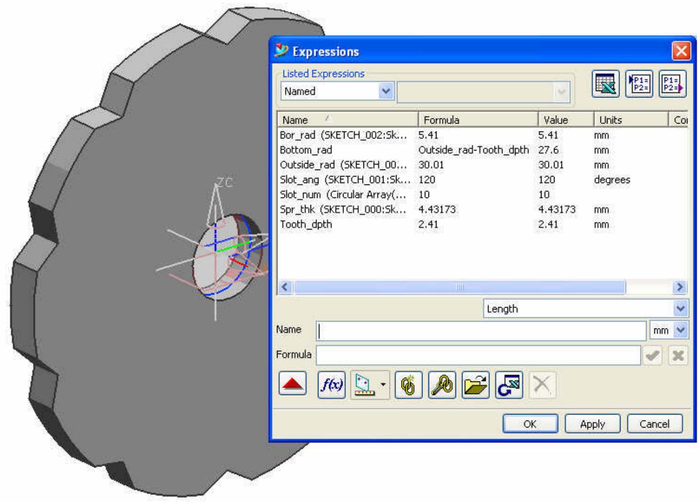
$$N = N_0 * e^{kt} \tag{3.7}$$

Where N is the parameter value after a time t; N_0 is the initial parameter value and k is the exponential constant (growth if positive or decay if negative). Returning once again to the radius dimension example, consider the final part being subjected to a chemical dipping process with exponential growth properties. In this case, the resulting robust parameter definition equation for the radius parameter would be as follows:

$$R\_Radius = (S_1\_Radius * Radius + S_2\_Radius * T\_Radius) * e^{k\_Radius * t\_Radius} \tag{3.8}$$

3.1.4 Fully Integrated Robust Parametric Equation Definition

With the simple sprocket example, after the various processing changes are incorporated manually into the model, the original design is very hard to deduce from only looking at the part parameters (see Figure 2).



Name	Formula	Value	Units	Co
Bor_rad (SKETCH_002:Sk...	5.41	5.41	mm	
Bottom_rad	Outside_rad-Tooth_dpth	27.6	mm	
Outside_rad (SKETCH_00...	30.01	30.01	mm	
Slot_ang (SKETCH_001:Sk...	120	120	degrees	
Slot_num (Circular Array(...	10	10		
Spr_thk (SKETCH_000:Sk...	4.43173	4.43173	mm	
Tooth_dpth	2.41	2.41	mm	

Fig. 2: Manually updated part to include scaling, tolerance and exponential growth factors for sprocket example.

As can be seen even with this simple example, after the various processing parameters are incorporated into the model, the original design is very hard to discover from only looking at the part parameters. Furthermore, while the amount of time required to manually update the part to incorporate scaling, tolerance and exponential growth information is not an overly time intensive process, the amount of time required to implement changes by hand later in the design process for more complex models quickly grows, further reducing the designers and manufacturing engineers time to spend actually designing and manufacturing the respective parts.

As opposed to simply updating the parameters by hand, a more robust parametric scheme could save update time and help preserve the engineering knowledge and decisions stored within the CAD model. The following scheme, when applied correctly, can do just that. The Robust Parametric Master Model Scheme (RPS) consists of a multiple value

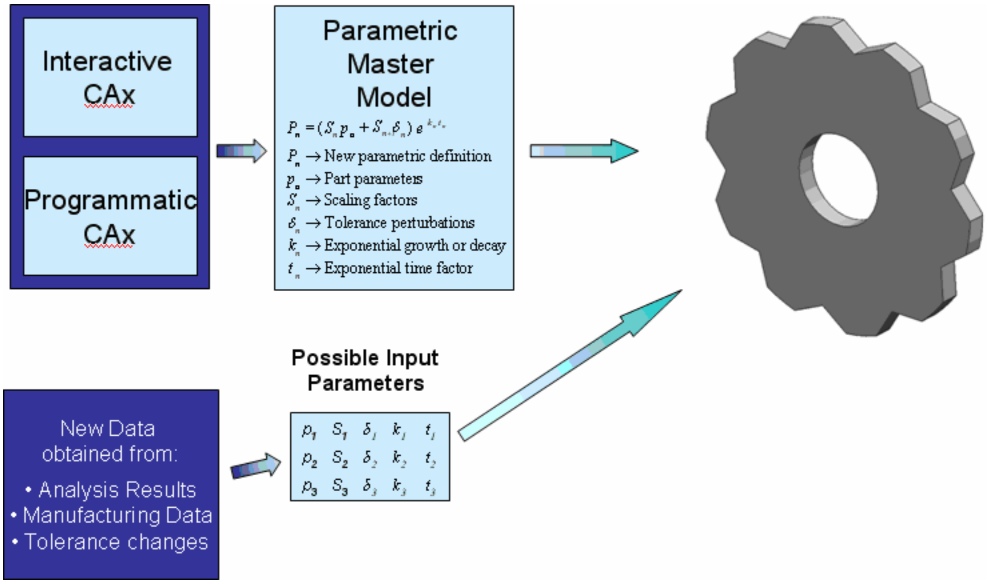


Fig. 3: Graphical representation of the RPMMS outlining the parameter relationships and their flow.

representation equation defining the parametric relationship between the main part driving parameters and process parameter values such as scaling factors, tolerance values and exponential process factors. This equation is defined as:

$$P_n = (S_1 p_0 + S_2 \delta_n) * f(t) \tag{3.9}$$

Where the individual parameters are defined as follows:

- P_n: new robust parametric definition equation parameter
- p_0: part parameter, as entered originally by user
- S_n: scaling parameters associated with the original dimension and tolerance values
- δ_n: tolerance perturbation value
- *f(t)*: specific process to be applied to the part

As previously noted, to demonstrate this methodology, the exponential process used was a natural exponential process as outlined in Equation (3.7). When the general function in Equation (3.10) is replaced with this process, the RPDE is defined as:

$$P_n = (S_1 p_0 + S_2 \delta_n) * e \tag{3.10}$$

Again, where *k* is the exponential growth or decay factor and *t* is the factor associated with the time of the exponential process (see Figure 3).

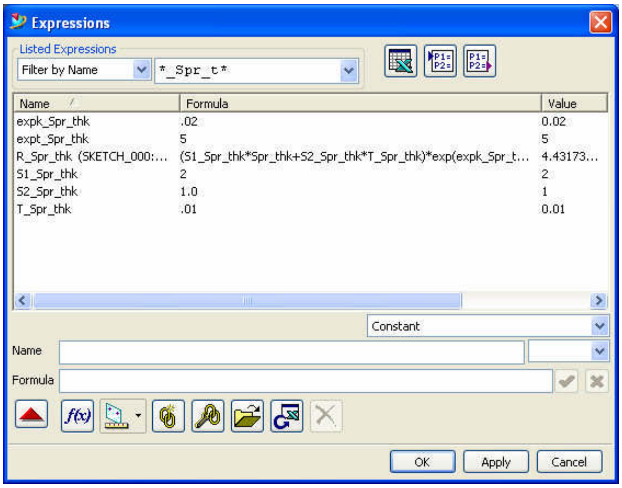
The RPS transitions the traditional single value representation parametric scheme to a multiple value representation system, increasing the flexibility of the parametric CADmm. To apply the RPS methodology to CADmms, a Robust Parametrics Plug-In (RPPI) was developed to be used in conjunction with a specific CAD package, namely NX 4.0. The RPPI allows users to automatically create RPDEs based on factors of scaling, tolerance and exponential growth and decay as applicable starting from a CADmm, with traditional parametrics applied. The RPPI automatically creates scaling, tolerance and exponential growth and decay parameters for all named parameters within the NX 4.0 parametric CADmm. After the initial parameters are created, the RPPI reorganizes the parameters and creates a RPDE, based on the original named parameter and the newly created scaling, tolerance and exponential growth and decay parameters. For explicit details on the development of the RPPI, including code, see [18]. The naming convention for the RPPI is defined as follows:

- Parameter_name: Original parameter name as defined by the user, ie. Spr_thk
- Scaling parameters: S1_Parameter_name and S2_Parameter_name, ie. S1_Spr_thk
- Tolerance perturbation value: T_Parameter_name, ie. T_Spr_thk
- Exponential growth or decay factor: expk_Parameter_name, ie. expk_Spr_thk
- Time associated with exponential process: expt_Parameter_name, ie. expt_Spr_thk
- Robust parameter definition equation: R_Parameter_name, ie. R_Spr_thk

The RPDE allows the user to change the main driving parameter (e.g. Spr_thk) through the expression editor in the traditional manner, but also allows the flexibility and ease of including scaling, tolerance and exponential growth and decay factors without losing the original engineering knowledge concerning the driving parameter. Along with preserving the engineering knowledge for driving parameter values, the RPS also maintains engineering and manufacturing knowledge about the scaling, tolerance and exponential growth and decay factors associated with the various analyses and manufacturing processes preformed on the part throughout the product lifecycle.

In the previous sprocket example, using the RPPI to apply the RPS methodology, the RPDE defining the Spr_thk parameter would be defined as outlined in Equation 3.11 and shown in the expression editor in Figure 4:

$$\mathrm{R\_Spr\_thk} = (\mathrm{S\_Spr\_thk} * \mathrm{Spr\_thk} + \mathrm{T\_Spr\_thk}) * \mathrm{e} \tag{3.11}$$



Name	Formula	Value
expk_Spr_thk	.02	0.02
expt_Spr_thk	5	5
R_Spr_thk (SKETCH_000:...	(S1_Spr_thk*Spr_thk+S2_Spr_thk*T_Spr_thk)*exp(expk_Spr_t...	4.43173...
S1_Spr_thk	2	2
S2_Spr_thk	1.0	1
T_Spr_thk	.01	0.01

Fig. 4: Expression editor screen shot of programmatically updated parameters to portray the RPS, highlighting the R_Spr_Thk RPDE.

4. ROBUST PARAMETRIC SCHEME DEMONSTRATION

To demonstrate the robustness of the RPS, it was compared with current parametric practices through two different examples, an aerospace and an automotive part. In both examples, the RPS methodology was demonstrated by comparing two CADmms of each example, one showcasing the new RPS methodology and the other employing traditional, single value representation parametrics. The two models were then manipulated to incorporate the various design and manufacturing operations consistent with an example product process plan.

In testing the RPS methodology, it was deemed a more robust definition of parametrics if the test CAD models employing the new RPS methodology updated to the various design and manufacturing changes with an equal or higher level of accuracy than those of the original test parts provided by PW and BYU PACE. This accuracy was rated based on two different metrics. The first was a Boolean metric, based on whether the parametric change was straightforward to implement or not. This ease of use metric, was rated by the following dimension:

- -1: No easy way to implement the parametric change, a lot of outside user work and manipulation is required to implement these changes.
- 0: No easy way to implement the parametric change, but little outside user work and manipulation is required to implement these changes.
- 1: Easy way to implement the parametric change, requires no outside user work and manipulation.

The second metric considered the time required to implement the parametric change into the various test case models, represented in seconds. The results of how each model preformed against one other, as well as brief descriptions of each example follow.

4.1 Turbine Case Example

Traditionally, the turbine is a later stage of a jet engine. In the example part this is no exception as the turbine case used follows directly behind the combustor portion of the jet engine (see Figure 5 on the following page).

The turbine is made up of bladed discs that gain energy from the hot gases leaving the combustor. As with all cyclic heat engines, a higher combustion temperature means greater downstream engine efficiency. The limiting factor, however, is the ability of the engine parts to withstand the extra heat and pressure. Because of the nature of the heat encountered in the turbine stage of a jet engine, turbine cases are subject to very high temperatures that require special manufacturing considerations. To deal with these conditions, different materials are chosen and special insulating coatings and machining processes that have been developed and implemented on the turbine case. Table 1 on the following page outlines the ten different processes applied to the two different CADmms in the demonstration of the RPS methodology against current parametric techniques. Also shown in Table 1 are the applicable parametric factors associated with the various processes. As you can see, all of the manufacturing processes incorporate tolerance factors and two of the processes integrate exponential factors for the turbine case.

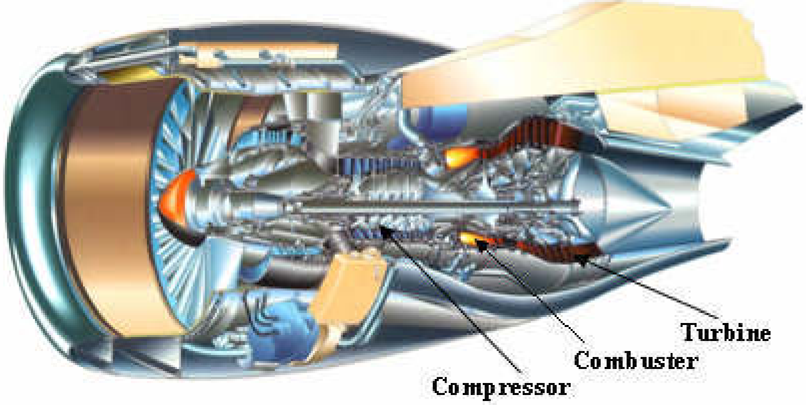


Fig. 5: A cut out PW6000 jet engine featuring the location of the turbine case.

		Applicable Factors		
		Scaling	Tolerances	Exponential
1	Semi-Finish Turn Outer Rails	N	Y	N
2	Semi-Finish Turn Inner Rails	N	Y	N
3	Finish Turn Front Rails	N	Y	N
4	Finish Turn Rear Rails	N	Y	N
5	Coating Process	N	Y	Y
6	Debur Front Flanges	N	Y	N
7	Debur Rear Flanges	N	Y	N
8	Debur Outer Hooks	N	Y	N
9	Debur Inner Hooks	N	Y	N

10	Final Coating Process	N	Y	Y

Tab. 1: Basic machining processes completed in the manufacture of a high pressure turbine case.

Due to the proprietary nature of the turbine case provided by PW, actual values for the various processes will not be explicitly stated here. Rather, the different models will be broadly compared to demonstrate the robustness of the parametric method established in this research. Table 2 shows the results for both methods.

		Ease of Use Comparison			Time to Implement Change		
		Factors	RPS	Control	RPS	Control	Time Diff
1	Semi-Finish Turn Outer Rails	T	1	0	8	42	*-34*
2	Semi-Finish Turn Inner Rails	T	1	0	6	25	*-19*
3	Finish Turn Front Rails	T	1	0	7	34	*-27*
4	Finish Turn Rear Rails	T	1	0	5	26	*-21*
5	Coating Process	T & E	1	-1	26	92	*-66*
6	Debur Front Flanges	T	1	0	9	20	*-11*
7	Debur Rear Flanges	T	1	0	7	18	*-11*
8	Debur Outer Hooks	T	1	0	9	35	*-26*
9	Debur Inner Hooks	T	1	0	8	20	*-12*
10	Final Coating Process	T & E	1	-1	19	108	*-89*
		Total	***10***	***-2***			***-316***

Tab. 2: Results of the ease of use and time to implement metrics for the turbine case example.

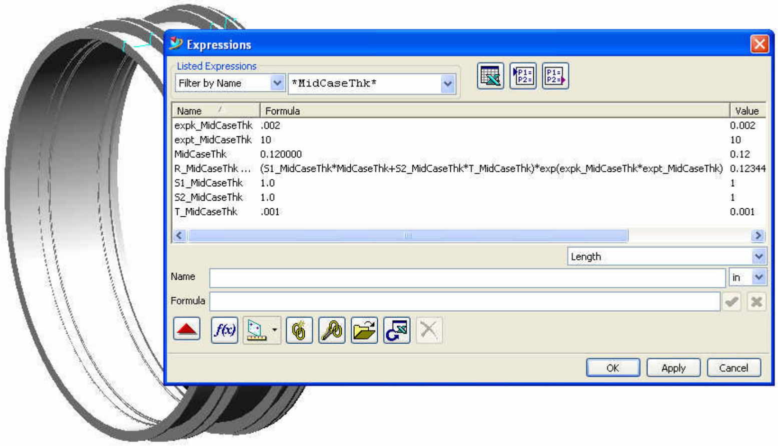
As can be seen, based on the ease of use and time to implement changes metrics, the RPS methodology outscores the traditional parametric method approach. Focusing on the time required to implement the various parametric changes, the results show the RPS methodology requires much less time than the traditional parametric approach. Specifically, when implementing changes for exponential processes, the time saved by using the RPS model is on the order of a five time speed up. As can be imagined, this potential time savings becomes particularly significant when multiple operations and tolerances are applied to different CAD models on a regular basis. These results demonstrate the RPS methodology's robustness and superiority on these levels for the turbine case example.

Perhaps of more importance than the ease of use and time saved with the RPS method is the amount of information stored in the CAD model at the end of the ten different operations. While the control model employs traditional single value representation parametrics, the RPS model has RPDEs consisting of various factor parameters including actual tolerance perturbations experienced from the different processes and specific exponential factors and times of exposure bringing the CADmm to its final state. If example an engineer was interested in how the overall case thickness had changed throughout the different processes. Looking through the scaling, tolerance and exponential parameters of the RPS model would quickly show what factors played a role throughout the different processes applied (see Figure 6).

Just as easily, an engineer could also determine original design intent concerning a specific dimension prior to any tolerances being added or manufacturing processes preformed. This information could be very beneficial in doing backwards analysis as well as determining how dimensions change from the various manufacturing processes preformed on the part and how the resulting changes could affect overall product performance. Contrast this with that of looking at the single value representations of the control model. It would be very difficult, if not impossible to easily deduce these types of information for various parameters using a traditional parameterization scheme.

4.2 Engine Block Example

The second example was based on the engine block used for the BYU PACE international collaborative design project. The engine block is a machined casting consisting of machined cylinders for the pistons of the engine to operate and other machined mating surfaces. The engine block model used was a 4-cylinder engine block, that, for design considerations of the BYU PACE collaborative project, was to be scaled up to a larger 6-cylinder size engine block. This step made the engine block a very interesting example in demonstrating the RPS methodology.



Filter by Name: *MidCaseThk*

Name	Formula	Value
expk_MidCaseThk	.002	0.002
expt_MidCaseThk	10	10
MidCaseThk	0.120000	0.12
R_MidCaseThk ...	(S1_MidCaseThk*MidCaseThk+S2_MidCaseThk*T_MidCaseThk)*exp(expk_MidCaseThk*expt_MidCaseThk)	0.12344
S1_MidCaseThk	1.0	1
S2_MidCaseThk	1.0	1
T_MidCaseThk	.001	0.001

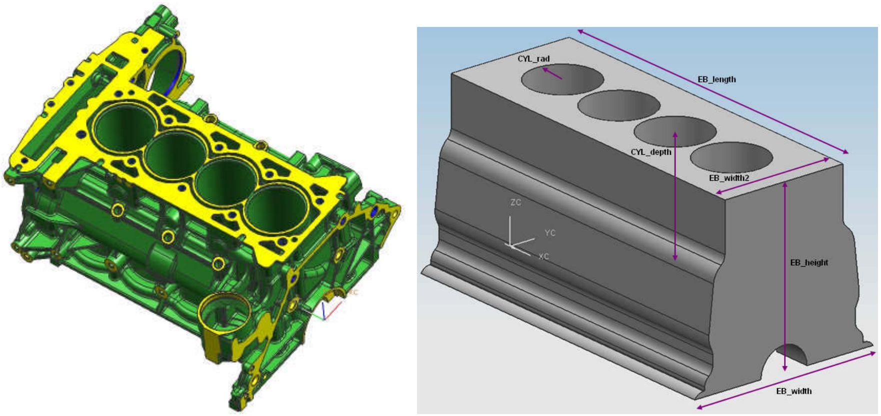
Length

Name: in

Formula:

Fig. 6: Screenshot of expression editor for the RPS model highlighting all robust parameters associated with the MidCaseThk parameter.

Generally, the engine block is a very complicated part with multiple additions and adaptations allowing various parts, such as the crankcase, coolant passages, engine mounts, etc, to be incorporated into the engine design. For the engine block CAD model supplied by the BYU PACE team, this was no exception, with multiple complex surfaces and features all throughout the model. To make the parameterization of the engine block model more straight forward, a simplified CAD model was created incorporating the original design intent. Both models are shown in Figure 7.



(a) (b)

Fig. 7: (a) Actual engine block model as supplied by the BYU PACE team, (b) Simplified engine block model.

In manufacturing an engine block, the first operation is that of casting the main shape. Following the casting, machining and various exponential processes are performed on the part to arrive at the final desired material specifications and product dimensions. Table 3 outlines the ten different processes, along with the associated parametric factors applied to the two different engine block CADmms to demonstrate the validity of the RPS methodology against current parametric techniques. As with the previous case study, tolerances continue to play a major role in the manufacturing of the engine block. More than the previous example, however, exponential processes play a role in the manufacturing process. Also, unlike the first case, the engine block will first be scaled up from a 4-cylinder to a larger 6-cylinder size engine. This additional step will allow for a demonstration of the scaling principle incorporated into the RPS methodology, as applied to a non-uniform scaling operation.

		Applicable Factors		
		Scaling	Tolerances	Exponential
1	Scale engine from 4 to 6 cylinders	Y	N	N
2	Cast part	N/A	N/A	N/A
3	Rough Mill cylinders	N	Y	N
4	Finish Mill cylinders	N	Y	N
5	Carburize cylinders	N	N	Y
6	Rough Bore cylinders	N	Y	N
7	Finish Bore cylinders	N	Y	N
8	Etching process	N	N	Y
9	Anodizing process	N	N	Y
10	Final Coating Process	N	N	Y

Tab. 3: Basic machining processes completed in the manufacture of an engine block.

In applying the different processes to the CAD model, the various processes were researched and general process data was used to approximate the actual machining processes undergone by typical engine blocks. As with the proprietary nature surrounding the PW turbine case, contrived process values, based on industry standards, were used to verify the RPS methodology. The values used were, however, true to function and react in a manner true to the actual processes. Table 4 shows the results for the ease of use and time to implement changes for both methods. Again, as with the turbine case example, based on these metrics, the RPS methodology outscores the traditional parametric method. Before the times required to implement the parameter changes for the different models can be compared however, a more detailed explanation of the scaling process must be defined. As graphically represented in Figure 7 (b), the engine block CADmm had eight different parameters. To maintain simplicity, the scaled up 6 cylinder model would remain of an inline configuration similar to the 4 cylinder model. Also, the actual cylinder radius size would not scale larger, but would remain constant. However, the number of cylinders would change thus increasing the length and the height of the engine block. This scaling operation is very simplified compared to the scaling of an actual engine block. A more rigorous method of scaling would scale the length by a ratio that increased the length only by two times the distance between the centers of two cylinders. This simplification was intentional however as to make it possible to implement by hand. Even with its inherent simplicity, the operation does, however, show the potential of the RPS methodology to more quickly and efficiently scale parametric CAD models with a more sophisticated method and reasoning behind the scaling process.

		Ease of Use Comparison			Time to Implement Change		
		Factors	Metric Value		Time (s)		
			RPS	Control	RPS	Control	Time
1	Scale engine from 4 to 6 cylinders	S	1	-1	27	95	*-68*
2	Cast part	N/A	---	---	---	---	---
3	Rough Mill cylinders	T	1	0	25	41	*-16*
4	Finish Mill cylinders	T	1	0	23	39	*-16*
5	Carburize cylinders	E	1	-1	41	96	*-55*
6	Rough Bore cylinders	T	1	-1	20	27	-7
7	Finish Bore cylinders	T	1	-1	21	30	-9
8	Etching process	E	1	-1	10	28	*-18*
9	Anodizing process	E	1	-1	53	153	*-100*
10	Final Coating Process	E	1	-1	45	123	*-78*
		Total	***9***	***-7***		***Total***	***-367 s***

Tab. 4: Results of the ease of use and time to implement metrics for the engine block example.

Even more so than in the previous example, the results outlined in Table 4 show that the time required to implement the various parametric changes is much less for the RPS methodology than for the traditional parametric approach. Important to note here is how the simplicity involved in the scaling of this case study may downplay the potential time savings of using the RPS methodology for scaling applications. Even so, the over three time speed up is significant when scaling highly complex and detailed parts with hundreds of driving parameters taking weeks to scale accurately.

The engine block example definitely showcases the potential benefit of the RPS methodology in being able to quickly and accurately scale CAD models based on specific scaling factors (see Figure 4 11). Because these times were gathered under the assumption that an experienced designer, with knowledge of the parametric workings of both models was implementing the changes, more favorable time results may have been recorded for the control model. If, as is often the case in industry, an engineer or designer was tasked to scale a CAD model they have little or no experience with, these times could potentially be much larger, further demonstrating the superiority of the RPS methodology when compared to the traditional single value representation approach.

Finally, as with the turbine case example, the amount of engineering and manufacturing knowledge stored within the CAD model at the end of the ten different operations is much greater for the RPS CAmm. Take for example the parameter controlling the length of the engine block (EB_length). For the control model, at the conclusion of all of the operations, the parameter is defined as:

$$\mathrm{EB\_length} = 700.50\ \mathrm{mm} \tag{4.1}$$

This definition gives no background information or insight to the engineering and manufacturing based decisions playing roles in the end parameter value. Compare this with the RPDE of the length parameter in the model incorporating the RPS methodology:

$$\mathrm{R\_EB\_length} = (\mathrm{S_1\_EB\_length{*}EB\_length}+\ \mathrm{S_2\_EB\_length{*}T\_EB\_length}){*}\mathrm{e}^{\mathrm{(expk\_EB\_length)(expt\_EB\_length)}} \tag{4.2}$$

This definition gives much needed insight into how the parameter changed and where it began. It also allows the designer to look at the various scaling, tolerance and exponential factors to determine exactly how the parameter evolved. Very easily a designer could find that the length of the engine block was scaled by a factor of 1.5 and various exponential growth factors with varying tolerances were applied to arrive at the final dimension value of 700.5. Alternatively, the designer could also easily determine what the engine block length was prior to the various processes incorporated into the model by looking at the EB_length parameter. Using the RPS methodology, this parameter would not have been altered from the original value, or 466.7 in this example.

5. CONCLUSIONS

Based on the metrics laid out in this research, namely ease of use and overall time required implementing various parametric changes, it has been shown that the proper application of the Robust Parametric Master Model Scheme (RPS) provides the following:

- More complete incorporation of the CADmm concept within the current design process.
- More effective and versatile incorporation of parametric changes based on scaling operations.
- Tolerancing applied to the CADmm on a feature-by-feature basis.
- Exponential growth or decay processes incorporated into CADmm.

The results discussed in this paper show how the RPS method allows users to easily incorporate parametric changes based on scaling operations. Beyond the built-in, CAD system, un-parameterizing scaling method, traditional parametric methods make scaling CAD models an awkward and time consuming process. Contrast this with the RPS method, which allows CAD models to be easily scaled by changing scaling parameters for the applicable features. Also, the RPS method gives the designers the freedom to scale models accurately based on feature specific scaling factors, and have the resultant scaled part still maintain the parameterization scheme and flexibility it did prior to the scaling process.

In addition to being able to better incorporate scaling abilities to CAD models, this paper also showed how the RPS method allows for tolerancing to be applied to the CAD model on a feature-by-feature basis. As opposed to current CAD system and parametric method abilities, the RPS method allows case specific tolerance values to be applied to different parameters in the design space, and not just as a side note on the engineering drawing.

Finally, this research has shown that the RPS method allows for much more design flexibility when it comes to incorporating exponential growth or decay processes into a design. As it currently sits, there is no out of the box method for applying this data into the CAD model, other than manually calculating and implementing the changes inherent with each respective process by hand. This tedious method is prone to human error which is unacceptable when compared with the ease of including the different process information associated with specific CAD models with the RPS method.

Additionally, by programmatically applying the RPS to current CADmm via the Robust Parametrics Plug-In (RPPI), designers will be better able to plan for and incorporate the various uncertainties and changes inherent with the design process in the creation of CADmm. Proper application of the RPS will also reduce the amount of time spent downstream by designers, analysts and manufacturing engineers reparameterizing and updating the CADmm to fit and apply new product specifications. Also, proper application of the RPS increases the amount of design, analysis and manufacturing knowledge stored within the CADmm. Ideally, the RPS would be implemented within CAD packages at a system level, allowing for proper parameter information storage and update to be handled by the CAD operating system. This CAD system level incorporation of the RPS would allow users to automatically and seamlessly create robust driving parameters based on factors of scaling, tolerance and exponential process factors and change their values as needed. Also, if the RPS methodology is directly incorporated within commercial CAD systems, proper application of the method will be much easier and the time spent updating CADmm and editing driving parameters could be reduced even further.

6. REFERENCES

[1] Lund, J. G.: The storage of parametric data in product lifecycle management systems, Masters Thesis, Brigham Young University, 2006.

[2] Sutherland, I.: Sketchpad, a man-machine graphical communication system, PhD thesis, Massachusetts Institute of Technology, 1963.

[3] Shah, J. J.; Sen, S.; Ghosh, S.: An Intelligent CAD environment for routine mechanical design, Computer in Engineering, 1, 1991, 111-117.

[4] Miner, R. H.: A method for the representation and manipulation of geometric features in a solid model, Masters thesis, Massachusetts Institute of Technology, 1985.

[5] Cunningham, J.; Dixon, J. R.: Design with features: The origin of features, ASME Computers in Engineering Conference, 1988, 237-243.

[6] Cutkosdy, M.; Tenenbaum, J. M.; Muller, D.: Features in process based design, ASME Computers in Engineering Conference, 1988, 557-562.

[7] Turner, G.; Anderson, D. C.: An object oriented approach to interactive, feature based design for quick turnaround manufacturing, ASME Computers in Engineering Conference, 1988.

[8] Monedero, J.: Parametric design: a review and some experiences, Automation in Construction, 9, 2000, 369-377.

[9] Jensen, C. G.; Tucker, S. S.; Jones, C. J.; Rohm, T.: Concurrent engineering with parametric design, Proceedings of the Third World Congress on Intelligent Manufacturing Processes & Systems, Cambridge, MA, 2000, 267-273.

[10] Chen, M.; Huang, D.; Zhang, D.; Zhang, J.: Three-dimensional geometric modeling based on parametric design and its application in enterprise production, Proceedings of the 2004 International Conference on Intelligent Mechatronics and Automation, 4(2), 2004, 733-738.

[11] Anderl, R.; Mendgen, R.: Modeling with constraints: theoretical foundation and application, Computer-Aided Design, 28, 1996, 155-168.

[12] Hoffman, C. M.; Joan-Arinyo, R.: CAD and the product master model, Computer-Aided Design, 30(11), 1998, 905-918.

[13] Zhengming, C.; Ji, G.; Shuming, G.; Qunsheng, P.: An incremental approach to converting design feature model to machining feature model, CAD/Graphics, August, 2001, 22-24.

[14] Ilies, H. T.; Shaprio, V.: An approach to systematic part design, Proceedings of the Fifth IFIP TC5/WG5.2 International Workshop on Geometric Modeling in Computer Aided Design on Product Modeling For Computer Integrated Design and Manufacture, 1997, 17-31.

[15] Nahm, Y. E.; Ishikawa, H.: A new 3D-CAD system for set-based parametric design, The International Journal of Advanced Manufacturing Technology, 29, 2006, 137-150.

[16] Pan, Y.; Geng, W.; Tong, X.; An intelligent multi-blackboard CAD system, Artificial Intelligence in Engineering, 10, 1996, 351-356.

[17] Nahm, Y. E.; Ishikawa, H.: Representing and aggregating engineering quantities with preference structure for set-based concurrent engineering, Concurrent Engineering, 13(2), 2005, 123 – 133.

[18] Berglund, C. L.: Robust Parameterization Schema for CAx Master Models, Masters Thesis, Brigham Young University, 2008.