Learning Multi-Dimensional Functions: Gas Turbine Engine Modeling ⋆

Chris Drummond

Institute for Information Technology
National Research Council Canada
Ottawa, Ontario, Canada, K1A 0R6
Chris.Drummond@nrc-cnrc.gc.ca

Abstract. This paper shows how multi-dimensional functions, describing the operation of complex equipment, can be learned. The functions are points in a shape space, each produced by morphing a prototypical function located at its origin. The prototypical function and the space's dimensions, which define morphological operations, are learned from a set of existing functions. New ones are generated by averaging the coordinates of similar functions and using these to morph the prototype appropriately. This paper discusses applying this approach to learning new functions for components of gas turbine engines. Experiments on a set of compressor maps, multi-dimensional functions relating the performance parameters of a compressor, show that it more accurately transforms old maps, into new ones, than existing methods.

1 Introduction

This paper discusses the inductive learning of predictive models where the output is not a label, nor a continuous value but a multi-dimensional function. It proposes using morphological analysis techniques [2] to learn a shape space capturing the common characteristics of a set of existing functions. These functions, and new ones, are points in this space, each produced by morphing a prototypical function located at its origin. The dimensions of the shape space define individual morphing operators. In this paper, the functions represent components of a gas turbine engine, primarily one used to power aircraft. However, the idea of morphing functions to be useful in new situations should readily generalize to other applications. Certainly, the expectation is that it can be used to learn functions that describe the operation of other complex equipment. It should also be useful in applications of reinforcement learning, where functions representing the solution to tasks [1] could be morphed to form solutions to other tasks.

The focus, here, is on the components of gas turbine engines, such as compressors, combustion chambers and turbines. These are described by performance parameter maps, which give the relationship between the input and output parameters. Component models based on these maps can be combined to produce an accurate simulation of an engine [12]. This simulation is an important tool in engine design and diagnostics. An essential, yet time and resource consuming activity, is to generate such maps, preferably prior to prototypes of the engine being available. A common approach, as is typical in much of engineering design, is to refine an existing solution to a similar problem. A current method used for producing a new compressor map is to take one for a similar compressor, according to expert judgment, and to multiply each of its dimensions by a single factor [11]. This is satisfactory if the compressor is "sufficiently similar", an ill-defined concept. More recently more sophisticated methods, with broader applicability and more accurate approximations, have been proposed [7, 6].

Selecting an existing map and then adapting it to apply to a new problem is very reminiscent of case based learning. This work certainly has many similarities with research into learning and adapting cases representing the real valued parameters of manufacturing equipment [4]. Learning how to morph maps based on the similarities between existing maps is analogous to learning adaptation rules from cases [5]. What sets this work apart is that the cases are multi-dimensional functions and more complex adaptive operators are needed. Overall, this work might best be compared to that combining instance-based and model-based learning. Quinlan's [9] used model based learning to generalize across multiple instances, an important component of this work.

The original motivation for this work was to exploit engineering knowledge within a learning algorithm used to predict, and diagnose, engine faults [8]. But, here, I argue that as a scaling procedure it stands on its own merits. Not only, as the experimental section will show, does it produce maps that are more than competitive with those produced by a recently introduced non-linear scaling method [7] but it achieves this through generalization of examples rather than a human analysis of the commonalities of compressors. This gives the approach a number of clear advantages. More examples can be added to improve accuracy. Extra information about the compressor can be added to improve accuracy. The approach should generalize to other components within the engine and to other complex equipment. The approach should be applicable not only in modeling new engines but also in accurately modeling older engines.

2 Generating New Functions

To generate a new function, an existing function must be selected and modified. For a compressor, two properties are of particular importance; the pressure rise from input to output and the mass flow, the rate of air flow through the compressor. In this work, the most similar map is selected using a distance measure calculated from these two properties. There are different types of compressor in a turbofan engine, the most common engine configuration used in commercial aircraft. At front of the engine, see figure 1, is the fan, this drives a large portion of the air around the core of the engine providing most of the thrust (at least in commercial aircraft). The air passing through the core is pressurized by the high pressure compressor. The core air is driven into the combustion chamber.

It there mixes with the fuel and is ignited, the resultant high velocity gases pass through the two turbines. The turbines extract energy from the flow and drive the fan and the high pressure compressor.

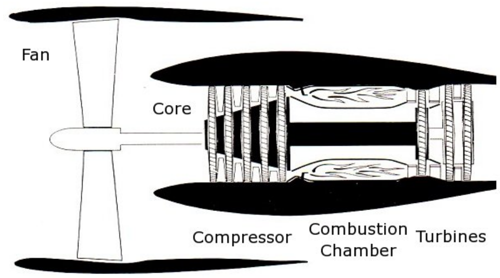

Fig. 1. A Gas Turbine Turbofan Engine

Not only are there different types of compressor, they also vary in the number and complexity of the rows of blades depending on the size and thrust of the engine. To give some feel for the range of variability, figures 2 and 3 show two examples of quite different compressor maps. Figure 2 is for a fan; figure 3 is for a high pressure compressor. The y-axis is the pressure ratio; the x-axis is the mass flow. The fan moves a lot of air, but produces only a small pressure rise. The high pressure compressor has a much larger pressure rise although it moves much less air. The data points, the black dots in the figures, have recorded values of mass flow, pressure ratio and efficiency, temperature rise for a given pressure rise, for the particular engine. They form lines, called speed lines, recorded from operation of the compressor at selected rotational speeds. Two dimensional b-splines are used here to generate the complete function. This produces the three dimensional function shown by the contours, representing the efficiency of the compressor in terms of the pressure ratio and mass flow. As the function also includes information about the compressor speed, it is really four dimensional. This representation, using speed lines, is the conventional mechanical engineering one, as it can be readily projected onto two dimensions making it much more easily interpretable by humans. The standard way to modify an existing map to generate a new one, in use for over 30 years [11], is to multiply each dimension by its own scale factor.

2.1 Non-linear Scaling

Looking back at figures 2 and 3, we can see that the general shape of the speed lines and the efficiency contours are similar but by no means the same. In figure 2 the contours are more spread out and each speed line covers a broader range on the x-axis. The shape of the contours also differs: in figure 2 a line down their center has a quite appreciable curve, in figure 3 this line is much straighter. The

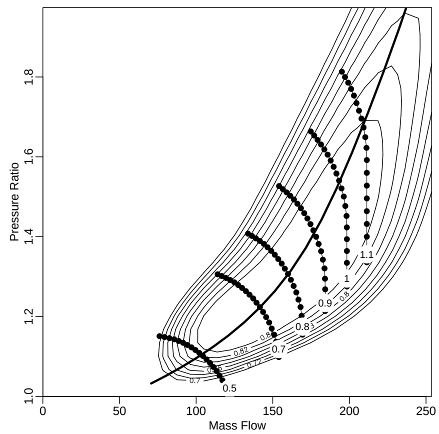

Fig. 2. Fan

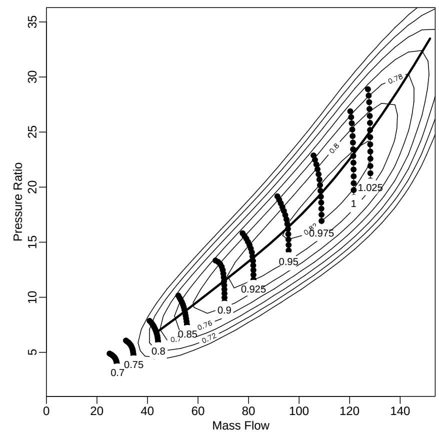

Fig. 3. High Pressure Compressor

range of speed values, as indicated by the values at the end of each speed line, is much greater in figure 2 than figure 3. Linear scaling does not account for these differences and research has continued to rectify these shortcomings. Kurzke and Riegler [7] analyzed a large number of compressor maps, to determine in what ways they are similar and in what ways they are different. They proposed an alternative non-linear scaling method.

The heart of this new scaling method is the ridge, or backbone, of the function, a line connecting points of maximum efficiency along each speed line. The axes are first normalized based on a reference point, the specifics of how it is identified can be found in the original paper [7]. The ridges on all the maps were found to be well approximated by an arc of a circle passing through coordinates (0,0) and (1,1). The radius of the circle depends on the pressure ratio at the reference point. Figure 4 shows an example of the scaling procedure in action. The bold solid black line is the ridge of the existing map; the bold dashed line is the ridge of the new one. The two adjacent thin lines are their approximation as arcs of a circle. First the range of each speed line in terms of mass flow is adjusted, according to the ratio of old and new values. The points on the ridge for each speed line are translated horizontally according to the different horizontal distances between the circles. The gray curves in figure 4 indicate one example of a translated and scaled speed line. New values are assigned to each speed line and the efficiency values rescaled, see the paper for details [7].

2.2 Morphological Scaling

The approach taken in this paper is to use morphological image analysis techniques [2] to identify the commonalities and differences between compressor maps. The method relies on identifying common points on all functions, called landmarks, and then finding the morphological operators needed to morph the points on one function to those on another.

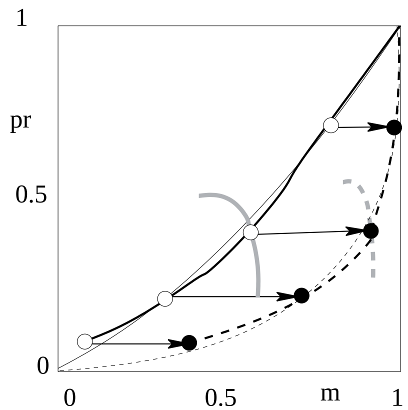

Fig. 4. Non-linear Scaling

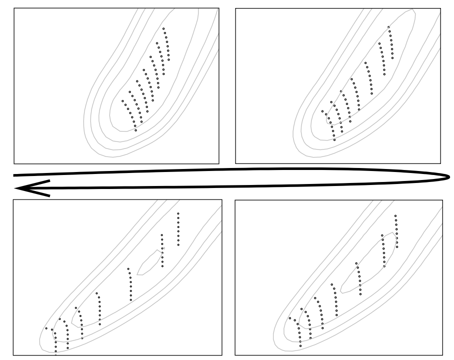

Fig. 5. Morphing between Maps

The speed lines are a central feature of the compressor maps. Following Kurzke and Riegler [7], a ridge is defined at the maximum efficiency point along these lines. Where the speed lines intersect with the ridge, an initial set of landmarks is defined. Unfortunately, the speed lines are not at the same compressor speeds for each map. To get a consistent set of landmarks interpolation and extrapolation are necessary. A method based on domain knowledge is best at minimizing error. This is particularly true for extrapolation, where many schemes can produce large deviations arising from small errors in values. A commonly accepted, and reasonably accurate, approximation of the relationship described by the speed lines is the cubic polynomial [3]. Polynomials are notoriously ill-suited to extrapolation and this one is only an approximation, albeit a very useful one. So, to incorporate this knowledge, while producing better behaved curves, splines are used. These are the linear combination of smooth functions, here b-spline basis functions. Instead of simply minimizing the squared error, a penalty function is included that penalizes large differentials [10].

Penalizing the fourth order differential means the smoothest possible function is a cubic polynomial. By requiring that the gradient of the function is zero when the mass flow is zero, we get a cubic of the required form. The process fits all the speed lines at the same time so that their common characteristics, defined by the form of the cubic, are included. New speed lines can then be generated at selected speeds. With these new speed lines and their intersection point with the ridge, 8 landmarks are generated along each of 7 speed lines. Splines have been used extensively in this work, with different differential functions to smooth the data in different ways. The ridge is located using an exponential penalty function which produces a curve that fits the ridge somewhat better than the circle discussed in the previous section. Though it is worth noting that for most maps the difference is small. A two dimensional cubic spline, with a third order differential penalty function, is used to generate the complete function. This gives a preference quadratics in two dimensions.

To extract the commonalities between maps,“relative warps” are used [2]. A prototypical map is produced by averaging the coordinates of each landmark separately for all maps. Morphing this will produce any other map in the space. A thin plate spline is used; it is easy to flex but bending it sharply is difficult. This encourages morphological operations that move the landmarks together in one direction and allows the smooth morphing from one map to another. This is controlled by the convex combination of coordinates in the shape space, converted back to the original coordinate system. Figure 5 shows the effect of morphing the maps shown earlier in figures 2 and 3. These were deliberately chosen to be very different. Yet, looking at figure 5, the arrow indicates the morphing direction, the intermediate functions certainly appear to be sensible compressor maps. So, we can produce a new map by averaging over the values of any number of existing maps. The only question is which maps to average over. Sensibly, the nearest neighbor, based on mass flow and pressure ratio, should be one of the maps. If we wish to interpolate, always safer than extrapolation, other points must be found that surround the desired reference point. By using Delaunay triangulation, three points are identified surrounding this point. Their shape space coordinates are averaged and the new map produced using the appropriate morphological operators.

3 Experiments

The experiment uses each of 19 maps as the target. Its aim is to determine which of the three scaling method best approximates the target, knowing only its pressure ratio and mass flow. The least squares distance between the values of pressure ratio, mass flow and efficiency at the landmark points, on the scaled and target maps, is used as the measure of fit. The pressure ratio and the mass flow are normalized, so that the units chosen have no influence on the results and the dimensions contribute in equal part. The nearest neighbor is chosen based on the Euclidean distance of the normalized coordinates. Figure 6 shows how the maps’ reference points vary in terms of pressure ratio and mass flow. The black circles are fans, they have very low pressure ratios and a large range of mass flow. The gray circles are low and high pressure compressors, they are designed for much greater pressure ratios but usually with smaller mass flows.

The results are shown in table 7. The best fit, the smallest squared error, is indicated by bold print. The performance using morphological scaling is in most cases the best. This form of scaling is less effective when no surrounding triangle is found. Originally extrapolation was used but this produces unrealistically shaped maps. Finding the nearest point on the black line in figure 6, the convex hull of all the maps (without the target map in the experiments) resulted in sensible maps and produced the results shown in the table. For instance, map 19, at the top, is not only outside the hull formed by the other maps, it is some distance away. Interpolation means averaging only maps 17 and 18. Here, the non-linear scaling is most effective, using human extracted knowledge. Similarly for map 9, on the far right, the nearest point on the convex hull is the nearest

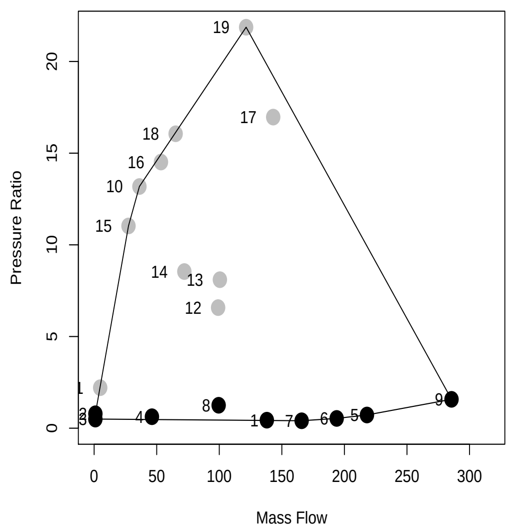

Fig. 6. Reference Points

Map	Linear	Non-linear	Morphed
1	0.774	**0.765**	0.808
2	0.880	**0.768**	0.856
3	0.880	1.329	**0.879**
4	1.160	**0.929**	0.969
5	0.109	0.096	**0.054**
6	0.109	0.226	**0.040**
7	0.774	0.805	**0.176**
8	4.188	3.229	**1.456**
9	0.236	**0.225**	0.236
10	**2.062**	2.416	2.522
11	1.051	2.569	**0.798**
12	2.791	**1.014**	1.346
13	**2.791**	2.828	3.057
14	4.177	5.051	**2.233**
15	3.274	3.676	**1.983**
16	0.989	1.174	**0.676**
17	1.996	1.769	**1.079**
18	2.931	5.152	**2.124**
19	1.986	**1.310**	2.177

Fig. 7. Mean Squared Error

neighbor itself, map 5, so the morphing error is identical. Again, non-linear scaling offers some improvement, through for this map it is rather small.

This method produces the best fit for compressor maps, further work is needed to see if the same is true for maps of other components. Maps for turbines are similar in many ways, so offer some promise. The same parameters, mass flow, pressure ratio and efficiency are used, although the shape is quite different. Other components also have maps, such as the combustion chamber. There are likely to be fewer such maps readily obtainable. But, with a few examples and greater use of domain knowledge, generalization should still be possible. Maps for several components of an engines would allow the effective incorporation of knowledge into learning algorithms used to diagnose and predict engine faults. Effectively modeling the normal performance of engines already in service would make identifying deviations associated with problems much easier.

The approach presented here should generalize beyond gas turbine engines and be applicable whenever functions are being learned. The general idea is using knowledge in support of learning functions. This is both theoretical, such as restrictions on the form of speed lines, and empirical, the set of maps for existing compressors. At a more detailed level, constraints in the form of penalty functions for splines, allow the input of theoretical knowledge without the overly strong constraints of a parametric model. The morphology approach does require the identification of landmarks. But if these can be found, the approach over a useful way of extracting the common characteristics of functions. In this author's earlier work [1], on transfer in reinforcement learning, previously learned solutions to subtasks were adapted to fit new ones. It would be worth revisiting this work to test the effectiveness of the approach taken here.

4 Conclusions

This paper introduced a way of learning multi-dimensional functions. Each functions is a point in shape space, produced by morphing a prototypical function located at its origin. The prototypical function and the space's dimensions, which define morphological operations, are learned from a set of existing functions. New ones are generated by averaging the coordinates of similar functions and using these to morph the prototype appropriately. The efficacy of this approach was experimentally demonstrated on a set of compressor maps, multi-dimensional functions relating various parameters of engine compressors.

References

1. C. Drummond. Accelerating reinforcement learning by composing solutions of automatically identified subtasks. *Journal of Artificial Intelligence Research*, 16:59–104, 2002.
2. I. Dryden and K. Mardia. *Statistical Shape Analysis*. John Wiley and Sons, Chichester, UK, 1998.
3. J. T. Gravdahl and O. Egeland. *Compressor Surge and Rotating Stall: Modeling and Control*. Springer Verlag, London, 1999.
4. A. D. Griffiths and D. G. Bridge. Formalising the knowledge content of case memory systems. In *Proceedings of the First UK Workshop on Progress in Case-Based Reasoning*, LNCS 1020, pages 32–41, 1995.
5. K. Hanney and M. T. Keane. The adaptation knowledge bottleneck: How to unblock it by learning from cases. In *Proceedings of the 2nd International Conference on CBR*, LNCS 1266, pages 359–370, 1997.
6. C. Kong, J. Ki, and M. Kang. A new scaling method for component maps of gas turbine using system identification. *Journal of Engineering for Gas Turbines and Power*, 125(4):979–985, 2003.
7. J. Kurzke and C. Riegler. A new compressor map scaling procedure for preliminary conceptional design of gas turbines. In *Proceedings of the ASME Turbo Expo*, 2000.
8. S. Létourneau, F. Famili, and S. Matwin. Data mining for prediction of aircraft component replacement. *IEEE Intelligent Systems Journal: Special Issue on Data Mining*, pages 59–66, 1999.
9. J. R. Quinlan. Combining instance-based and model-based learning. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 236–243, 1993.
10. J. O. Ramsay and B. W. Silverman. *Functional Data Analysis*. Springer-Verlag, New York, 1997.
11. J. F. Sellers and C. J. Daniele. DYNGEN-A program for calculating steady-state and transient performance of turbojet and turbofan engines. Technical Report NASA-TN D-7901, NASA, 1975.
12. W. Visser and M. Broomhead. GSP, a generic object-oriented gas turbine simulation environment. In *Proceedings of the ASME Turbo Expo*, pages 8–11, 2000.