Proceedings of the 1997 IEEE
International Conference on Robotics and Automation
Albuquerque, New Mexico - April 1997

Experimental Task Analysis

Jane Mulligan
Department of Computer Science
University of British Columbia
Vancouver, B.C. CANADA
mulligan@cs.ubc.ca

Alan K. Mackworth
Department of Computer Science
University of British Columbia
Vancouver, B.C. CANADA
mack@cs.ubc.ca

Abstract

Rigorous analysis and evaluation of real implemented robotic systems for intelligent tasks are rarely performed. Such systems are often extremely complicated, depending not only on 'interesting' theoretical parameters and models, but on many assumptions and constants which may be set almost arbitrarily. We view all task implementations as particular parameterizations of the task goals they represent. Through fractional factorial experiments we establish the statistically significant parameters and parameter interactions for a 'sensorless' model-based push-orienting task. This type of analysis is a necessary step to understanding integrated intelligent systems. It reveals aspects of system implementations which cannot easily be predicted in advance, and gives a clear picture of the task requirements, given the strengths and weaknesses of the observed system.

1 Introduction

For any task goal there are many potential robotic systems which could execute a plan to achieve it. These would employ a variety of actuators, information sources and computations. We would like to design task solutions which are optimal, but in order to evaluate and compare systems which employ diverse components and methods, we need a framework and criteria by which to measure them. In the absence of clear and detailed specifications for multifaceted robotic tasks, the process of elucidating task requirements will inevitably be experimental and cyclical. We therefore propose performance-based techniques for analysing the requirements of implemented robotic manipulation tasks.

There are many ways to represent a robotic task and most do not take into account the complexity and variety of the hardware and software elements required for actual implementation. The study of physical systems typically cycles through phases of model building, prediction, and experimental verification. Viewing robotic systems as parameterizations, with features, thresholds, constants, and constraints as the task level parameters of interest, gives us a unifying representation common to such diverse fields as economics, statistics, control, and inverse theory. It allows the use of well established methods for analysis and comparison.

Factorial experiments [2] are applied to systems where a number of parameters or factors may interact to affect system outcome. They are designed to give the researcher information on the effects and interactions of all factors simultaneously while limiting the number of experimental trials required. "Taguchi's method" is a version of such designs advocated for improving manufacturing processes [7]. The theory of experimental design also gives us analysis of variance techniques which allow us to determine whether effects observed as a result of setting parameters to various levels are statistically significant. Thus for a system which we hypothesize to have k possibly interacting parameters, we can run specifically designed experiments and use statistical tools to determine whether each factor or interaction has a significant effect on outcome. We can then devote our resources to optimizing those system parameters with significant impact on task performance.

2 The Part Orienting Task

To demonstrate our ideas about task parameterization and experimental analysis we have chosen the 2D part orienting task. Although it is generally quite simple (resolving only one degree of freedom of the part), it illustrates the variety in task performance that can be observed using empirical techniques. Push orienting was selected because, as a "sensorless" method, it simplifies some of the details involved in integrating sensor systems into robotic tasks.

Peshkin [6] describes a planning method for generating sequences of manipulations to transform a part from an unknown initial orientation to a desired final orientation. His work is based on the design of fence feeders, but can equally well apply to a robot arm holding a fence and applying similar pushing operations. We have implemented a version of the latter. The planner is based on a construct called the *configuration map*, a matrix which groups discretized initial orientations according to the final orientations which result after a push at a particular fence angle α. Final orientations are determined by the corner of the part contacting, then rotating to align with the fence. Rotation direction is determined by the friction between part and fence, and the centre of mass of the part [3]. For some discretization of the full range of fence angles, these maps are computed. Planning is achieved by a search of the tree of all push sequences with pruning. We have augmented this planner slightly by allowing transitions to more than one outcome, when model uncertainty makes the exact transition point from clockwise to counterclockwise rotation uncertain. Very recently more direct methods for computing fence angles for fence feeders have been proposed [8], but the system described here, although slower, is nonetheless effective.

3 Fractional Factorial Experiments

Like other statistical experiments, factorial experiments apply treatments in a structured manner to extract as much information about the effects and interactions of factors as possible. In the general case, for a complete factorial design we select some number of levels l_i for each of k factors and test all possible combinations of all levels of the various factors. Such an arrangement is called an $l_1 \times l_2 \times \ldots \times l_k$ factorial design. If all $l_1 = l_2 = \ldots = l_k = l$ then the design is referred to as an l^k symmetrical factorial experiment. Commonly 2^k experiments are used, where each of k factors are tested at two levels; we shall describe these in what follows.

We are interested in testing the effect of each factor individually, or its main effect, and the interactions among factors, or interaction effects. The main effect of a factor is computed from the set of responses observed under each treatment combination as follows,

$$\text{Main Effect} = \begin{pmatrix} \text{avg effect} \\ \text{while factor} \\ \text{is high} \end{pmatrix} - \begin{pmatrix} \text{avg effect} \\ \text{while factor} \\ \text{is low} \end{pmatrix}.$$

In other words an observation has coefficient 1 where the factor is high, or coefficient -1 if the factor is low. Coefficients for computing interaction effects can be generated by elementwise multiplication of the coefficients for the main effects of the factors involved in the interaction [1]. Treatment combinations are generally distinguished from computed effects by the use of lower case versus upper case factor labels.

The general formulation for such experiments for p treatments is

$$Y_i = \mu + \tau_i + e_i, i = 1, 2, \ldots p. \quad (1)$$

In other words the observation Y_i is composed of its underlying mean μ plus the treatment effects τ_i plus a normally distributed random error e_i. The null hypothesis for such experiments is that all τ_i are zero, implying that the mean for all treatment populations is the same. We can then use analysis of variance (ANOVA) techniques to accept or reject these hypotheses. Thus we begin with the null hypothesis that the various factors and interactions between them have no effect on observed responses. Referring to equation (1), the treatments τ_i are the various treatments proscribed by the factor levels, for example A_{hi} versus A_{lo} is one partition of the observed samples. The sums of squares for the contrasts used to estimate effects, have a χ^2 distribution with 1 dof for the 2^k case, we can therefore construct an ANOVA table for our factorial designs. Every 'effect' described above is actually the difference between average observations for two treatment levels spanning the entire set of experimental observations [7]. Using ANOVA we can apply the F test to determine whether our experiments provide sufficient evidence to reject the null hypothesis.

Fractional factorial experiments take advantage of the redundancy available in the set of observations to allow the experimenter to make fewer runs in the initial stages of his investigation. This is particularly important when the number of factors tested is large and hence the number of trials required by a complete design is huge. In this technique a subset of the factor/level combinations is used. Generally the selection of the subset is done such that higher order interactions are confounded or aliased to main effects or low order interactions, under the assumption that interactions among many factors are less likely to be significant [1]. In other words if you eliminate some of the treatment outcomes on which effect computations are based, some effects will inevitably become indistinguishable. The subset of observations is chosen based on an identity set of high order effects, which become completely confounded. For any choice of the identity relationship we can generate the resulting confounding relationships with all other effects via computing generalized interactions with the identity [2].

4 Experiments and Parameterizations

The premise of this investigation is that any robotic system which performs a task, implements an underlying physical 'computation'. We view such systems as parameterizations of this computation. Anyone who builds systems knows that they contain many parameters and thresholds which must be chosen by the designer: sometimes based on theory, sometimes via intuition and limited testing. All of these design choices affect the performance of the system. Below we enumerate one possible parameterization of the part orienting task represented by the push planner.

Eleven parameters or factors were selected as representative for the push planning system. These are described in brief with their assigned values in Table 1. The parameters F, C and V indicate the accuracy of the measured values of the friction coefficient, centre of mass and vertices respectively. D is the discretization factor for the fence angles and hence the branching factor for the push planner's tree search. In our coordinate system fence angles near 0 or π are essentially end-on to the part and are too steep to allow viable pushes. We therefore set a limit L on the set of fence angles, searching only from $(0 + L)$ to $(\pi - L)$.

Our experience with executing push plans has shown that steeper fence angles are less reliable because, with limited fence length and push distance, parts tend to "fall off" the end of the fence or wind up under the fence because they were left out of its range by the preceding fence angle. As a result we have added a constraint S to the push planner which prefers plans with shallower (closer to $\frac{\pi}{2}$) fence angles. We chose to evaluate the presence and absence of this constraint to determine its usefulness and its effects on other aspects of the planner.

Peshkin [5] describes methods for bounding the required push distance to align a part with a fence, based on the slowest possible centre of rotation. We have used his formulation to compute a bound on maximum push distance for each push in a plan. Minimizing push distance is another constraint P, we added because of the real physical limits of an implemented push planner. Plans with the shortest total push distance or shortest maximum push are preferred.

In terms of costs for an active push orienter, each additional push is expensive because it requires a motion sequence. We have therefore added a constraint N which prefers plans with fewer pushes.

Finally factors A, B, and K indicate whether the uncertainty compensation described in Section 2 is turned on for friction, centre of mass and vertex uncertainty respectively.

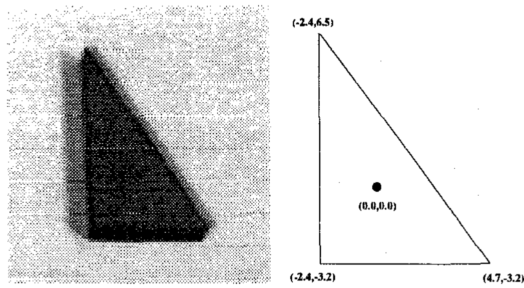

Figure 1: On the left is an image of the Triangular part, on the right its model.

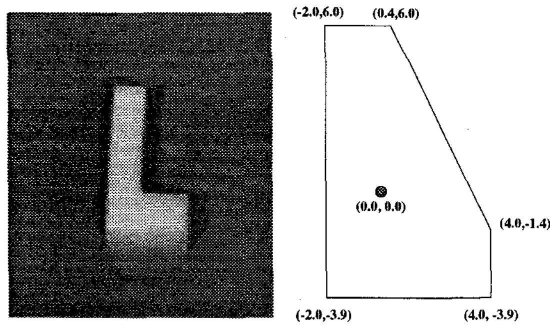

Figure 2: On the left is an image of the L-shaped part, on the right its model.

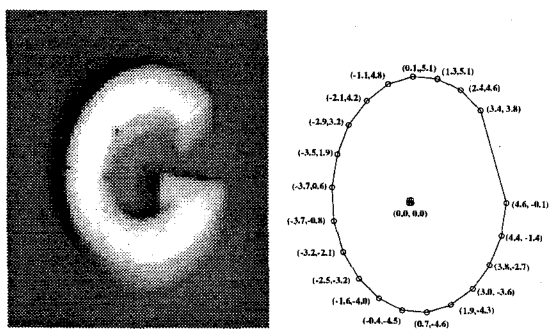

Figure 3: On the left is an image of the G-shaped part, on the right its 20 vertex model.

The experiments described in this paper were performed on the parts pictured in Figures 1, 2 and 3. For the push planner, these parts span a range of complexity. The triangle (T) has three stable sides. The "L" has two short sides which tend not to act as stable resting positions and the "G" has been modelled by a polygon with 20 vertices rather than a true curve.

4.1 The $\frac{1}{8}$ Fractional Factorial Design

Clearly we cannot run 2^{11} trials for the 11 parameters we have selected for the push planner. We

Parameter Levels			
Parameter	Label	High Value	Low Value
Measured Friction	F	T $\mathcal{N}(0.33, 0.04)$ L $\mathcal{N}(0.30, 0.03)$ G $\mathcal{N}(0.23, 0.015)$	0.33 0.30 0.23
Measured Centre of Mass	C	$\mathcal{N}(CofM, 0.25cm)$	Measured Value CofM
Measured Vertex Position	V	$\mathcal{N}(V_i, 0.1cm)$	Measured Position V_i
Discretization Factor	D	60	30
Fence Angle Limits	L	$[5, (180-5)]$ deg	$[30, (180-30)]$ deg
Fence Steepness Limit	S	TRUE	FALSE
Push Distance Limit	P	TRUE	FALSE
Plan Length Limit	N	TRUE	FALSE
Friction Uncert. Compensation	A	TRUE	FALSE
C of M Uncert. Compensation	B	TRUE	FALSE
Vertex Uncert. Compensation	K	TRUE	FALSE

Table 1: Parameter values for push planner. Qualitative factor values are denoted TRUE and FALSE indicating whether the property is active. $\mathcal{N}(\mu, \sigma)$ represents a value drawn from a normal distribution with the indicated parameters.

will therefore use a $\frac{1}{8}$ fractional factorial experiment with $\frac{2^{11}}{2^3} = 2^8$ treatments tested. The identity relation used to generate the fraction of treatments for the push planner is $I = CVDLSP = FVDLNA = FCSPNA = FCDLBK = FVSPBK = CVNABK = DLSPNABK$. The generated fractional factorial experiment requires 256 trials, but provides information on the main effects and all two-factor interactions, confounded with only three or higher order interactions.

Although we have a working system to execute push plans, we used a simple simulator (tested extensively against the real system's performance) because of time limitations and the sheer number of trials required. The simulator takes the part model and the selected plan (with associated maximum push distances) and outputs a success or fail based on the same *cw* or *ccw* rotation decisions used by the push planner. In addition, however, the simulator approximates the part's speed of alignment with the fence which means plans can fail because the part never becomes fully aligned. The part's slide along the fence after alignment is simulated and may cause it to "fall off" the limited extent of our fence. If some inaccuracies are assumed in the part features, a normally distributed perturbation is added to the appropriate vectors potentially causing "bad" rotations. Outcome proportions of success are based on simulating 1000 plan executions with uniformly distributed random initial part orientations.

5 Observed Outcomes

Tables 2 through 4 contain the analysis of variance computations for the proportion of success for the three parts under consideration. All main effect values are indicated but two factor interactions with F value less than $F_\alpha = 2.71$ for $\alpha = 0.1$, have been pooled.

5.1 Triangle Outcomes

ANOVA - Success Rate for Triangle					
Source		dof	SS	Mean Sqrs	F
	F	1	0.018	0.018	0.110
	C	1	0.531	0.531	3.249
	V	1	0.008	0.008	0.047
	D	1	0.203	0.202	1.239
	L	1	1.616	1.616	9.889
	S	1	0.479	0.479	2.931
	P	1	0.007	0.007	0.046
	N	1	0.218	0.212	1.295
	A	1	0.034	0.034	0.206
	B	1	0.017	0.017	0.102
	K	1	0.019	0.019	0.114
	DL	1	0.777	0.777	4.755
	LS	1	0.702	0.702	4.299
	LP	1	0.484	0.484	2.960
	SK	1	1.223	1.223	7.486
Error	ε	176	28.762	0.163	1.000
Total	T	255	35.091	214.729	1.000

Table 2: Analysis of Variance summary for proportion of success for triangle part.

For the triangle, Table 2 tells us that the significant effects are C, L, S, DL, LS, LP and SK. The most significant effect is the negative effect for angle limit L. In other words, the rate of success when L is low is greater than that when it is high. S also exhibits significant positive effect. Clearly our efforts to limit steep fence angles are successful for the triangle part. Given that uncertainty in centre of mass C significantly degrades results, we see that plans are sensitive to inaccuracy in this measured feature. Uncertainties in friction and vertex location show less significant effects, probably because variation in centre of mass is very much like perturbing all vertices simultaneously, thus resulting in greater impact on outcome.

The most significant interaction effect is SK which is negative. In this case vertex compensation K has a large negative effect when steepness minimization S is used and a strong positive effect when S is not used. We see significant interactions for L with D, P and S. For LP and LS the effects are positive. In the case of DL we see a negative effect supported by the negative effect of L and the insignificant, but also negative effect of D.

5.2 "L" Outcomes

ANOVA - Success Rate for "L"					
Source		dof	SS	Mean Sqr	F
	F	1	0.167	0.167	1.081
	C	1	0.445	0.445	2.886
	V	1	0.075	0.075	0.488
	D	1	1.011	1.011	6.551
	L	1	2.455	2.455	15.909
	S	1	0.271	0.271	1.756
	P	1	0.030	0.030	0.194
	N	1	0.442	0.443	2.868
	A	1	0.058	0.058	0.377
	B	1	0.039	0.039	0.252
	K	1	0.002	0.002	0.010
	DL	1	0.493	0.493	3.194
	LS	1	1.231	1.231	7.980
	LP	1	0.500	0.500	3.240
	SP	1	0.609	0.609	3.950
	SK	1	1.211	1.211	7.846
Error	ε	175	27.002	0.154	1.000
Total	T	255	36.041	233.584	1.000

Table 3: Analysis of Variance for success for "L" part.

Again we examine our tabulated ANOVA (Table 3) to determine the significant effects for the "L" part. The effect L is again the largest effect and is negative. The D effect has increased to a significant negative effect and the interaction of the two, DL, is negative.

Other significant effects include the negative effect of noise in the centre of mass, and the negative effect of N, minimizing the number of pushes in a plan. LS is a positive effect; the effect of P with L low was negative. Examining the negative effect for SK suggests a strong negative effect of K when S is high.

5.3 "G" Outcomes

ANOVA - Success Rate for "G"					
Source		dof	SS	Mean Sqrs	F
	F	1	0.286	0.286	4.244
	C	1	0.936	0.936	13.902
	V	1	4.855	4.855	72.102
	D	1	0.337	0.337	5.002
	L	1	0.153	0.153	2.277
	S	1	1.960	1.960	29.104
	P	1	0.010	0.010	0.152
	N	1	0.000	0.000	0.002
	A	1	0.003	0.003	0.045
	B	1	0.012	0.012	0.180
	K	1	0.034	0.034	0.511
	FV	1	0.216	0.216	3.203
	CV	1	0.985	0.985	14.635
	VL	1	0.184	0.184	2.728
	VS	1	1.029	1.029	15.278
	LS	1	0.403	0.403	5.981
	FP	1	0.285	0.285	4.226
	DK	1	0.233	0.233	3.454
	SK	1	0.378	0.378	5.612
	NK	1	0.196	0.196	2.912
Error	ε	171	11.514	0.067	1.000
Total	T	255	24.008	356.551	1.000

Table 4: Analysis of Variance summary for proportion of success for "G" part.

The most noticeable difference between the "G" and the other parts is the extreme sensitivity to noise in part features (Table 4). All of the noise factors F, C and V have significant negative effects and they interact with each other and with other factors in significant ways. The oddest effect is perhaps that noise in the friction measurement is aggravated by minimizing push distance (interaction FP). Perhaps shorter push distances imply steeper fence angles and these are more likely to depend critically on friction vectors.

The strongest non-noise effect is steepness minimization S which is positive. D again is a significant negative effect and L, while not above the significance test, is also negative. The positive interaction VL merely tells us that L has a negative effect for the low noise condition. The negative effect of VS indicates S's positive effect at low noise. For LS, detailed examination shows that the effect of S is positive for both high and low levels of L. Although K has a small positive effect at low N and D settings, it imposes a negative effect for the more significant high S.

5.4 Observations

For the three parts studied (the triangle, "G" and "L") the push orienting system demonstrated different behaviours and different optimal parameter settings. This is true even for this very basic manipulation operation of orienting a part. Most of the information relevant to orienting the triangle appears to be contained in parameters S, L and V. For the "L" part the most significant parameters are centre of mass variation C, fence angle discretization D, the limit on steep angles L and minimization of plan length N. The "G" fared badly under push orienting with extreme sensitivity to the noise parameters F, C, and V. The relevant information for this part and orienting method is captured by F, C, V, D and S.

For the push planner this analysis clearly indicates the increasing complexity of the parts with respect to push orienting. Although, for example, the triangle and the "L" would appear very similar under the model used, analysis demonstrates that the performance of the system for each part and the parameters which exhibit significance are quite different. Clearly the individual part information and its inherent complexity play a major role in task definition.

6 Summary and Conclusions

In this paper we have advocated treating task implementations as task parameterizations which can be explored experimentally. We have described a performance-based experimental method for analysing robotic task implementations. We demonstrated the usefulness of treating robotic systems as complex parameterizations of tasks, then employing factorial experiments to identify significant parameters and interactions among parameters for each of 3 parts (a triangle, an "L" shape and a "G" shape) for a push orienting task. Perhaps the most surprising results were the differences between the sets of parameters which proved most significant for each part.

The most significant lesson learned from this analysis is that extensive analysis of actual robotic implementations is a necessary step to understanding such systems and the tasks they perform. This type of analysis reveals aspects of the implementation which cannot be predicted *a priori*. It also clearly defines what information is necessary for a particular task implementation, and what the strengths and weaknesses of these implementations are.

In many ways the sensorless push orienting system examined here is one of the simplest we could choose. We have also applied these methods to more complicated sensing-action systems, including an integrated vision and manipulation system, as part of a larger methodology for analysis and comparison of robot manipulation tasks [4].

Experimental systems in vision and robotics have rarely been rigorously tested and documented. An important conclusion of this work is that it is vacuous to propose a robot solution without a clear analysis of its system parameters and how they interact with each other, and with task success. This is especially true as the task becomes more complicated. In our analysis we have clearly demonstrated that determining key task parameters and their effect on performance, is critical to providing and understanding robust task solutions. These results point the way toward a theory of practical and economic application of robotics in manufacturing and other industries.

References

[1] Aloke Dey. *Orthogonal Fractional Factorial Designs.* Halsted Press, John Wiley and Sons, New York, NY, 1985.

[2] Oscar Kempthorne. *The Design and Analysis of Experiments.* John Wiley and Sons, New York, N.Y., 1952.

[3] Matthew T. Mason. Mechanics and Planning of Manipulator Pushing Operations. *International Journal of Robotics Research*, 5(3):52–71, 1986.

[4] Jane Mulligan. *Empirical Evaluation of Information for Robotic Manipulation Tasks.* Computer science, University of British Columbia, Vancouver, B.C., August 1996.

[5] Michael A. Peshkin and Arthur C. Sanderson. The motion of a pushed, sliding workpiece. *IEEE Journal of Robotics and Automation*, 4(6):569–598, 1988.

[6] Michael A. Peshkin and Arthur C. Sanderson. Planning robotic manipulation strategies for workpieces that slide. *IEEE Journal of Robotics and Automation*, 4(5):524–531, 1988.

[7] Ranjit K. Roy. *A Primer on the Taguchi Method.* Competitive Manufacturing Series. Van Nostrand Reinhold, New York, NY, 1990.

[8] Jeff Wiegley, Ken Goldberg, Mike Peshkin, and Mike Brokowski. A complete algorithm for designing passive fences to orient parts. In *ICRA '96*, volume 2, pages 1133–1139, Minneapolis, Minnesota, April 1996.