Evaluation of alternative dynamic behavior representations for automated model output classification and clustering

Nisa Önsel
Boğaziçi University, Industrial Engineering Department
34342 Bebek Istanbul Turkey
+90 212 359 73 43
nisa.guler@boun.edu.tr

İbrahim Emre Önsel
Istanbul Technical University, Mining Engineering Department
34469 Maslak Istanbul Turkey
+90 212 285 61 40
onsel@itu.edu.tr

Gönenç Yücel
Boğaziçi University, Industrial Engineering Department
34342 Bebek Istanbul Turkey
+90 212 359 73 43
gonenc.yucel@boun.edu.tr

Abstract

Automated behavior mode identification and clustering are potentially valuable additions to the analysis toolset of a system dynamics (SD) modeler. The key component for such tools is the feature vector construction; selecting a set of features to represent the dynamic behaviors to be classified or clustered. In this study, we evaluate a set of alternative feature vectors in clustering basic behavior modes encountered in SD practice. As the first case, coefficients of the polynomials fitted to the dynamic behavior are used as the features. In the second case, a given set of curves are fitted to the dynamic behavior, and the degree of fit to these curves are used as the features. The third case constructs feature vectors based on the changes in the signs of slope and curvature of the behavior. In other words, the feature vector represents the original behavior as a sequence of atomic behavior modes. In our preliminary evaluation, the third approach outperformed the former two. Later, we propose a set of extensions to the third approach in order to improve its performance while dealing with oscillatory behaviors. The modified version of the third approach is evaluated to perform better than the original one in clustering both non-oscillatory and oscillatory dynamic behaviors.

Key words: Pattern recognition, clustering, system dynamics, dynamic behavior pattern

1. Introduction

In simulation modeling a modeler needs to evaluate the output of a model for purposes like parameter estimation, calibration, scenario analysis and policy analysis.

In system dynamics modeling, the aim is to analyze the dynamic behavior pattern of the outputs, not to develop a point prediction as regression models intend to. For example, in sensitivity analysis, it would be a great help if one had the chance to specify a range for a certain parameter set; use an algorithm to group the outputs in terms of dynamic behavior patterns and obtain a classification indicating which set of parameter values yields which dynamic behavior pattern (e.g. goal seeking increase). Currently, for doing such a comprehensive parameter space search, one needs to evaluate every combination of the parameters by running the model once for each parameter set, and then labeling the resulting dynamic behavior by manual inspection. The cost of conducting such a comprehensive analysis in terms of both time and effort is apparent to any SD modeler.

Imagine a tool in your toolbox such that you feed the real data that represents the main dynamics you wish to reproduce with your model. Then this tool provides you a range of values for a set of preselected model parameters that will lead to a similar dynamic behavior. This is possible with the use of algorithms that can identify and label our model output in terms of their dynamic pattern features. We are not at that stage yet. But integration of new optimization tools to SD method and software has been listed as one of the key future challenges of the field. (Richardson 1999; Coyle 2000). In the literature there are many examples of implementations regarding the classification and clustering of dynamic behaviors.

Barlas and Kanar (1999) proposed a supervised algorithm for dynamic pattern classification. They obtain the output behavior of a model and run their algorithm for classifying the dynamic pattern. Using the mean, slope, and curvature information of the behavior, the algorithm classifies the output. The output behavior belongs to one of the 17 behavior patterns. This classifier is used for structure-oriented model validation purposes, in which model generated behavior pattern is evaluated against the modeler's expectation under certain extreme conditions (Barlas 1996).The fundamental behavior patterns are presented in Figure 1.

Using the dynamic pattern classifier of Barlas and Kanar (1999), Yücel and Barlas (2011) created a model calibration tool called *Pattern-oriented Parameter Specifier (POPS).* The user specifies the desired behavior and provides an initial set of feasible parameter values. POPS uses the initial set to create new sets of feasible values and reports the solution set which best fits to the desired output behavior. The search of the feasible parameter space to find the best parameter values is done with a genetic algorithm which is an optimization heuristic.

These tools are designed for classification purpose. There are only 17 dynamic patterns that can be recognized by the algorithms. For a given output, the algorithms determine which of the 17 classes the given output belongs to. In pattern recognition, besides classification, there is a method called clustering for grouping the similar instances. Yücel (2012) uses this method and proposes an approach for clustering the behaviors. The clustering algorithms measure the differences between instances and groups the similar ones in the same cluster. In clustering, the algorithm does not need to know the labels of the behaviors. Every kind of behavior can be clustered with the similar behaviors. In other words, the classification algorithms will fail to classify a behavior which has an unusual shape that it is not possible to label before observing it, since it does not belong to any of the 17 dynamic pattern classes. However, with clustering, one can group model outputs that carry the characteristics of this unusual

pattern together, though not being to name this group. The algorithm can provide the parameter value range yielding that unusual dynamic output behavior.

Providing a limitless output pattern evaluation opportunity, clustering seems promising as an addition to the analysis toolset of an SD modeler. In this study, a clustering algorithm is implemented and a set of alternative feature vectors in clustering basic behavior modes encountered in SD practice are evaluated. As the first case, coefficients of the polynomials fitted to the dynamic behavior are used as the features. In the second case, a given set of curves are fitted to the dynamic behavior, and the degree of fit to these curves are used as the features. The third case constructs feature vectors based on the changes in the signs of slope and curvature of the behavior. In other words, the feature vector represents the original behavior as a sequence of atomic behavior modes.

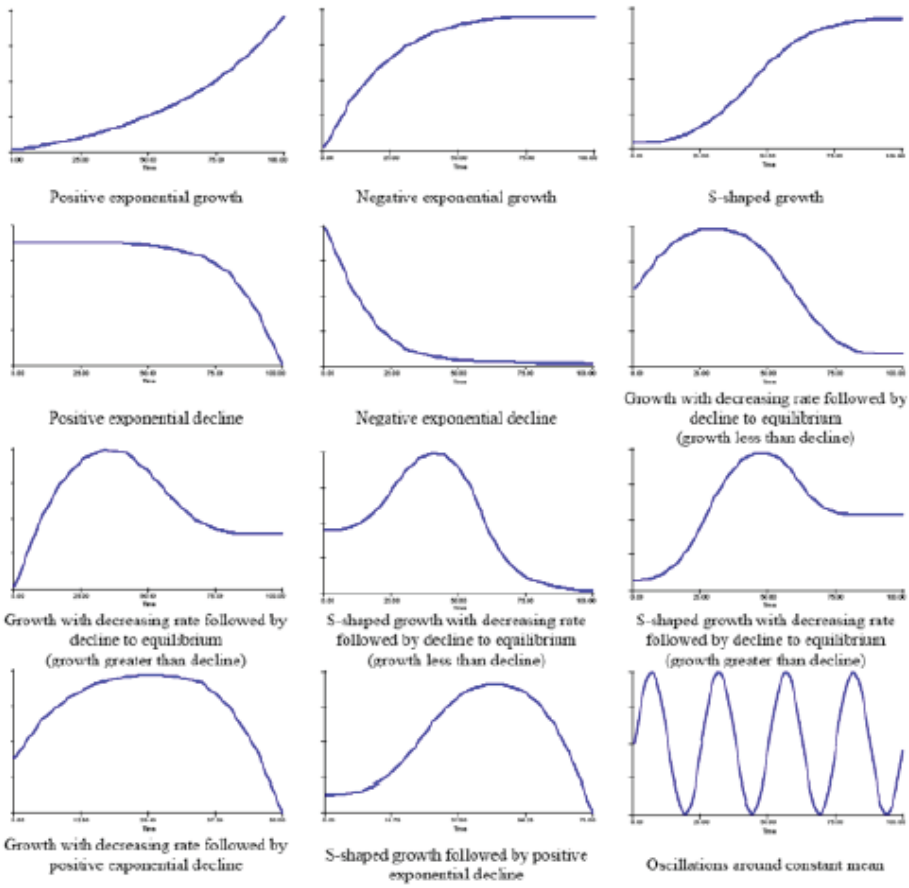


Figure 1: The fundamental behavior patterns that can be recognized by the algorithm of Barlas and Kanar (1999)

2. Pattern Recognition

Pattern recognition is the assignment of a label to a given input value by an algorithm. The inputs, which are called data instances, have certain patterns. Pattern recognition algorithms are trained with the past observations or some common features of instances. Then the algorithms are used for labeling future data instances. There are two main categories of pattern recognition methods. The first one is supervised pattern recognition where the aim is to learn a mapping from the input data instance to an output label whose correct values are provided by a supervisor. For example, a car seller can project what kind of a car the next customer may want to buy by

considering his/her certain features. If the customer has children and has a good income, he may prefer a high class family car while if he is young and rich, then his preference is most probably a sports car. Once the algorithm learns the ranges of features that result in each kind of cars, labeling the next customer with a car is done by the algorithm. In this example, each customer is an instance, the car types are the labels and the attributes of the customers (age, income, family size) are the features. The aim is to find the correct label for each instance using the features. This procedure is called classification where the class labels and their features are trained to the algorithm. The Hidden Markov Model-type classifier proposed by Barlas and Kanar (1999) makes a classification based on three features; mean, slope, and curvature of behavioral outputs of models. The algorithm is trained using a set of pre-labeled behavior instances (e.g. positive exponential growth, negative exponential growth). The second type is the unsupervised pattern recognition. In unsupervised learning, there is no supervisor. The algorithm only has the input data. The aim is to find the regularities in the input. Clustering is a commonly used unsupervised pattern recognition technique. In clustering, the aim is to find clusters or groupings of input based on the similarities of the instances. The algorithm learns how to calculate the distances between the instances and makes the clustering accordingly. The method explained by Yücel (2012) is a clustering algorithm using the information of sequential atomic behavior modes as the features of instance.

In classification, number of classes is definite. The algorithm learns the classes, and labels each instance as belonging to a particular class. In clustering, the algorithm learns only how to measure the distance between two instances. The instances which are close to each other in terms of the defined similarity measurement are grouped in the same cluster. So the instances in one group are more similar to each other than instances in different groups. In the problem of dynamic-pattern recognition, when we apply clustering, we do not need to know the classes beforehand. The behavior can be an exponential growth, an S-shaped growth or a completely different shape for which we don't have a name.

The similarity measurement is composed of a number of features of the instances which contain the necessary information to be used in determination of the similarity and dissimilarity of the instances. The vector comprises the features is called the feature vector.

In this study, an *Agglomerative Hierarchical Clustering* algorithm is used for clustering the instances. The algorithm starts with N groups where N equals to the number of instances. At each step, two groups are clustered until there is a single cluster.

For selecting the new groups to be clustered at each iteration, three approaches are tried.

> **i. Single-link clustering:** the distance between two groups is the smallest distance between all possible pair of elements of the two groups.
>
> **ii. Complete-link clustering:** the distance between two groups is the largest distance between all possible pairs.
>
> **iii. Centroid:** the distance between two groups is the distance between the centroids (means) of the two groups.

The first two methods are the most frequently used measures to choose the two closest groups to merge (Alpaydın, 2010). In this study, all the three distance

measurement methods are applied and the best solutions are reported if there is any superiority.

For calculating the distance between each instance, Euclidean distance between the values of feature vectors is employed. Euclidean distance is calculated as follows;

$$d_{x^r,x^s} = [\sum_{j=1}^{dim} (x_j^r - x_j^s)^2]^{1/2}$$

where,

x^r, x^s : feature vectors of two instances,

$x_j^r - x_j^s$: distance between j[th] dimension of x^r and x^s,

dim : number of dimension of the feature vector.

3. Different Approaches for Selecting Features from Data

The success of a clustering algorithm mainly depends on measuring the distances between the instances correctly, which in turn reduces to the problem of correctly determining a suitable set of features from data. The instances that are to be clustered in this problem are the outputs of system dynamics models, which are curves. So, the instances are essentially time-series data. The feature vector can be the time series data itself, or it can be composed of some features of the curves which represent the overall behavior pattern.

In this study, three approaches are tested for defining the feature vectors. First, various orders of polynomials are fitted to each instance (i.e. dynamic model output). The coefficients of the polynomials are taken as the features. Second, different kinds of curves (such as exponential curve) are fitted to each instance. The R-squared value of each curve-fitting is considered as one feature of the instance. Third, the slope and curvature information is used for forming the feature vector. This method is proposed by Yücel (2012).

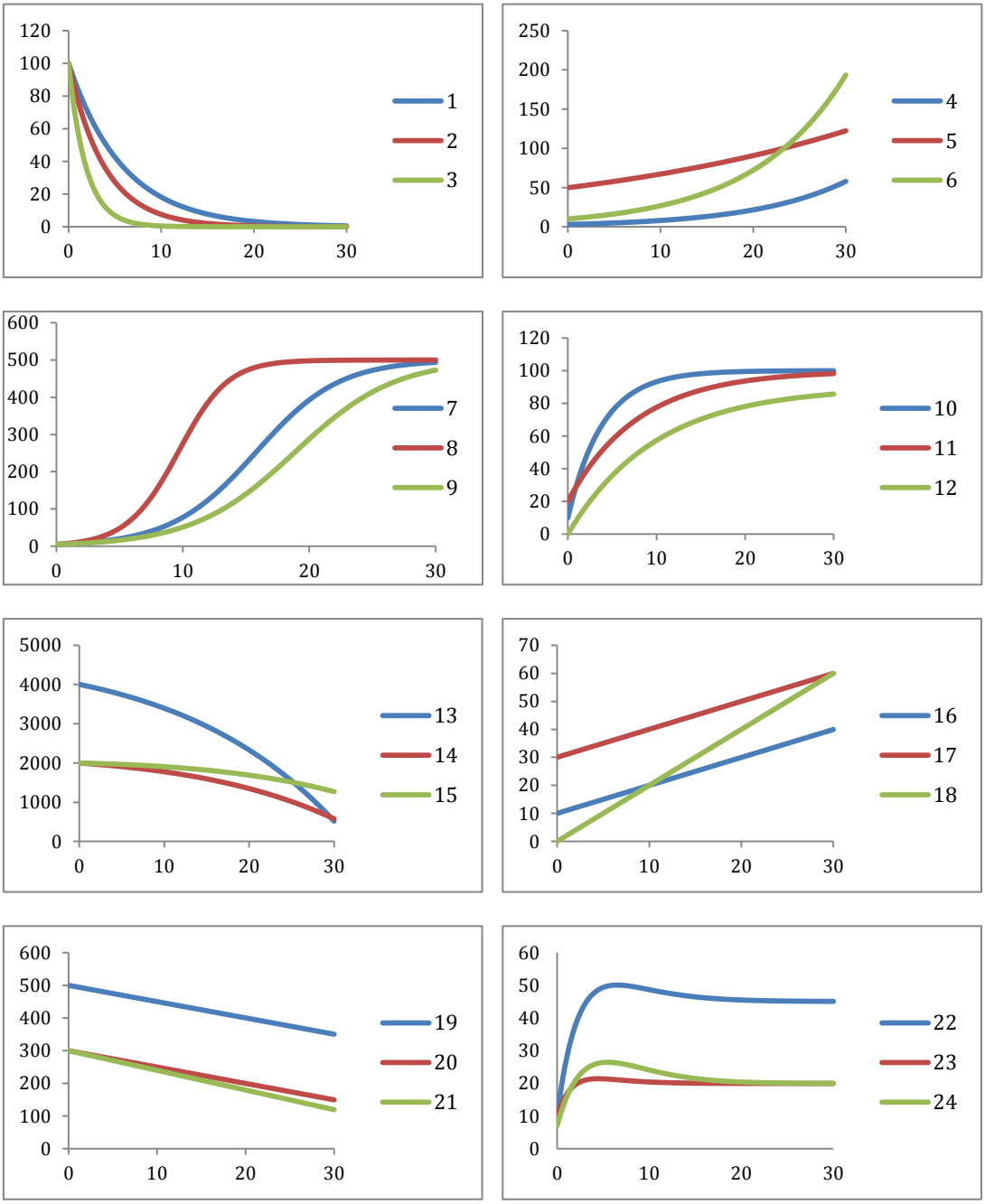
In order to evaluate these three different feature vector construction approaches, we created a sample set of dynamic patterns (i.e. instances to be clustered) representing hypothetical model outputs. These instances are presented in the following figure. There are 24 data instances to be clustered. In the ideal case, we expect to achieve a clustering result as given in the figure.

3.1. Polynomial Curve Fitting

A polynomial curve is fit to each instance and the coefficients of the polynomial are taken as the features. In the first trial, a third order polynomial is fit. The following equality represents a third order polynomial.

$$y = p_3x^3 + p_2x^2 + p_1x + p_0$$

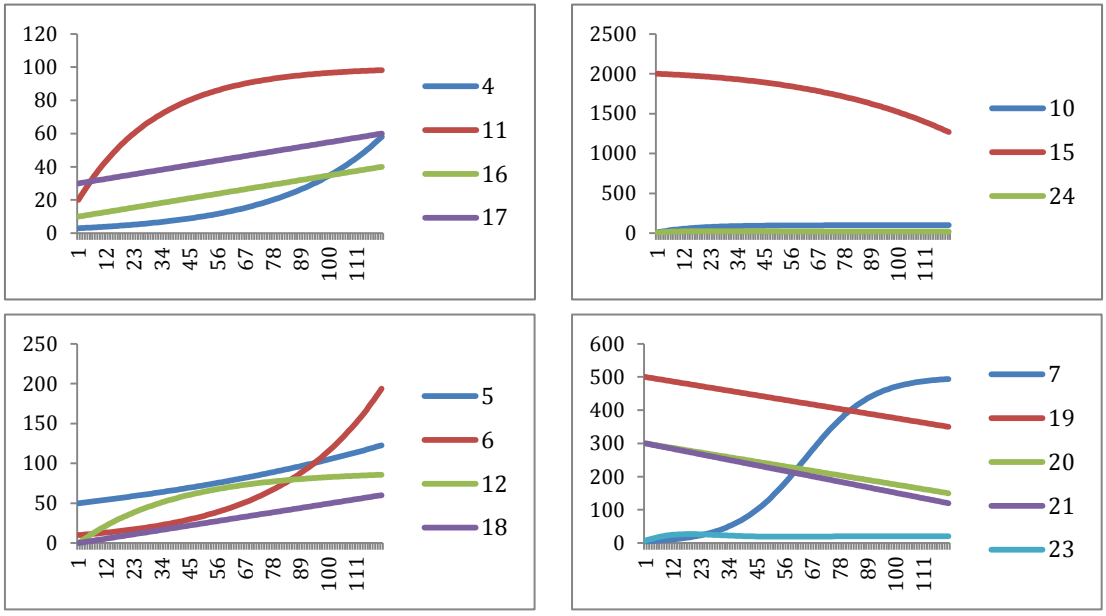
Table 1: Sample Data Instances



The coefficients $p_1, p_2, and\ p_3$ are the features of each instance. p_0 is not taken as a feature because it represents the starting point of the curves. Since the important character is the shape of the curve not the scale, it is better to ignore this information.

In Table 2, one can see the clusters obtained using these features. As may be apparent, this clustering is significantly different from the desired clusters.

Table 2: Clusters obtained by third order polynomial curve fitting

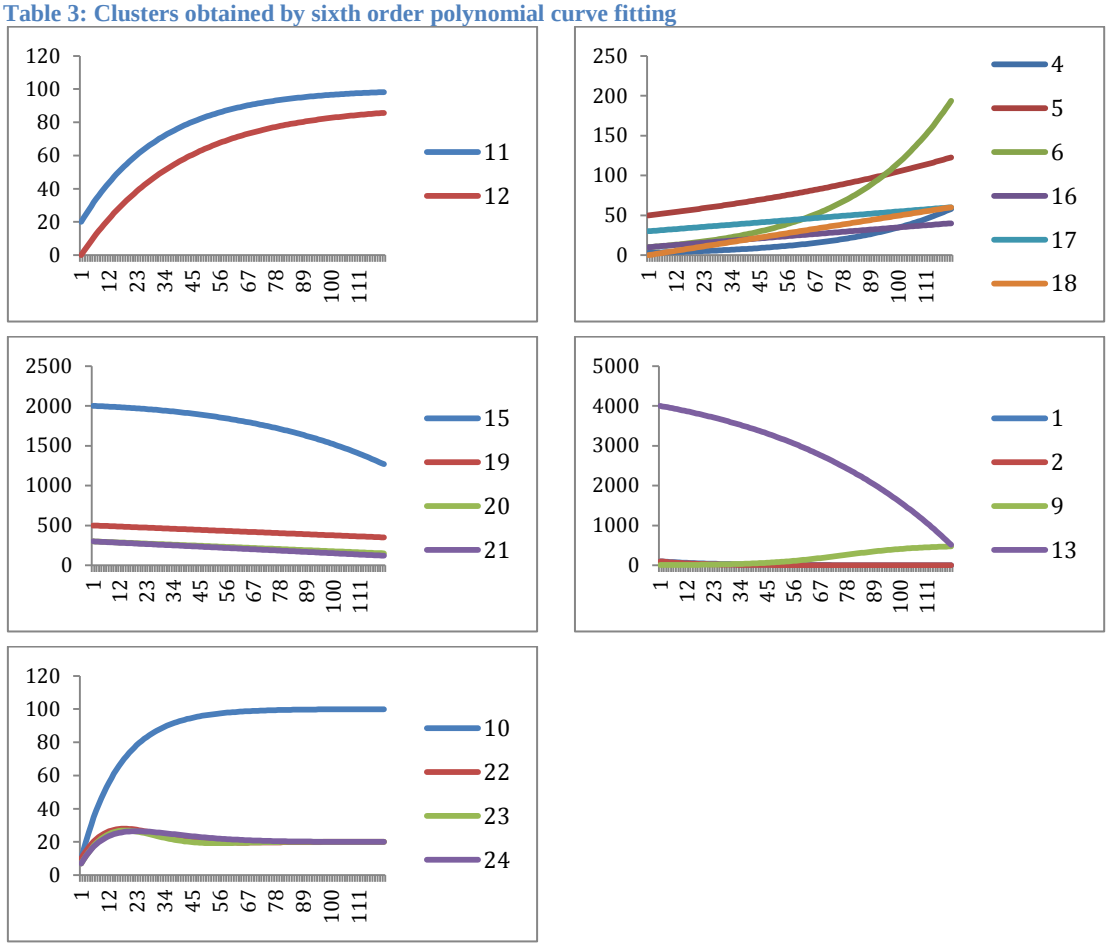


Increasing the order of polynomial curve fitting does not provide better clusters. In Table 3, the clusters obtained by taking the coefficients of a 6th order polynomial are seen. The equation below represents a sixth order polynomial. Coefficients p_6 through p_1 are taken as the features and p_0 is ignored.

$$y = p_6x^6 + p_5x^5 + p_4x^4 + p_3x^3 + p_2x^2 + p_1x + p_0$$

It is possible to understand the reasoning behind the weakness of the method by analyzing the polynomial curves fit to the instances. In Figure 2, the coefficients and R^2 values of third order polynomial curve fitting are seen. The first curve with number 10 has a steeper slope than the curves with number 11 and 12. The R^2 of curve 10 is lower than the others. If one visually examines the trend line on the curve, it is obvious that the polynomial trend line has a completely different shape than the curve. The situation is not the same for the curves 11 and 12. The R^2 values are much better and the trend line seems to fit well to the curves. The coefficients of the trend lines provide information as well. The coefficient behind the x0 term (the rightmost coefficient in the trendline equation) is not considered since it represents the starting point of a curve. The first coefficient taken as a feature is the coefficient behind the x term. The first feature of curve 10 is 13.13. For the curves 11 and 12 that coefficient is around 8. This coefficient is a kind of a measure of the steepness of the slope. All the other coefficients (the coefficients behind x2 and x3 terms) are close to zero. Therefore the dominant factor in distance calculation is the first coefficient (the coefficient behind the x term). This is an indication of the fact that when the coefficients of the polynomials are taken as the features of the instances in clustering, curves with steep slopes are to be labeled different than the curves with flat slopes.

Table 3: Clusters obtained by sixth order polynomial curve fitting



In Figure 3, the fitting results of sixth order polynomials are seen. All of the R^2 values are 1. However, when the coefficients behind the x terms are examined, it is seen that curve 10 with coefficient 22.536 seems different than curve 11 with coefficient 10.141 and curve 12 with coefficient 9.1086. So it can be concluded that the polynomial curve fitting provides a clustering based on the steepness of the slopes.

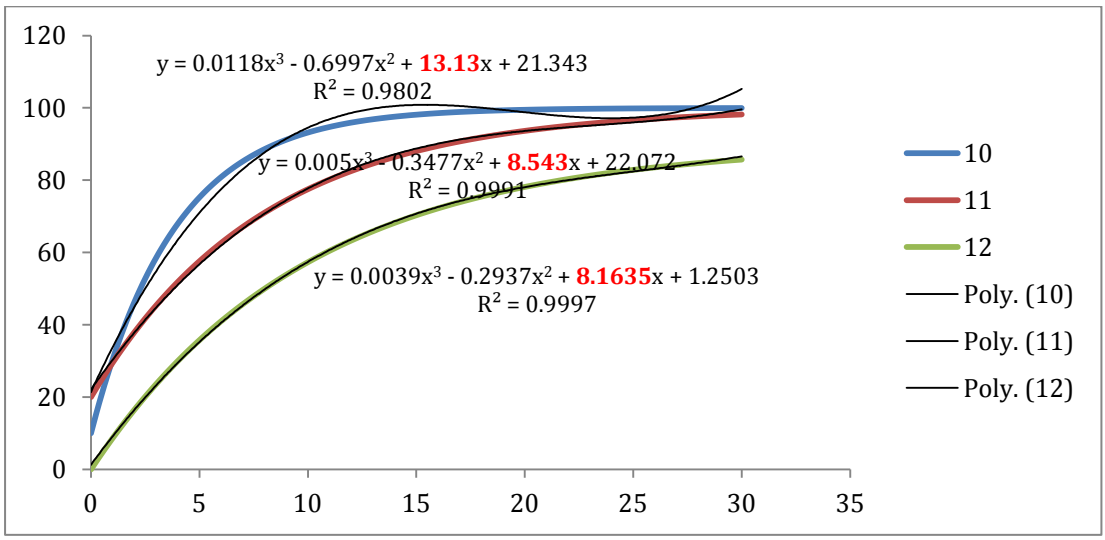


Figure 2: Third order polynomial curve fitting to a goal seeking increase

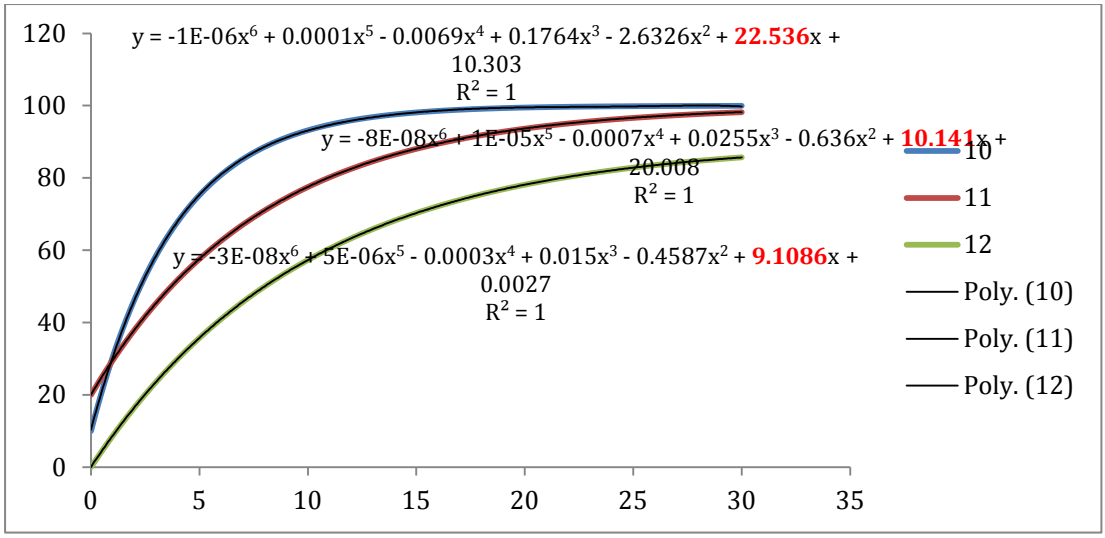


Figure 3: Sixth order polynomial curve fitting to a goal seeking increase

Observe the S-shaped growth curves. In Figure 4, the third order polynomial fit to curve 8 with a very low R^2 value. The shape of the trend line is completely different than the shape of the curve. As in the negative exponential growth case in the figure above, the S-shaped growth with number 8 is steeper than the other two. The polynomial curve fitting method fails to fit to this curve. In Figure 5, the sixth order polynomial curve fittings are seen. The R^2 values are better. However, the coefficients are far from being informative in terms of similarity.

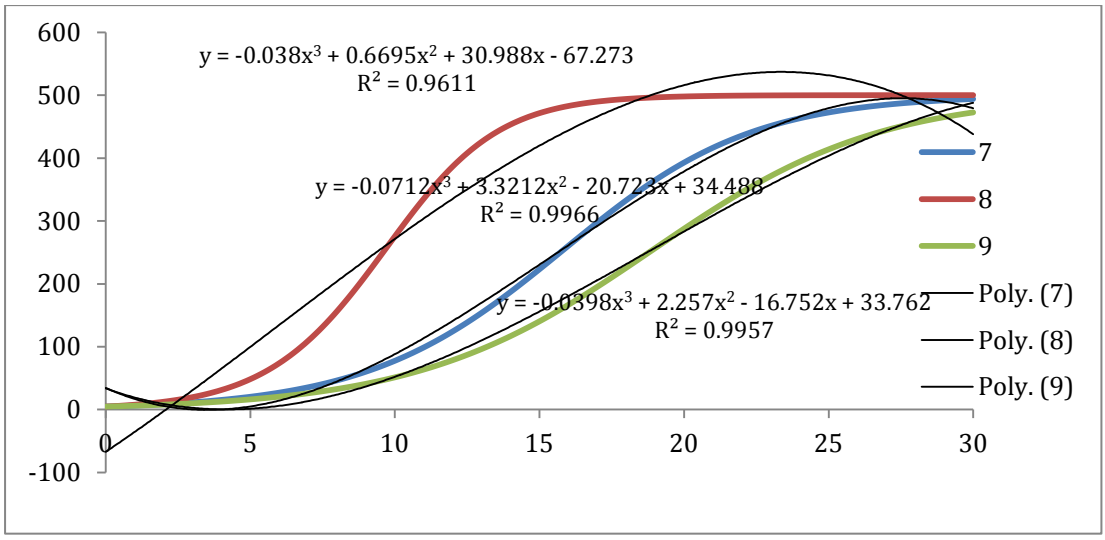


Figure 4: Third order polynomial curve fitting to an S-shaped growth

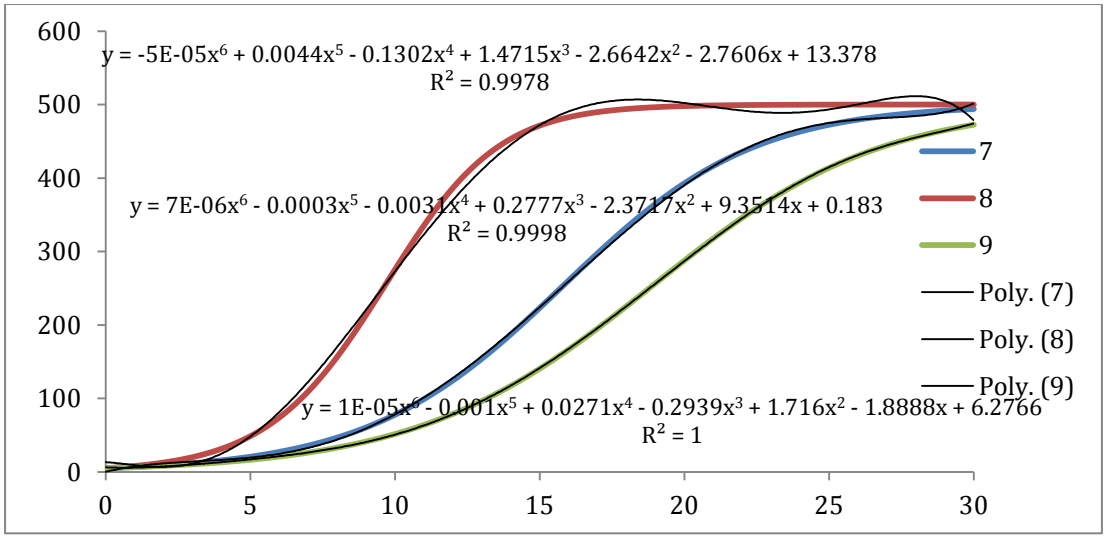


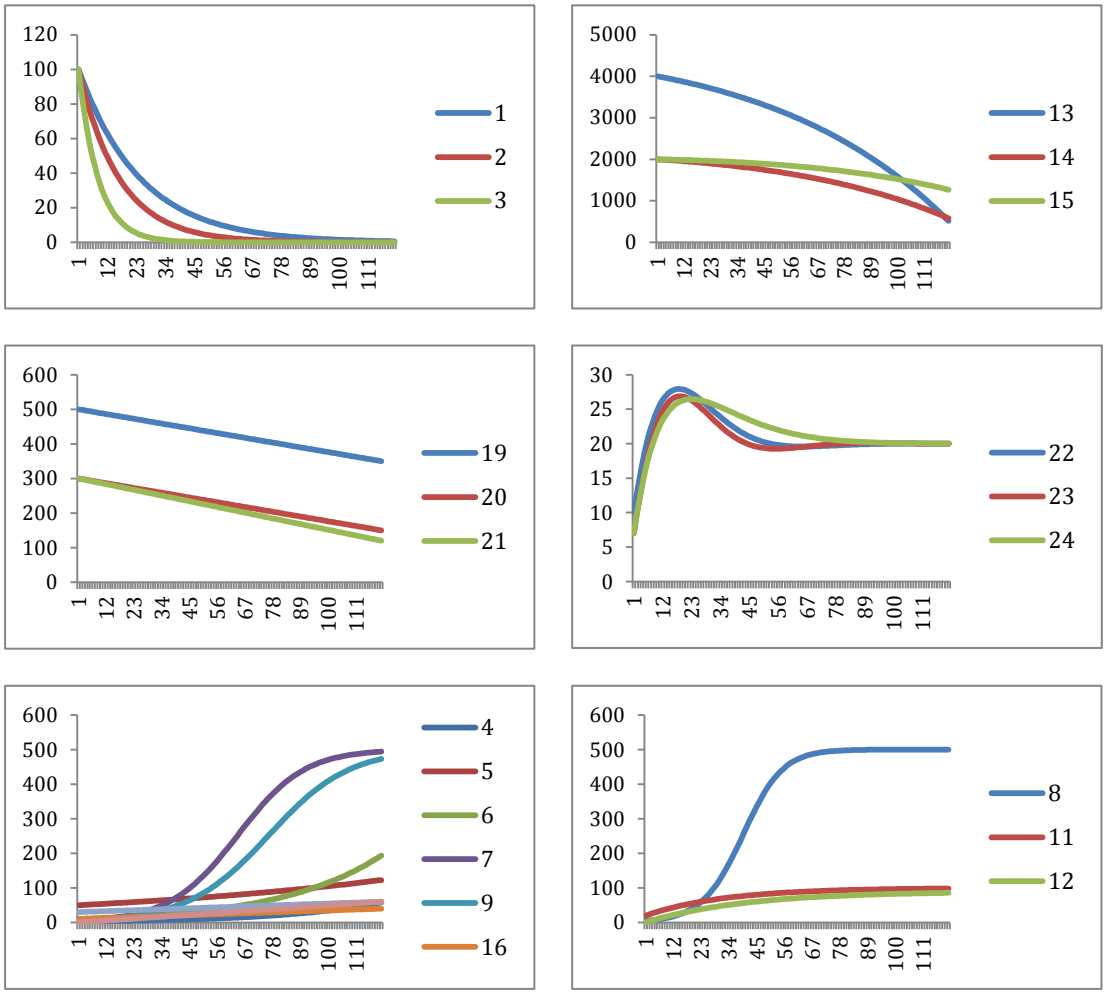
Figure 5: Sixth order polynomial curve fitting to an S-shaped growth

3. Goodness of Fit

Different curves are fitted to data and the R-squared values of each curve fitting are taken as a feature. *Power*, *Exponential*, *Linear* (Polynomial 1), *Third order polynomial* and *Gaussian* curves are fitted. R-squared values are taken as the features of the instances.

The clusters obtained by this method are much more successful than the clusters provided by the polynomial curve fitting method which is explained in section 2. The algorithm provides 7 clusters that you can see in Table 4. The first four clusters are correct. The last two have problems. Instance 10 is clustered in a single group. Some variations of the method are also tried as keeping only the maximum R^2 values and equating the other values to zero; keeping only the maximum two R^2 values and equating the others to zero. None of the trials provided a better clustering.

Table 4: Clusters obtained by the goodness of fit method



3.3. Slope & Curvature information

Yücel (2012) proposes a method for measuring the similarity of the instances based on qualitative features of these instances. He emphasized the fact that the characteristic of a curve lays in the sequence of atomic behavior modes that it consists of. Each atomic behavior is represented by a slope and curvature pair. The sign of slope and curvature is sufficient to define an atomic behavior. The numeric values of slope and curvature provide information regarding the steepness and speed of a curve. If one wants to cluster the curves only with respect to the shape but not the steepness and speed of the shape, then only keeping the sign information of slope and curvature is sufficient. The sequence of the slope-curvature pairs determines the complete behavior. There are a limited number of atomic behavior modes as shown in Table 5.

Table 5: The atomic behavior modes (Yücel, 2012)

		1st Derivative (Slope)		
		-	0	+
2nd Derivative (Curvature)	-	Exponential Decrease (-,-)	Constant* (0,-)	Goal-Seeking Increase (+,-)
	0	Linear Decrease (-,0)	Constant (0,0)	Linear Increase (+,0)
	+	Goal-Seeking Decrease (-,+)	Constant** (0,+)	Exponential Increase (+,+)

*Infinitely short period just before a decrease phase
** Infinitely short period just before an increase phase

Any curve must be composed of a sequencing of the atomic behavior modes. For example, in an S-Shaped growth, there are three of the atomic behavior modes in Table 1. An S-shaped growth starts constant (0,0), continues with an exponential increase (+,+) and ends with a goal seeking increase (+,-) atomic behavior. So, an S-shaped growth is represented with (0,0), (+,+), (+,-). Each of the signs can be taken as a feature. A (+) sign can be quantified by (1) and a (-) sign can be quantified by (-1). So the feature vector of an S-shaped growth would be in the following form; [0,0,1,1,1,-1]. Each curve can be represented as a sequence composed of 0, 1 and -1.

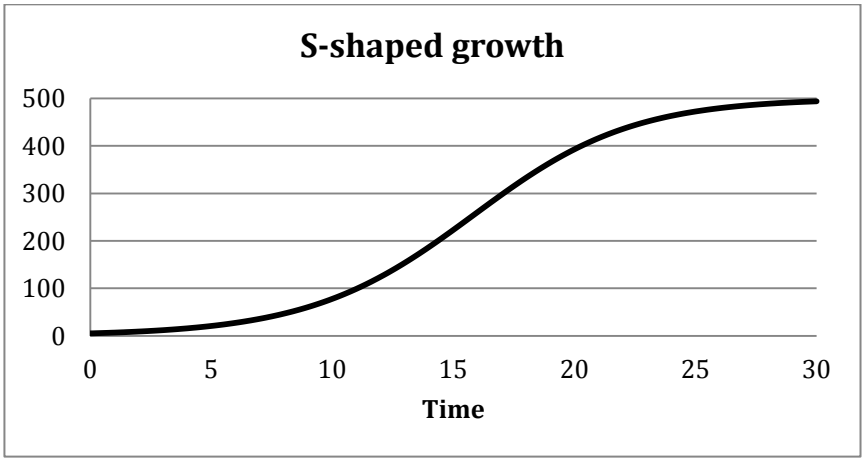


Figure 6: An example dynamic behaviour

The slope and curvature is approximated from the time series data. The following equations are proposed by Yücel (2012) for the approximation.

$$slope(t) = x(t) - x(t-1)$$
$$curvature(t) = slope(t) - slope(t-1)$$

The number of repeating atomic behavior modes of the curves other than the oscillations, are at most 4. An instance with only one atomic behavior has only two features. For example the feature vector of an exponential growth is (1,1). However, an S-shaped growth has three atomic behaviors leading to six features (0,0,1,1,1,-1). In calculating the distances between instances, we employed Euclidean distance as

explained in section 2. The length of the feature vectors should be the same for distance calculation. In order to equate the lengths of the feature vectors, the short feature vectors need to be lengthen. Yücel (2012) proposes a sister creation method for this purpose. A sister of an original feature vector is identical to the original one in terms of the sequence of atomic behaviors but it is longer in length. Yücel (2012) proposed to create sisters from each atomic behavior, and then to select one of the sisters. In this study, we created sisters only from the last observed atomic behavior and we experimentally found that this is sufficient for the sake of clustering, besides being more timesaving. The maximum number of atomic behaviors in a curve is 4, having 8 features. So we extend each feature vector to 8 dimensions. The last observed atomic behavior is repeated until completing 4 atomic behaviors. For example, for a curve having an exponential increase behavior, the original feature vector is (1,1). We add three more (1,1) at the tail of the feature vector to complete the dimension to 8. Similarly suppose there is a curve with the following three atomic behaviors; (1,1), (+1,-1), (-1,-1). We add the last observed atomic behavior which is (-1,-1) and make it an 8-dimensional feature vector. In Table 6, example feature vectors are provided.

Table 6: Example feature vectors of instances

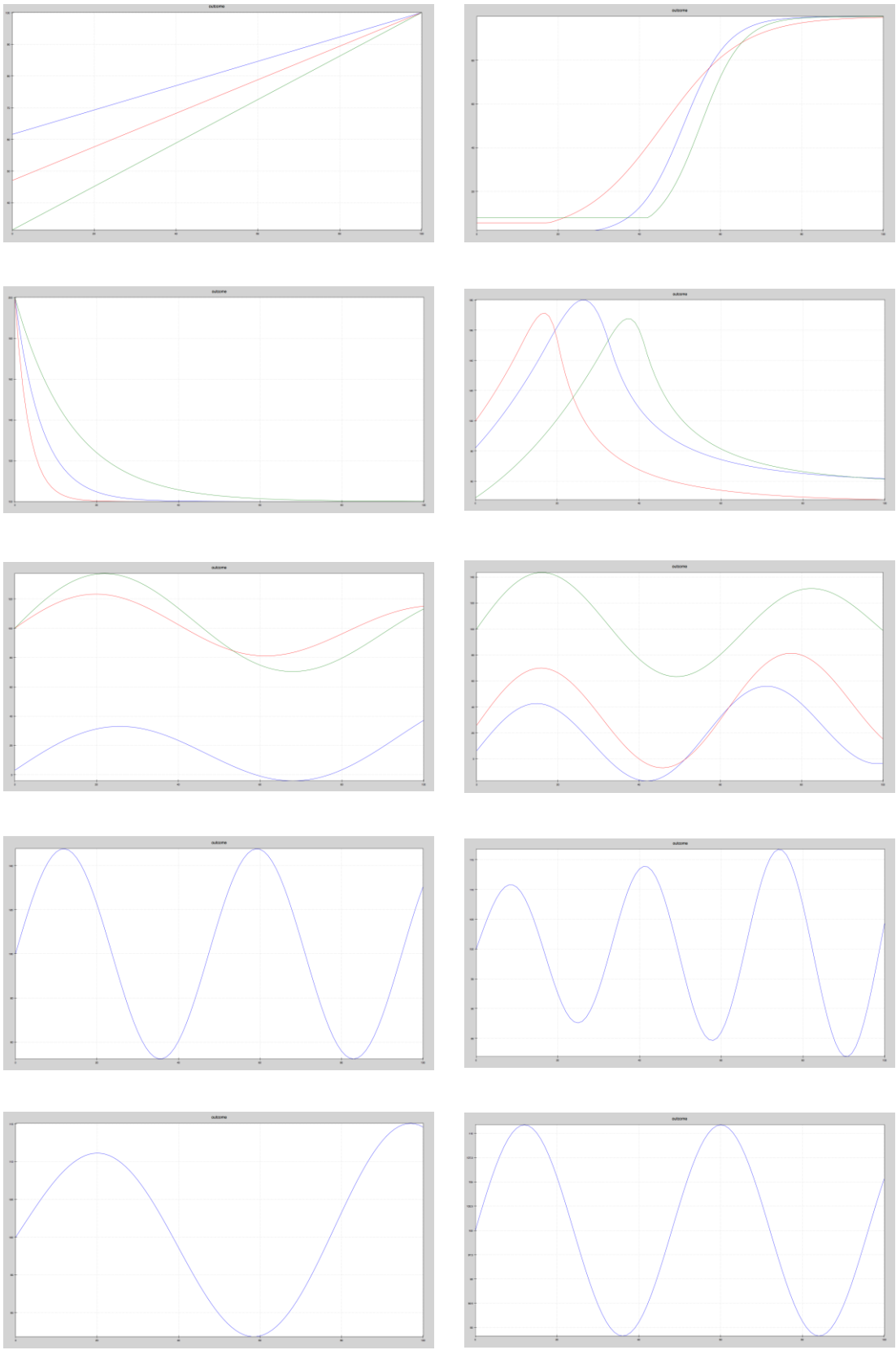
Original atomic behavior mode sequence	Extended feature vector
[1,1]	[1,1, 1,1, 1,1, 1,1]
[1,-1]	[1,-1, 1,-1, 1,-1, 1,-1]
[1,1, 1,-1,]	[1,1, 1,-1, 1,-1, 1,-1]

Using the extended feature vectors, the hierarchical clustering algorithm clusters all the instances correctly.

4. Improvements in Slope-Curvature Method

The feature selection method proposed by Yücel is found to be successful in determining features. However, the algorithm proposed by Yücel (2012) does not recognize the oscillatory behaviors correctly. In order to visually examine the mis-clustering of the instances by that method, a 24-instance data set is created and his algorithm is run. The resulting clusters are seen in Table 7. The four of the clusters in the table are correct. But the oscillatory behaviors are considered as single clusters. The algorithm cannot recognize the similarity between the oscillating instances. In this section a method for recognizing and clustering the oscillations is proposed using the slope and curvature information as the features of the instances.

Table 7: Clusters obtained by Yücel (2012) with the 24-instance data set.



A preprocessing phase is designed for detecting the oscillatory instances and defining suitable feature vectors for them. First, the algorithm is trained for detecting if an

instance is an oscillatory one. For all of the dynamic behaviors other than oscillations, the number of sign changes is limited. If the number of sign changes is more than a threshold value, the algorithm labels the instance as an oscillation. A new feature column is added to the feature matrix for labeling a behavior as an oscillation or not. The oscillatory instances have value 1 and the others have value 0 in that column.

There are various kinds of oscillations. The amplitude of an oscillation can be increasing (growing oscillation), constant (constant oscillation), or decreasing (stable oscillation). Moreover, an oscillation can have an upward or downward trend. For keeping track of these two features, two more columns are added to the feature matrix.

The new features, defined specially for the oscillatory instances, are *trend* and *amplitude*. The features, taking +1, 0, -1 values, are explained in Table 8.

Table 8: *Trend* and *Amplitude Change* quantization of oscillatory behaviors.

		Trend		
		-1	0	+1
Amplitude	-1	Stable Oscillation with decreasing trend	Stable Oscillation with no trend	Stable Oscillation with increasing trend
	0	Constant oscillation with decreasing trend	Constant oscillation with no trend	Constant oscillation with increasing trend
	+1	Growing Oscillation with decreasing trend	Growing Oscillation with no trend	Growing Oscillation with increasing trend

For determining the trend and amplitude position of the instances, the following attributes of the instances are used;

first max: the value of the curve at the first maximum peak

first min: the value of the curve at the first minimum peak

last max: the value of the curve at the last maximum peak

last min: the value of the curve at the last minimum peak

max: the maximum value of the oscillation

min: the minimum value of the oscillation

If the amplitude of the first period (peak period), which can be calculated as “first max-first min”, is not 15% greater than the amplitude of the last period, than the oscillation is stated as a constant oscillation. If difference in the amplitudes is greater than 15%, than the algorithm checks the difference between first amplitude (first max – first min) and the maximum amplitude (max - min). If the first amplitude is larger than the maximum amplitude, than the oscillation is a stable oscillation. If the first amplitude is smaller than the maximum amplitude, than the oscillation is a growing oscillation.

The trend position of the instances is determined as follows; if the value of the center of the first period is at least 4.5% greater than the value of the center of the last period, than the oscillation has a downward trend. In the opposite way; if the value of

the center of the first period is at least 4.5% less than the value of the center of the last period, than the oscillation has an upward trend. If the difference is lower than 4.5%, the oscillation has no trend.

For every kind of oscillation, the features representing the atomic behaviors are set the same sequence of atomic behavior modes which is (0,0),(-1,-1),(-1,+1),(0,0). This sequence of atomic behaviors stands for the behavior in Figure 7, which is a common component in each oscillation.

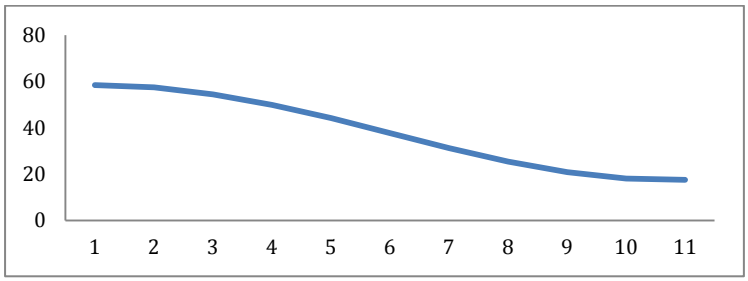


Figure 7: A short dynamic pattern piece common in all of the oscillations.

The trend and amplitude features are also added to the feature vectors of other instances. All the trend and amplitude values are assigned "0" in the preprocessing.

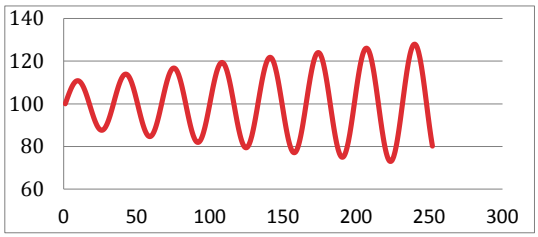
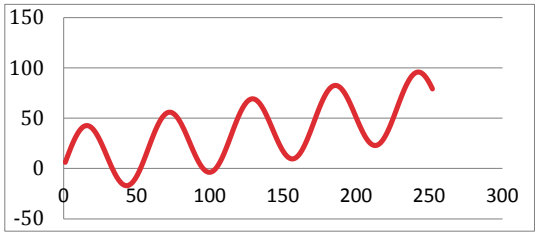
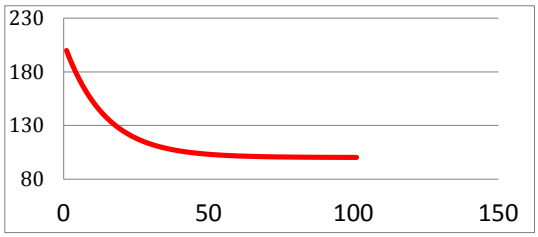
The final feature matrix, which is composed of each feature vector, is presented in Table 9.

Table 9: Feature Matrix Format

				Atomic Behavior Modes			
Instances Nodjgdjgrı	Oscillation	Trend	Amplitude change	Atomic 1	Atomic 2	Atomic 3	Atomic 4
Instance 1 Instance 2 …….. Instance n							

In Table 10, there are examples of feature vectors of oscillating and non-oscillating instances. The amplitude and trend elements of the feature vectors are shown bold and red.

Table 10: Example feature vectors and corresponding behaviors

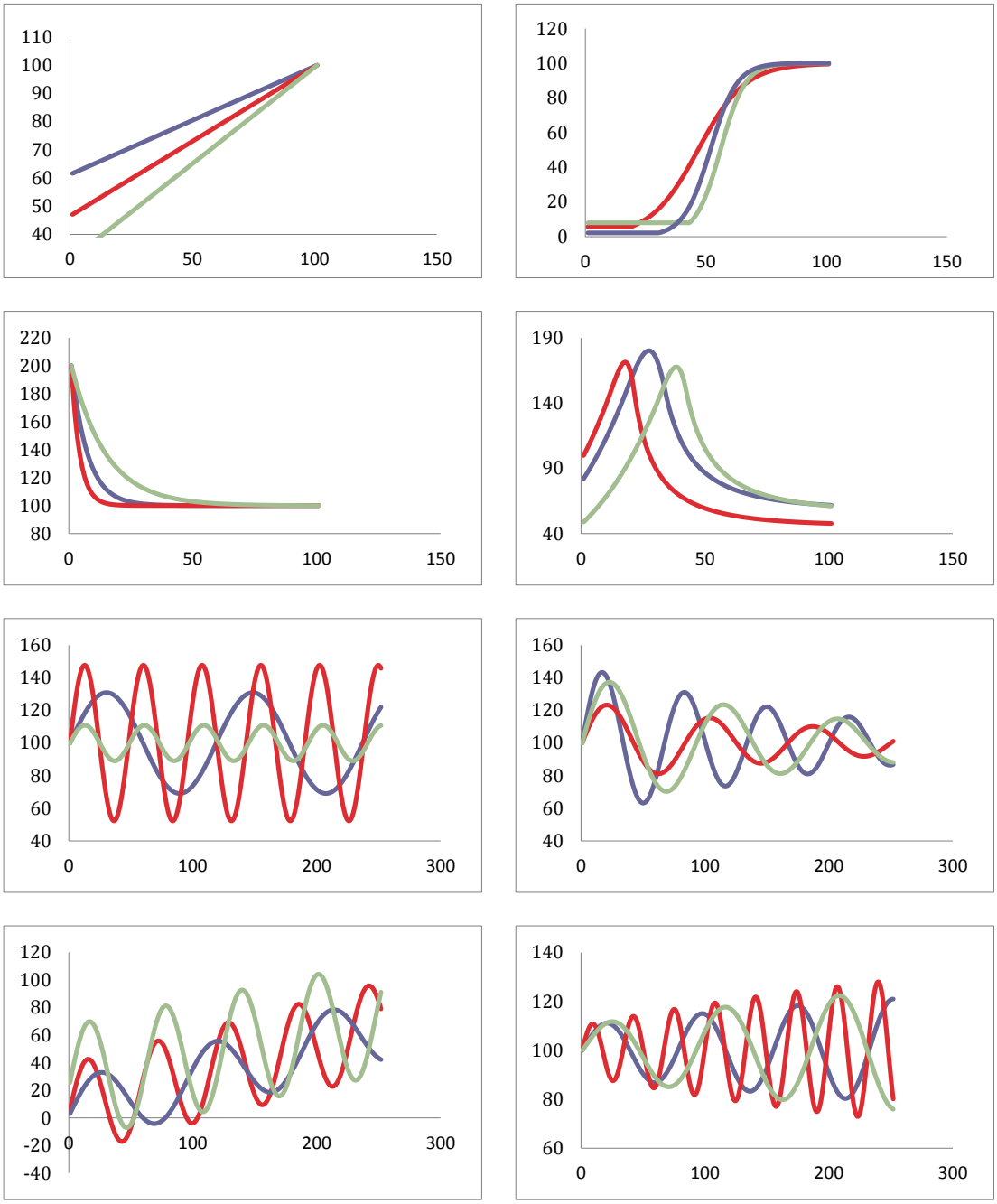
	Feature vector	Shape
Growing Oscillation with no trend	[1,**1,0,**0,0,-1,-1,-1,1,0,0]	
Constant oscillation with increasing trend	[1,**0,1**,0,0,-1,-1,-1,1,0,0]	
Non-oscillating behavior	[0,**0,0**,-1,1,-1,1,-1,1,-1,1]	

In summary, some of the values of the feature matrix are filled in the preprocessing. The oscillation feature is first assigned 1 or 0 according to being an oscillating or non-oscillating instance. The trend and amplitude features of non-oscillating instances are assigned 0. The atomic behavior modes features of oscillations are assigned the signs of a portion of a one period-oscillation.

At the end of the preprocessing, the whole feature matrix is ready to be used in the clustering algorithm. A 300-instance data is used for validation. The algorithm successfully clusters the data. For visual examination, the 24-instance sample data set which is presented at the beginning of this section is clustered with the new algorithm. The final clusters are presented in Table 11. The algorithm clusters the instances perfectly.

A second improvement of this algorithm is that the number of clusters is determined automatically by the algorithm. In the algorithm, the *mean* and *variance* of each cluster is calculated at each iteration (each addition of a new instance into a cluster). *Mean* of each cluster is a vector composed of the means of each feature of the instances in that cluster. *Variance* is defined to be the sum of the differences of each instance from the mean of the cluster. A maximum number is assigned for the maximum allowed variance and the variance of each cluster is kept under the maximum allowed variance value. The method stops with a reasonable number clusters.

Table 11: Clusters obtained by the final clustering algorithm



Discussion

As the features of the instances, using the slope and curvature pairs representing the atomic behavior modes is found to be the best-working method out of the methods analyzed in this study. The challenging issue in this method is the correct-determination of the atomic behavior modes. It is impossible to use the raw data because of the inherent noise. The data needs to be filtered/smoothed. Though, the filtering mechanism used in this study is a limited one. It may not be sufficient to smooth a noisier data set. Various filtering methods are examined in the study. Sometimes the shape of the curve is changed because of the filtering. An exponential moving average is found to be very successful method in smoothing the data but it

causes changes in the shape of the curves in particular cases. For example, when exponential moving average is applied, the head of a linear curve looks like an exponential growth. Similarly, a goal seeking increase gains a short-length portion of exponential increase and looks like an S-shaped growth. The shape of the curves seems to be changed but the changed portion of the curves is very short in length. It is possible to train the algorithm to ignore such short-length behaviors. However, this may cause an information loss for other curves. Remember that this algorithm is designed for clustering the instances so that a modeler will be able to recognize the resulting behavioral clusters of the output of his model. A short-length behavior may be important for some kinds of models depending on the purpose of the model. At this stage, we do not prefer to ignore these behaviors.

A completely different method for finding the slope and curvature is presented in Barlas and Kanar (1999). For any instance, they divide the curve into 6 equal segments. They fit a first order polynomial to each segment. The second coefficient of the first order polynomial is taken as the slope of the segment. Similarly, a second order polynomial is fit to each segment and the curvature value is obtained using the coefficients of the second order polynomial. This method seems promising by not necessitating any filtering on data. However, the method seems to need some improvements. Dividing a curve into equal-length segments may cause troubles. Based on the experience gained in this study, it is known that, the slope-sign is easier to determine than the curvature sign when the data is noisy. So we may use the slope-sign as proposed in this study but determine the curvature sign by employing part of the method proposed by Barlas and Kanar (1999). So, the alternative method that we propose is to examine the changes in the sign of the slope with the method used in this study. Afterwards, for finding the curvature, the curves are divided into segments. The curve is divided from the points where a slope-sign change occurs. Between the slope-sign-change points, second order polynomials are fit. The curvature of each segment is determined by using the coefficients of the second order polynomial. Implementing this alternative procedure constitutes the first step in future research.

Besides the challenges in filtering and slope-curvature determination, it is worthy to talk about the distance calculation. In this study, the widely used Euclidean distance method is employed for calculating the distance between instances. Euclidean distance assumes that the dominance of each feature is the same. Though, it is possible to assign different weights to the features and change the dominance of the features according to their importance in similarity measurement. The extension of the study includes further seeking for new features and reasonably assigning their weights in the weighted Euclidean distance formulation.

5. Conclusion

Considering the need for automatically evaluating the outputs of system dynamics models, we seek an algorithm for clustering a wide range of dynamic patterns. The crucial issue in clustering algorithm development is the design of a proper feature vector. The selected features of the output instances should represent the overall behavior pattern. Different approaches for designing feature vectors are examined using a sample dynamic behavior set. As the first approach, polynomial curves are fit to the data instances and the coefficients of the polynomials are taken as the features. The second approach is fitting various kinds of curves for each data instance and taking the R-square values as features regarding the R-square value as an indicator of goodness of fit. The third approach is adopted from a previous study and found to be

the best way in designing the feature vector. This approach simply describes a dynamic behavior as a sequence of atomic behavior modes. The atomic behavior modes are represented by pairs of slope and curvature.

Adopting the third approach for designing the feature vectors, we develop a clustering algorithm with a preprocessing stage for labeling the oscillatory behaviors which are challenging to handle. The feature vector is extended to deal with the two attributes of the oscillations; trend and amplitude change. The algorithm successfully clusters the dynamic behaviors. The validation of the algorithm is provided with a 300-instance data set.

In the discussion section, possible improvements for the methods used in various stages of the algorithm are discussed. The future research looks promising in terms of successful implementation of the algorithm on larger data sets and improvements via alternative methods.

Acknowledgements:

We thank Yaman Barlas and Ethem Alpaydın from Boğaziçi University for their valuable comments and suggestions.

References

Alpaydın, E. (2010). *Introduction To Machine Learning* (2nd ed.). Cambridge: MIT Press.

Barlas, Y. (1996). "Formal aspects of model validity and validation in system dynamics." System Dynamics Review **12**(3): 183-210.

Barlas, Y. and Kanar, K. (1999). A dynamic pattern–oriented test for model validation. *4th System Science European Congress*. Valencia, Spain.

Coyle, G. (2000). "Qualitative and quantitative modelling in system dynamics: some

research questions." System Dynamics Review **16**(3): 225-244.

Richardson, G. P. (1999). "Reflections for the future of system dynamics."Journal of the Operational Research Society." **50**(4): 440-449.Yücel, G. (2012). A novel way to measure (dis)similarity between model behaviors based on dynamic pattern features. *The 30th International Conference of the System Dynamics Society.* St. Gallen.

Yücel, G. and Barlas Y. (2011). "Automated parameter specification in dynamic feedback models based on behavior pattern features." System Dynamics Review 27(2): 195-215.