Journal of Achievements in Materials and Manufacturing Engineering
VOLUME 16
ISSUE 1-2
May-June
2006

Optimizing scale factors of the PolyJet™ rapid prototyping procedure by genetic programming

T. Brajlih [a,*], I. Drstvensek [a], M. Kovacic [b], J. Balic [a]
[a] Faculty of mechanical engineering, University of Maribor, Smetanova ul. 17, SI-2000 Maribor, Slovenia
[b] Store Steel d.o.o. Železarska cesta 3, SI 3220 Store, Slovenia
* Corresponding author: E-mail address: brajlih@yahoo.com

Received 15.11.2005; accepted in revised form 15.04.2006

Analysis and modelling

ABSTRACT

Purpose: The main purpose of our article is to represent results of our research that investigated the implementation of genetic programming methods into optimization process of the scale factor values used in PolyJet™ rapid prototyping procedures.

Design/methodology/approach: The first step in our research was to test the influence of the recommended scale factor values on the dimensional accuracy of the finished parts. Then, the genetic programming was used in optimization of scale factor values regarding to the part's properties. Finally, the optimized values were tested on another test series of parts.

Findings: The optimized scale factor values yield better results in terms of accuracy than values recommended by the manufacturer.

Research limitations/implications: Due to the large increase in part's build time/cost the data range of the Z-axis dimensions of our test series was somewhat narrow, leaving the detailed study of Z-axis scale factor values for further research.

Practical implications: The optimized scale factor values can be used in the RP machine software package in order to achieve higher accuracy of manufactured prototypes.

Originality/value: This paper can be used as a guideline in implementation of genetic programming in optimization process of various manufacturing parameters of RP technologies. Additionally, any user of the PolyJet™ RP machine can use optimized scale factor values described in the paper.

Keywords: Analysis and modelling; Rapid prototyping; Artificial intelligence methods; Quality assessment

1. PolyJet™ rapid prototyping procedure

PolyJet™ rapid prototyping procedure starts with preparation of the CAD three-dimensional model for manufacturing. CAD model (in the STL format) is imported into Object studio™ software package [1]. In the software's graphic environment care is taken of optimal placing of the model on the working tray followed by automatic determination of supports and layering of the model in vertical direction (Z-axis). The layer thickness amounts to 16µm on the EDEN330™ machine that was used in this research [2].

Each individual layer represents a cross-section of the model under manufacturing with added supporting material. The printer applies the model material onto the surface of "model colour" and the support material onto the surface of "support colour". The material is applied to the tray file by the piezoelectric printing head consisting of 1536 nozzles through which 10.000 drops stream per second. The printing head jets onto the metallic base which, after jetting of each layer moves down for the thickness of one layer. Each layer applied polymerizes on its application under the influence of the UV light coming from two UV light bulbs fixed on both ends of the printing head [3].

2. Scale factors

2.1. Implementation of scale factors

The main accuracy problem of the PolyJet technology is shrinking of the building material during the phase of polymerization. Therefore the manufacturer of EDEN330 machines has developed a method of compensating for material shrinkages by implementing a scale factor into the machine's software package [2].

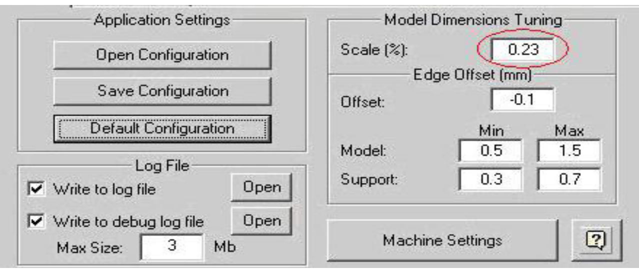


Fig. 1.Uniform scale factor implementation

The CAD models can be scaled (enlarged) uniformly (for the entered scale factor value in all axes equally-Figure 1), or with different scale factor values for individual axes (according to the model's orientation in the machine's workspace-Figure 2). According to the EDEN330 manufacturer, the recommended value of the compensation factor is 0,23%.

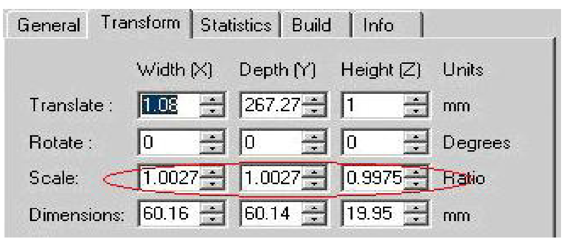


Fig. 2.Individual scale factor implementation

2.2. Effects of scale factors

In order to test the effects of scale factors on the accuracy of the EDEN 330 machine, two series of 12 various objects were produced and measured. When building the first series of objects the scale factors were set at 0 (Figure 3). Therefore the software package did not compensate for the shrinkages. In the second series the recommended value of scale factors was used (0,23%).

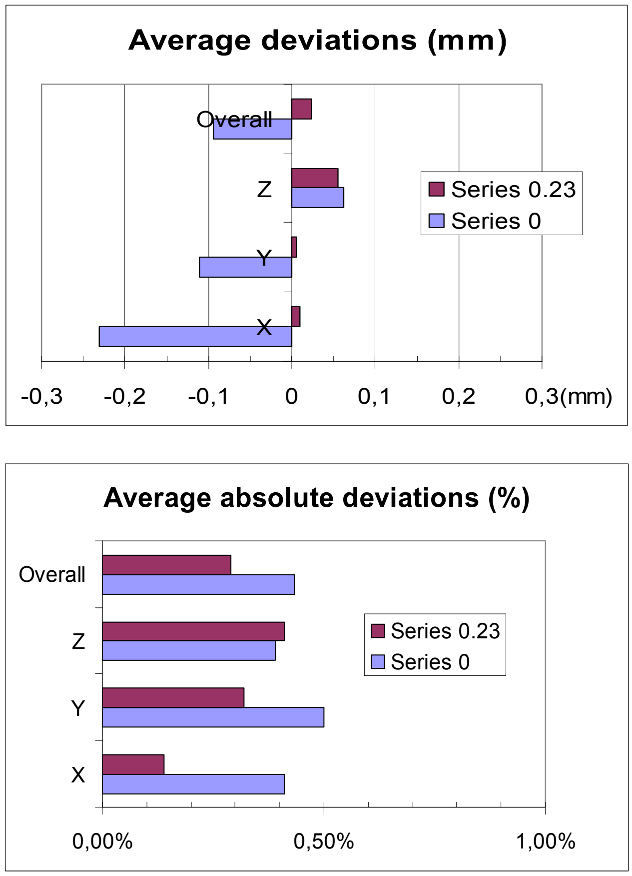


Fig. 3.Average deviations (in mm) and average absolute deviations (in %) for series 0 and 0.23

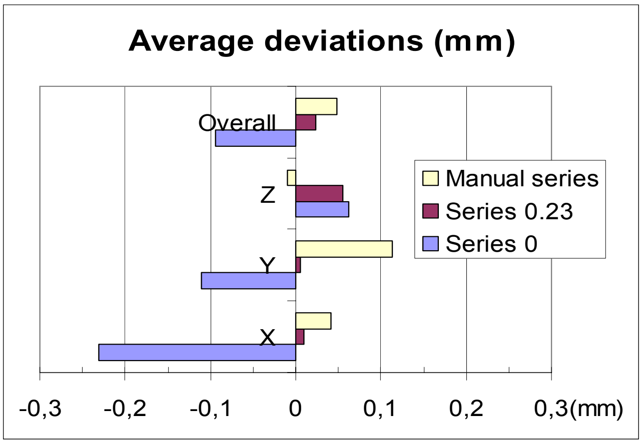
Observing the Figure 6 we can determine the effects of scale factors on the accuracy of the EDEN 330 machine. The 0,23 series produced considerably better results in terms of accuracy than the 0 series. With the recommended value of the compensation factor, the average absolute deviation was reduced from 0,44% to 0,29%. However we were interested if it is possible to optimize the process of scale compensation in order to achieve better overall accuracy than with series 0,23.

2.3. Manual scaling

The effects of model's dimension tuning were further tested with manufacturing of so called "manual series" of objects. Regarding to series 0 results, the CAD models measures were manually changed (according to Table 1) during modeling for an estimated value (in mm) in order to achieve greater accuracy.

Table 1.

	0-50mm	50-100mm	100mm-
X	0,15	0,2	0,25
Y	0,15	0,2	0,25
Z	-0,05	-0,05	-0,05



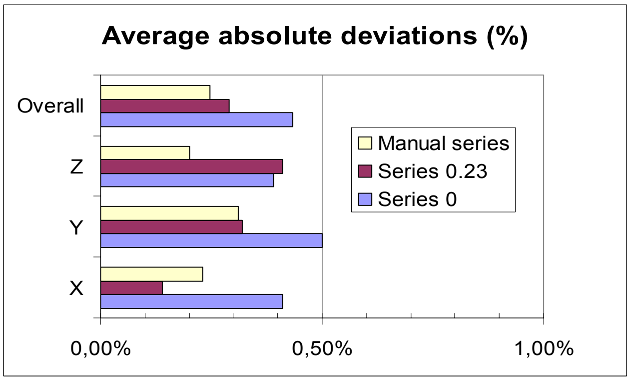


Fig. 4. Average deviations (in mm) and average absolute deviations (in %) for series 0, series 0.23 and manual series

The manual series has proven that it is possible to further improve the scaling process, by adjusting the values of scaling regarding to the nominal measures of different objects and using different values in individual axes. Especially great improvement was achieved in the Z axis (Figure 4).

3. Optimizing scale factor values with genetic programming

3.1. Genetic programming

Genetic programming was used to establish a mathematical relation between nominal measures of the object's CAD model in individual axes, scale factor values and final measures of finished objects. Then this mathematical model was used to determine the optimal scale factor values regarding to the nominal measures of individual objects in each axis. The optimal value of scale factor is calculated in a case of equal nominal and final measure (regarding to the mathematical model).

Genetic programming starts with a primal population of thousands of randomly created computer programs. This population of programs is progressively evolved over a series of generations. The evolutionary search uses the Darwinian principle of natural selection (survival of the fittest) and analogs of various naturally occurring operations, including crossover (sexual recombination), mutation, gene duplication, gene deletion [4,5]. I our case, each of this computer programs will represent a mathematical function, which will more ore less accurately define the final measure of an object (in individual axis) regarding to the nominal measures and the scale factor value used. The final mathematical model will include the most accurate function (the fittest program) for each axis.

3.2. Using the genetic programming

Prior of running the genetic programming five preparatory steps must be completed [6]. In the first step the set of terminals (independent variables and constants) must be defined (Fig 5).

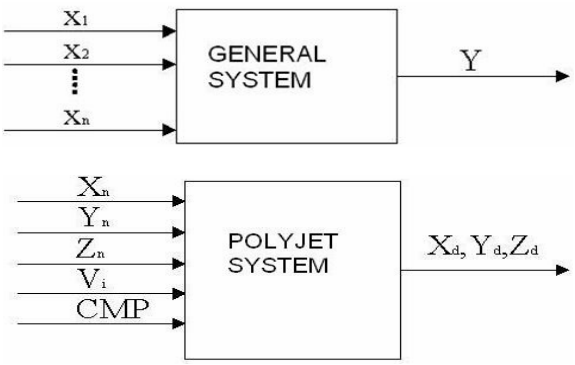


Fig. 5. Independent variables (input) and output of the general and PolyJet™ system

For our problem we defined the nominal measures in Xn, Yn and Zn axis, scale factor's value used (CMP) and so called "volume ratio" (Vi) as independent variables. Vi variable (Fig 6) was used to describe a distribution of material mass in the individual model.

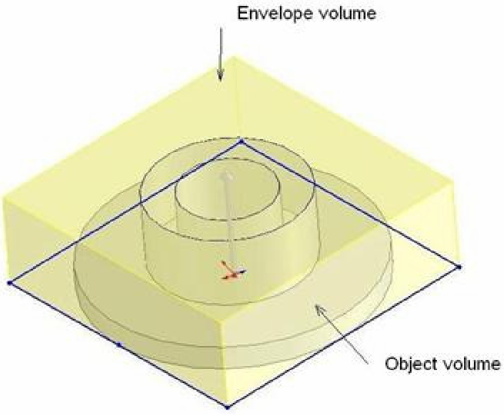


Fig. 6. Vi-ratio (Envelope volume/Object volume)

Secondly the set of primitive functions was defined. We choose the basic arithmetic operation of addition, subtraction, multiplication and dividing. In the third step the fitness measure was defined. In our case, the fitness measure was applied regarding to the difference between the current function's value (F) and the measured final dimension (Xd,Yd,Zd) of the manufactured object. The smaller is this difference, greater genetical potential will be assigned to that function.

In the forth step the control parameters of the genetic programming run are defined. Those include: population size, probability of performing certain genetic operation and the maximum size of individual programs. In the last step the termination criterion is defined. We can define the maximum number of generations or some problem specific terminate condition. The most practical solution is to manually monitor and manually terminate the run when the values of fitness for numerous successive best-of-generation individuals appear to have reached a plateau.

3.3. Results of the first run and simplification of the model

After the completion of the genetic programming run, the mathematical model was created with the search for most accurate functions (with highest genetical potential) in individual axes [7]. The first finding was that in the Vi factor was not present in neither of the most accurate functions (Fig 7). Therefore, it was presumed that the Vi factor does not have a significant impact on the dimensional accuracy of the models.

While trying to rearrange equations in order to calculate the optimal scale factor values, we encountered the "divide by zero" problem (Fig 8). So it was decided, to exclude dividing from the set of primitive functions of genetic programming, in order to avoid this problem.

Additionally the Vi factor was excluded from the set of terminals and finally the Z-axis run was excluded altogether. It was decided that in Z axis the manual series has given the satisfactory results and that the same method of scaling will be used in the Z axis of the final (optimized) test series models. The simplified genetic programming run parameters are represented in Figure 9.

Out[74]=

$$X + \Big(1. \Big(0. + 1.\,(CMP - X)\left(-7.76809 + X + \frac{1.\,(X-Y)}{Y}\right) +$$

$$\frac{1.\left(-15.5362 + CMP - 17.5362\,X - \frac{1.\,(X-Y)}{CMP} + \frac{1.\,X\,Y}{CMP} - \left(0. + X + \frac{1.\,(X-Y)}{X}\right) Y - \frac{1.\,Y^2\left(-7.76809 + X + \frac{1.\,Y}{CMP}\right)}{X-Y}\right)}{CMP\;Y} -$$

$$\frac{1.\,Y}{-7.76809 + X + \frac{1.\,X\left(7.76809 - X - \frac{1.\,(CMP-Y)}{Y} + \frac{1.\,X\,Y}{CMP}\right)}{X-Y}} - \frac{1}{X\,(X-Y)}$$

$$\Big(0.11405\,(X + 0.128732\,(-1. + X)\,X)\,Y \Big(-7.76809 + X + 0.128732\,(X-Y) + \frac{0.\,Y}{-15.5362 + X - Y + \frac{1.\,Y}{X}}\Big)$$

$$\left(-7.76809 + X + \frac{1.\,Y}{\left(0. - X + \frac{1.\,X}{CMP} - \frac{1.\,(X-Y)}{Y}\right)\left(X - \frac{1.\,Y}{X-Y}\right)}\right)\Big)\Big) + 0.128732$$

$$\left(-Y + CMP\,(-8.76809 + 2\,X + 0.128732\,(X-Y))\right)\left(-7.76809 + X + \frac{1.\,X\,(CMP-Y)}{Y}\right)\left(X - \frac{1.\,Y}{-X + 1.\,Y}\right)\Big)\Big) \Big/$$

$$\left(X\left(X + \frac{1.\left(-X - \frac{1.\,(X-Y)}{CMP} + \frac{1.\,Y}{-7.76809 + X} - X\,Y + \frac{1.\,X\,Y}{CMP}\right)}{Y^2}\right)\left(7.76809 + \frac{1.\left(7.76809 + \frac{1.\left(7.76809 - X - \frac{1.\,(X-Y)}{Y} + \frac{1.\,X\,Y}{CMP}\right)}{Y}\right)}{Z}\right)\right)$$

Fig. 7. The most accurate function (X-axis)

```
In[70]:= X = 60
         Y = 60
         Z = 20
         Solve[f == 60, CMP]

Out[70]= 60

Out[71]= 60

Out[72]= 20

         Power::infy : Infinite expression 1/0 encountered. More…

         Power::infy : Infinite expression 1/0 encountered. More…

         Power::infy : Infinite expression 1/0 encountered. More…

         General::stop : Further output of Power::infy will be suppressed during this calculation. More…

         ∞::indet : Indeterminate expression
            0. + 52.2319 (-60 + CMP) + ComplexInfinity + ComplexInfinity + ComplexInfinity encountered. More…
```

Fig. 8.“Divide by zero” problem

3.4. Results of the second run

Again the results were analyzed by searching for the most accurate functions in X and Y axis (Figure 10). Then both of those functions were used to calculate the optimal scale factor values regarding to the nominal measures (Figure 11).

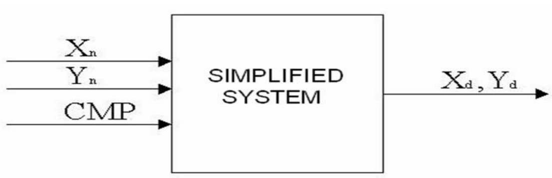


Fig. 9. Parameters of the simplified run

```
Out[78]= CMP + 2. (-0.293558 + CMP) (0.315099 + CMP) (2.03191 + CMP) (0.0886152 - 0.0850712 CMP + CMP^2)
           (4.09409 + 2.86888 CMP + CMP^2) (1.10738 + 2.59087 CMP + 8.31979 CMP^2 - 1. CMP^4 + CMP^3 (-0.193469 + X)) + }

In[79]:= X = 60
         Solve[f == 60, CMP]

Out[79]= 60

Out[80]= {{CMP → -2.03196}, {CMP → -1.43443 - 1.42703 ⅈ}, {CMP → -1.43443 + 1.42703 ⅈ},
          {CMP → -0.319341 - 0.121729 ⅈ}, {CMP → -0.319341 + 0.121729 ⅈ},
          {CMP → 0.0375149 - 0.367728 ⅈ}, {CMP → 0.0375149 + 0.367728 ⅈ}, {CMP → 0.122217 - 0.155143 ⅈ},
          {CMP → 0.122217 + 0.155143 ⅈ}, {CMP → 0.243261}, {CMP → 59.946}}
```

Fig. 10. The most accurate function (second run-X-axis)

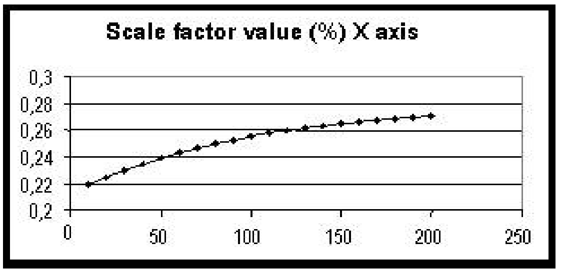
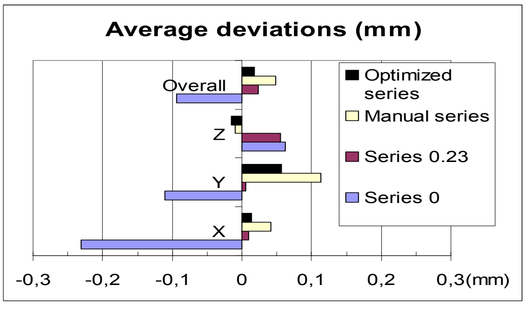


Fig. 11. Optimized scale factor values regarding to the nominal measures (X-axis)

4. Test of optimized scale factor values

In order to test the optimized scale factor values we have produced another series of test objects. This time we have calculated compensation factor value for each test object for each individual axis separately according to the mathematical model. The results of the optimized series show additional improvement in accuracy of the PolyJet rapid prototyping procedure over previous series (Figure 12).



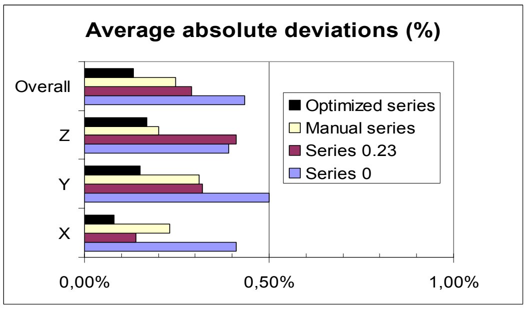


Fig. 12. Results of the optimized series

The average absolute deviation was reduced from 0,44% of the series 0 to 0,13% of the optimized series. Especially large improvement has been achieved with the optimized values of compensation factors in the X-axis of the machine. (0,41% of series 0 to 0,08% of the optimized series).

5. Conclusions

The optimisation of the scaling process has definitely improved the accuracy of the PolyJet procedure. The problem of our method is that we are able to optimise scale value of a model based on only one dimension of a model in a particular axis. Because most "real-life" prototypes have many different dimensions in individual axes, choosing the optimal dimension on which to calculate the scale factor can be difficult. However, for the common usage of rapid prototyping the recommended value of scaling enables satisfactory results. Our optimisation method becomes useful, when we have to manufacture a prototype with one dimension that has very high accuracy demands. In that case, we can calculate the appropriate value of the scale factor for that particular dimension and than scale the whole prototype (correctly orientated in workspace) in appropriate axis by this factor value.

References

[1] I. Drstvensek, Layered Technologies, University of Maribor Faculty of mechanical engineering, Maribor, 2004
[2] Object geometries homepage available at http://www.2objet.com/home.asp
[3] I. Pahole, M. Ficko, I. Drstvensek, J. Balic, Rapid Prototyping processes give new possibilities to numerical copying techniques, J. Mater. Process. Technol. 164/165 (2005) 1416-1422
[4] M. Brezocnik, J. Balic, Z. Kampus, Modeling of forming efficiency using genetic programming, J. Mater. Process. Technol. 109 (2001) 20-29
[5] M. Brezocnik, J. Balic, K.Kuzman, Genetic programming approach to determining of metal material properties, J. Mater. Process. Technol. 13 (2002) 5-17
[6] J.R. Koza, , Genetic programming : on the programming of computers by means of natural selection, The MIT Press, Cambridge, London, 1996
[7] I. Drstvensek,. M. Ficko, I. Pahole, J. Balic, A model of simulation environment for prediction and optimization of production processes, J. Mater. Process. Technol. 155/156 (2004) 1641-1646