Online available since 2012/Nov/12 at www.scientific.net
© (2012) Trans Tech Publications, Switzerland
doi:10.4028/www.scientific.net/AMR.590.399

PRECISION POINT DESIGN OF A CAM INDEXING MECHANISM

Alberto Borboni[1,a], Francesco Aggogeri[1,b], Nicola Pellegrini[1,c], Rodolfo Faglia[1,d]

[1]Università degli Studi di Brescia

Dipartimento di Ingegneria Meccanica e Industriale

Via Branze, 38 - 25123 Brescia Italy

[a]alberto.borboni@ing.unibs.it, [b]francesco.aggogeri@ing.unibs.it, [c]nicola.pellegrini@ing.unibs.it, [d]rodolfo.faglia@ing.unibs.it

Keywords: cam indexing mechanism, motion law, precision point, genetic algorithm.

Abstract. This work regards on the design of a cylindrical cam indexing mechanism with a motion law that passes through a positional precision point. A numerical algorithm is proposed to solve this problem and, particularly, a genetic algorithm. The algorithm and the encoding of the problem are described.

Introduction

A cam indexing mechanism is composed by a moving element (the cam) and by a rotating plate (turret), that carries a series of rollers; these realize a shape coupling in every instant with the cam [8]. According to the type of adopted cam [8], the most common cam indexing mechanisms are classified as: cylindrical, globoidal and planar. This paper examines only cylindrical cam indexing mechanisms and we want to focus our attention on fast mechanisms, i.e., where inertia actions are important. In these conditions, the system exhibits phenomena associated to compliance of mechanical components thus it must be designed accurately and its mechanical behavior can be properly describe with a two degrees model [1, 15].

Different requirements must be met while improving the quality of the system: reduce the dimension of the cylindrical cam; reduce inertia actions; reduce the motor size; reducing vibrations.

Besides these common needs in the design of fast cam mechanisms, the movement of our indexing mechanisms must be coordinated with the movements of other machines that are parts of a more complex system, thus the turret can necessitate to pass through a precision point. This design problem is widely known in literature, because it is present in different industrial machines and it is commonly solved with cam law blending techniques [12]. A positive tolerance (also when it is small) is practically acceptable for this precision point, while other just mentioned requirements must be met in the indexing mechanism design [16], therefore this work investigates an efficient method for the numerical solution of this problem avoiding just known cam law blending techniques.

More in detail, the core of indexing mechanism design is the definition of the motion law (ML) associated to the cam profile mixing properly different functional requirements of the system. This activity can be performed approximately using proper coefficients [8] or precisely with iterative algorithms that optimize the ML to respecting imposed constraints [1].

Different optimization algorithms are present in literature, they are described in the second paragraph of this work and particularly we will chose genetic algorithms [6], due to their interesting characteristics.

Finally, different mathematical forms are shown in literature to describe ML of double-dwell fast cams, and some of them are [9]: cycloidal, modified sine, trapezoidal acceleration, modified trapezoidal acceleration (mML), polynomial (pML), trigonometric (tML), splines.

The attention will be focused on three classes of ML: pML, mML and tML. They will be studied to evaluate their suitability to solve the precision point problem with the chosen optimization algorithm.

Definition of the problem

We consider a cylindrical cam indexing mechanism, as shown in Fig. 1, where the follower consists of a turret (follower wheel) with a number of rollers spaced around a pitch circle, while the cam is properly shaped on its lateral cylindrical surface.

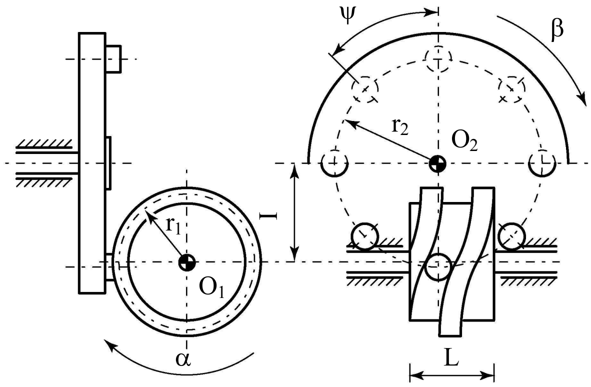

Figure 1 - Cylindrical Cam Indexer.

As shown in Fig. 1, the centers of the rollers are symmetrically placed on a circumference with radius r_2, and its value is approximately equal to the center-to-centre I between mover and follower. The mechanism is single pass with a ψ index period equal to the angular distance between two subsequent rollers, as described in (1), where n is the number of rollers.

$$\psi = 2\pi / n \quad (1)$$

To study the cam in detail, its lateral surface is fictitiously unrolled as shown in Fig. 2. Thus, if r_1 is the median radius of the cam, the obtained strip has a length $2\pi r_1$ and a width L. On this strip the segment $\alpha_a r_1$ corresponds to the motion angle of the follower and the segment $(2\pi-\alpha_a)r_1$ corresponds to its dwell.

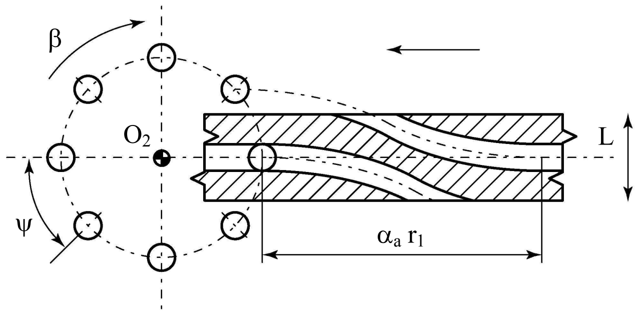

Figure 2 - Unrolled cylindrical cam.

The obtained mechanism is a translating mover that sets the follower in motion. The only difference with a usual swinging follower mechanism is that the follower rollers do not return to their starting position at the end of the motion cycle but they continue in the same direction and are replaced by the subsequent rollers.

Then, the design of the system demands a proper definition of the ML, that consists in a sequence of rise and dwell phases. We focus the attention on a single rise phase and the choice of the ML. The ML besides allows a correct rise, must also pass through a precision point for coordinating reasons between the turret of the indexing mechanism and other part of the entire machine, that is extremely compact, thus interferences and impacts must be avoided between the turret and other moving elements of the entire machine.

Such a motion may be easily designed using the cam law blending technique [12]. The term "blending" is used to describe the use of contiguous periods of motion to an adjacent different type of motion. In this way, two different ML are designed according to a desired dynamical behavior of the system and, then, a local modification is realized on the rise phase due to the constraint associated with the precision point.

A different approach consists in the use of a ML in series, i.e. a pML, or in sequence, i.e. splines [9] with same free parameters, thus they can be set analytically to allow the passage through a precision point before the implementation of other dynamical demands for the systems, i.e.: smooth movements, reduced vibrations, limited accelerations, reduced motor torque.

In both approaches, the exact crossing through a precision point is imposed to the detriment of other important dynamical demands of the system and this choice cannot be easily altered. Anyway, in practice, often the passage through a precision point is not an exact demand, but an error can be tolerated and, in some situations, this error can be rather wide. For this reason, we want to investigate a different method to realize the passage through a precision point with a positive tolerance error. This result can be achieved with numerical optimization methods and, particularly, with numerical minimization methods, that allows to minimize a general mathematical function. In our case, this function to be minimized can be the distance d between the ML and the precision point measured as the smallest segment connecting the precision point with a point on the ML (Fig. 3). Then we are interested in minimizing this distance to an acceptable tolerance level that is a value grater than zero.

Proposed solution method

The next problem is the choice of an optimization algorithm sufficiently adequate to our situation. Some classes of algorithms are studied to describe their pros and cons according to knowledge in literature [10] and to our experiences done on different problems [1, 3, 4, 14], collecting a global judgment in Tab. 1, where LO, GO, S, M and CO stays respectively for local optimum, global optimum, speed of the algorithm, memory consumed, complexity of the objective.

Table 1. Comparing different search algorithms.

Algorithm	LO	GO	S	M	CO
Gradient Based	+++	---	++	-	---
Hessian Based	+++	---	+++	--	---
Random Search	--	+++	---	+	+++
Stochastic Hill climbing	+	++	+	-	+
Simulated Annealing	++	++	+	-	++
Symbolic Artificial Intelligence	+	++~	-	-	+++
Genetic Algorithms	++	+++	-	+	+++

In our specific situation, we are interested to a global optimum problem, thus Random Search, Genetic Algorithms (GA) and also Symbolic Artificial Intelligence (AI) can be chosen. Symbolic Ai is discarded because we want to easily modify the algorithm for future works and Symbolic AI is very problem-specific, therefore this class of algorithms forces to restart the each time the activity. We do not pretend a fast computation, because this is not a real-time problem, but at the expense of a slight complexity, the GA allow much more computational speed than the Random Search, thus we chose the GA. This choice is common in literature and produced positive results [1, 4, 7 , 11-13].

The GA are iterative stochastic processes that operate on a family of potential solutions (called population) to alterate it and to pick up the global optimum of the problem to be solved. A wide literature is present on this subject [Goldberg, Back], where a complete treatment of GA can be found, whereas in the followings only basic concepts are hinted to properly describe the proposed GA and the exact encoding of the problem.

To solve different optimization problems, among which there is the passage through a precision point, a sufficiently general GA is implemented using different types of ML, by means of object oriented and virtual classes techniques. With this approach, different encoding options are set after that the problem to be solved and the type of ML are set. A general ML is considered as shown in (2), where x and y are the real spatial variables respectively at the input and at the output of the indexer, f() is the type of ML, a_i is the i-th of n parameters that the designer can adjust with a value belonging to the set A (that can be real, natural, integer or a different set of values according to the type of ML).

$$y = f(x; a_1, a_2, \ldots, a_i, a_{i+1}, \ldots a_n); \quad a_i \in A, \quad x \in \mathbb{R}, \quad y \in \mathbb{R} \tag{2}$$

It is thus evident that, chosen a type of ML f ( ), the ML can be individuate with a vector of parameters a, as shown in (3).

$$\mathbf{a} = < a_1, a_2, \ldots, a_i, a_{i+1}, \ldots a_n >; \quad a_i \in A \tag{3}$$

The implemented GA generate randomly m ML $\mathbf{a}_j$ producing an initial population $\mathbf{P}^0$ (4) of n potential solutions to a determined problem.

$$\mathbf{P}^0 = < \mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_j, \mathbf{a}_{j+1}, \ldots \mathbf{a}_m > \tag{4}$$

The term "initial" associated to the superscript 0 of the symbol $\mathbf{P}^0$ implies that the GA ha an iterative nature, thus the superscript k associated to a generic generation $\mathbf{P}^k$ (5) individuates the k-th step of the algorithms that produced it.

$$\mathbf{P}^k = < \mathbf{a}_1^k, \mathbf{a}_2^k, \ldots, \mathbf{a}_j^k, \mathbf{a}_{j+1}^k, \ldots \mathbf{a}_m^k > \tag{5}$$

Therefore, in analogy with evolutive processes, the population $\mathbf{P}^k$ is also called k-th generation of the GA, the element $\mathbf{a}_j^k$ of the population $\mathbf{P}^k$ is also called j-th individual of the k-th generation, while the I-th element of the j-th individual is called gene.

After the definition of a specific problem, the k-th generation is ordered according to a proper quality indicator that measures the fitness of an individual $\mathbf{a}_j^k$ to the problem to be solved. Thus the population $\mathbf{P}^k$ is an ordered vector in decreasing order of fitness to the problem. The sorting is realized either with a bubble sort algorithm for populations with a reduced number of individuals or with a quick sort algorithm for populations with a wide number of individuals. The generation $\mathbf{P}^{k+1}$ is produced from the generation $\mathbf{P}^k$, with proper operators that act in the following sequence:

1 – coupling,
2 – heredity,
3 – cross-over,
4 – mutation.

The coupling consists in picking up two individuals ($\mathbf{a}_{j1}^k$,$\mathbf{a}_{j2}^k$) on which the other three operators will be applied to generate a single individual $\mathbf{a}_j^{k+1}$. The couple of individuals ($\mathbf{a}_{j1}^k$,$\mathbf{a}_{j2}^k$) is chosen according to (6).

$$\forall j1 \in \{1, 2, \ldots, j, j+1, \ldots m\}, j2 = random(1, j1) \in \mathbb{N} \tag{6}$$

Thus the j1-th element is chosen scanning the population of individuals and each individual assumes at least one time the j1-th role, while the j2-th element is chosen randomly among a group of individuals between the first and the j1-th. Because the vector $\mathbf{P}^k$ is sorted with a decreasing fitness to the problem, the best individuals have a higher coupling probability. Particularly, the first individual is coupled at least one time with itself and the population $\mathbf{P}^{k+1}$ has the same number of individuals of the population $\mathbf{P}^k$. If two individuals in the couple have the same number of genes, the descendant individual has generally the same number of genes of the parents, but it is also provided the possibility of incrementing or decrementing randomly the number of genes of the descendant. When the two parents have different numbers of genes, the parent with a higher number of genes is divided (randomly or not) in two parts. The part with the same number of genes of the other parent is coupled as just described, while the remaining part is coupled with itself or with a sub-individual randomly generated; thus the descendant has generally the same number of genes of the parent with a higher number of genes.

Once defined the couple of parents and the number of genes of the unique descendant, the subsequent operators can be applied. The heredity consists in the random selection of a parent of the couple and it donates a single gene (among the undonated genes), then the gene is copied directly in the descendant in the same vector position assumed in the selected parent. This process is repeated e times, equal to the number e of inherited genes.

Then, the cross-over operator is applied. It is realized choosing randomly a parent and extracting from it a gene among those not inherited and not crossed-over. This gene is copied in the descendant in a random position (not necessarily the same vector position assumed in the selected parent). This process is repeated r times, equal to the number r of crossed-over genes.

Finally the mutation operator is applied: selecting randomly the donating parent and again randomly a gene of this parent among those not inherited, not crossed-over and not mutated. Then this gene is varied randomly around the value assumed in the parent individual and the descendant inherits it. The process is repeated u times, equal to the number u of mutated genes.

The application of these three operators is better explained with the example in Fig. 4.

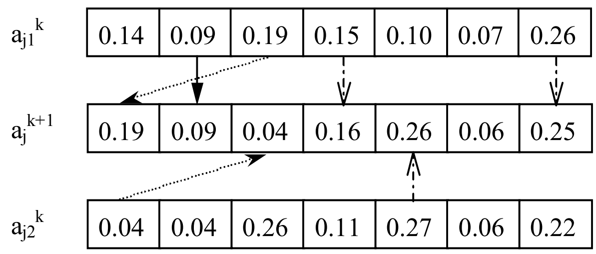

Fig. 4. Example of individual production

The individual $\mathbf{a}_j^{k+1}$ generated in this way can sometimes be not belonging to the set A. For example, if A is the set of real numbers respecting (7), it can happen that the GA generates a real number that do not respect (7), thus it is necessary to put $\mathbf{a}_j^{k+1}$ into the set A, through a proper transformation technique as proposed in (8).

$$\sum_{i=1}^{n} a_{j,i}^{k+1} = 1 \tag{7}$$

$$a'^{k+1}_{j,i} = a_{j,i}^{k+1} \Big/ \sum_{h=1}^{n} a_{j,h}^{k+1}, \quad i = 1, 2, \ldots n \tag{8}$$

Conclusions

The problem of the passage through a precision point was studied for cylindrical cam indexers with an iterative automatic numerical approach providing for the possibility, in the next future, to solve more complex problems. A class of algorithms was individuated that is adapt to the demand of our problem and these are the genetic algorithms, an encoding was proposed for these algorithms and the precision point problem was solved for different classes of motion laws. Numerical simulations must be performed to obtain further results.

References

[1] Antonini , M., Borboni, A., Bussola, R., and Faglia, R., 2006, "Automatic Procedures as Help for Optimal Cam Design," ASME Conference on Engineering Systems Design and Analysis, Torino, Italy.

[2] Back, T., 1996, "Evolutionary algorithms in Theory and Practice," Oxford University Press.

[3] Amici, C., Borboni, A., Faglia, R., A compliant PKM mesomanipulator: Kinematic and dynamic analyses, Advances in Mechanical Engineering (2010) art. no. 706023

[4] Borboni, A., Bussola, R., Faglia, R., Magnani, P., and Menegolo, A., 2008, "Movement optimization of a redundant serial robot for high-quality pipe cutting," Journal of Mechanical Design, 130(8).

[5] Fung, E., Hau, L., and Yau, D., 2004, "Multiobjective genetic algorithm approach to the optimization of a rotating flexible arm with acld patch," ASME International Mechanical Engineering Congress and Exposition, 1, pp. 1–8.

[6] Goldberg, D., 1989, "Genetic Algorithms in Search Optimization and Machine Learning," AddisonWesley.

[7] Hacker, K., and Lewis, K., 2002, "Robust design through the use of a hybrid genetic algorithm," ASME International Design Engineering Technical Conference, 2, pp. 703–712.

[8] Magnani, P. L., Ruggieri, G., 1986, "Meccanismi per Macchine Automatiche," UTET.

[9] Norton, R. L., 2002, "Cam Design and Manufacturing Handbook," Industrial Press Inc.

[10] Press, W. H., Teukolsky, S. A., Vetterling, W. T., Flannery, B. P., 2007, "Numerical Recipes: The Art of Scientific Computing. Third Edition," Cambridge University Press.

[11] Rao, S., and Xiond, Y., 2005, "A hybrid genetic algorithm for mixed-discrete design optimisation," Journal of Mechanical Design, 127, pp. 1100–1112.

[12] Reeve, J., 1995, "Cams for Industry," Mechanical Engineering Publications Limited.

[13] Renner, G., and Ekar, A., 2003, "Genetic algorithms in computer aided design," Computer-Aided Design, 35, pp. 709–726.

[14] Antonini, M., Borboni, A., Bussola, R., Faglia, R., 2006, "A genetic algorithm as support in the movement optimisation of a redundant serial robot," Proceedings of 8th Biennial ASME Conference on Engineering Systems Design and Analysis, ESDA2006 2006.

[15] F. Aggogeri, E. Gentili. Six Sigma methodology: An effective tool for quality management. International Journal of Manufacturing Technology and Management. Volume 14, Issue 3-4, (2008) 289-298

[16] E. Gentili, F. Aggogeri, M. Mazzola, The effectiveness of the quality function deployment in managing manufacturing and transactional processes, ASME International Mechanical Engineering Congress and Exposition, Proceedings 3 (2008) 237-246.

Mechatronic Systems and Materials Application
10.4028/www.scientific.net/AMR.590

Precision Point Design of a Cam Indexing Mechanism
10.4028/www.scientific.net/AMR.590.399