A SOLVING MODULE OF AN EXPERT SYSTEM FOR NONLINEAR BEAM DYNAMICS

S. Andrianov, A. Dvoeglazov, SPbSU, Bibliotechnaja pl.2, 198904 St.Petersburg, Russia

Abstract

In this paper we describe basic concepts which are took as a base of an expert system for an investigation of nonlinear behavior of particle beams. Modelling process for beam line design is based on the matrix formalism for Lie algebraic tools and computer algebra methods. This allows to use an object-oriented approach for describing both physical and mathematical objects. [1]

1 INTRODUCTION

It is known that human problem solving is much characterized by its enormous range as by the huge amounts of detailed specific knowledge that can be effectively mobilized for a particular domain. For scientific modelling problems one comes up tedious and time consuming calculations that are a source of numerous errors. That is why it is necessary to rearrange the main amount of work to be done from a researcher to computer. This aim can be reached in different ways. We consider one of these based on symbolic representation of necessary information. This approach allows to create databases and corresponding environment and use them if necessary [1]. The most proper tools for this is the matrix formalism for Lie algebraic methods [2].

2 THE BASIC MATHEMATICAL DEFINITIONS

It is known that any dynamical system can be described with the help of a Lie transformation (map) $\mathcal{M}(U;t|t_0)$:

$$X = X(X_0, U; t|t_0) = \mathcal{M}(U;t|t_0) \circ X_0,$$

where X and X_0 are current and initial phase vectors, U is a control vector, t is an independent variable (for example, the length along a reference orbit). For the general case of nonautonomous systems we can write

$$\mathcal{M}(F|U;t|t_0) = \mathrm{T}\exp\left\{\int\limits_{t_0}^{t} \mathcal{L}_{F(X,U,\tau)} d\tau\right\}, \qquad (1)$$

where $\mathrm{T}\exp$ is a chronological exponent operator, $\mathcal{L}_F$ is a Lie operator, associated with some function (for example, a right part of a ODEs system) $F(X,U;t)$. From (1) we can proceed to the Magnus representation in the form of a routine exponential operator for a new Lie operator:

$$\mathcal{M}\left(G|U;t|t_0\right) = \exp\left\{\mathcal{L}_{G(X,U;t|t_0)}\right\},$$

where the function $G(X,U;t|t_0)$ can be calculated with the help of a continuous analog of the well known CBH formula. Thus one can build the following chain of mappings:

$$F(X,U;t) \Longrightarrow \mathcal{L}_{F(X,U;t)} \Longrightarrow$$

$$\Longrightarrow \mathcal{M}(F|U;t|t_0) = \mathrm{T}\exp\left\{\int\limits_{t_0}^{t} \mathcal{L}_{F(X,U;\tau)} d\tau\right\} \Longrightarrow$$

$$\Longrightarrow \mathcal{M}(G|U;t|t_0) = \exp\left\{\mathcal{L}_{G(X,U;t|t_0)}\right\}. \qquad (2)$$

Let $F(X,U;t)$ be represented in the form

$$F(X,U;t) = \sum_{k=0}^{\infty} \mathbf{P}^{1k}\left(U;t\right) X^{[k]}, \qquad (3)$$

where $X^{[k]} = \underbrace{X \otimes \ldots \otimes X}_{k-times}$ is the Kronecker power of X of k–th order and $\mathbf{P}^{1k}(U;t)$ are coefficient matrices of the dimensions $\left(n \times \binom{n+k-1}{k}\right)$.

The mapping (2) and the expansion (3) generate the next mapping

$$\mathbf{P}^{1k}(U;t) \Longrightarrow \mathbf{G}_k(U;t|t_0),$$

where $\mathbf{G}_k(U;t|t_0)$ is a coefficient matrix for homogeneous polynomial of k-th order $G_k(X,U;t|t_0)$ and

$$\mathcal{L}_G = \sum_{k=0}^{\infty} \mathcal{L}_k, \quad \mathcal{L}_k = \mathcal{L}_{G_k}, \quad G_k = \mathbf{G}^k X^{[k]}.$$

The matrices $\mathbf{P}^{1k}$ (<u>system matrices</u>) are defined from initial physical model. The matrices $\mathbf{G}_k$ can be called system matrices too. Their form and the order N, $k \leq \mathrm{N}$ depend on an approximating model, which is chosen by a researcher as the series in (3) must be truncated . In this case we proceed to so called N–jet approach. After this it is convenient to use the Dragt–Finn factorized representation [3]

$$\mathcal{M} = \ldots \circ \mathcal{M}_k \circ \ldots \circ \mathcal{M}_1 = \tilde{\mathcal{M}}_1 \circ \ldots \circ \tilde{\mathcal{M}}_k \circ \ldots, \quad (4)$$

where $\mathcal{M}_k$ ($\tilde{\mathcal{M}}_k$) is a new map associated with a new Lie operator $\mathcal{L}_k$ ($\tilde{\mathcal{L}}_k$), $\mathcal{L}_k = \mathcal{L}_{H_k(X,t)}$, $H_k(X,t) = \mathbf{H}_k X^{[k]}$. According the matrix representation for Lie transformation we can write

$$\mathcal{M}\left(F|U;t|t_0\right) \circ X_0 = \sum_{k=0}^{\infty} \mathbf{M}^{1k}\left(F|U;t|t_0\right) X_0^{[k]}.$$

Matrices $\mathbf{M}^{1k}\left(F|U;t|t_0\right)$ are called <u>solving matrices</u>. So we can give the following definition.

[1] This work is supported by the Russian Foundation for Fundamental Researches 96-02-17335

Definition 1 . *Let $\mathcal{M}(F|U;t|t_0)$ be a Lie transformation generated by a dynamical system with the vector field $\mathcal{F}(X,U,t) = \mathcal{L}_{F(X,U,t)}$. The creation of the chain*

$$\begin{bmatrix} \textit{coordinates systems} \\ + \\ \textit{approximating} \\ \textit{external control fields} \end{bmatrix} \Longrightarrow \begin{bmatrix} (X,U) \\ + \\ \textit{an independent} \\ \textit{variable – "time" } t \end{bmatrix}$$

$$\Longrightarrow \left[\mathbf{P}^{1k},\ k \le N\right] \Longrightarrow \left[\mathbf{G}_k,\ k \le N\right]$$
$$\Longrightarrow \left[\mathbf{H}_j,\ j \le N\right] \Longrightarrow \left[\mathbf{M}^{1j},\ j \le N\right]$$

will be named solution process in the matrix formalism.

For $\mathbf{M}^{1k}$ we used symbolic formulae which have only algebraic character (see [2]). As a starting–point we can consider both motion equations in the form of ODEs and Hamiltonian description. For N–jet representations we have

$$\mathcal{M}\left(F|U;t|t_0\right) \Longrightarrow \mathcal{M}_{\mathrm{N}}\left(F|U;t|t_0\right):$$
$$\mathcal{M}_{\mathrm{N}}\left(F|U;t|t_0\right) \circ X_0 = \sum_{k=0}^{\mathrm{N}} \mathbf{M}^{1k}\left(F|U;t|t_0\right) X_0^{[k]}.$$

So our main goal is to calculate $\mathcal{M}_{\mathrm{N}}\left(F|U;t|t_0\right)$ (in the terms of $\mathbf{M}^{1k}\left(F|U;t|t_0\right)$) for given set of $\mathbf{P}^{1k}$, $k \le \mathrm{N}$ which describes a designed beam line.

3 SOLVING MODULE DESCRIPTION

3.1 Structure of a Solving Module

Symbolic character of all objects: from $\mathbf{P}^{1k}$ up to $\mathbf{M}^{1k}$ allows us to create databases of these matrices and corresponding database management system. Besides these matrices we must create rules according which one has to manipulate by these objects for his goal. This set of rules is the basic contents of proposed solving module. As above mentioned the necessary operations have mainly algebraic character. These operations we can separate on several submodules:

- Submodule of algebraic manipulations over noncommutative variables, which are necessary for evaluation of similar to CBH expansions and so on [4].
- Submodule of matrix algebra extended by including the Kronecker sum and product. All operations are made for abstract forms of matrices [5].
- Submodule for integration procedures for calculation of $\mathbf{G}_j$ and $\mathbf{H}_j$, $j \le \mathrm{N}$ (for example, for the Magnus representation) and other necessary integration procedures. For this purpose it is convenient to create a database of formulae for some set of support functions [2].
- Submodule for the factorization procedures (see (4)).
- Submodule for formation of the solving matrices $\mathbf{M}^{1k}$ for a selected initial variant of the beam line [8].
- Submodule for finding of explicit solutions for some classes of the Lie transformations [7].
- Submodule of manipulations by system matrices $\mathbf{P}^{1k}$ for an investigation of symmetry properties of the dynamical system under study (the level of building of a protoproject) and so on.
- Submodule for calculation of the invariants and symmetries for the designed beam line (the level of formation of beams with desired characteristics) [6].
- Submodule of calculation of envelope beam matrices (σ – matrices) and distribution functions in phase and/or configuration spaces [8].
- Submodule for maps construction including space charge [9].
- Submodule of calculation of object functions and condition functions for an optimization procedure. Here the optimal control theory and nonlinear programming methods are used .
- Submodule for numerical calculations for a selected set of system parameters – **dynamics level** and numerical optimization – **optimization level**.
- Submodule (if it is necessary and possible) for testing of modelling with the help of known packages (for example, such as $MARYLIE$, $COSY$, MAD) and/or using general mathematical packages (for solving, for example, ODEs) [10].
- Submodule for visualization of calculation results.

3.2 A Hierarchy of Submodules

For the effective working of all submodules we must create databases of ready objects. The symbolic forms of these objects allow to do it and a researcher can fill up these databases simultaneously with accumulation of his knowledge.

The submodules are combined into a solving module which is surrounded by a human interface. This interface can be provided by a special language for a restricted problem class or by more wide language which can be adapted under transfer from one problem to another. In this case we can talk about an object–oriented interface. For realization of the modelling process one must go through the following steps:

- to select transport system elements in desired N–order approximation (including all effects which are necessary) — to define the system 0matrices $\mathbf{P}^{1k}$, $k \le \mathrm{N}$;
- to build Magnus representation (if it is necessary) — to calculate the matrices $\mathbf{G}_k$, $k \le \mathrm{N}$;
- to calculate the matrices $\mathbf{H}_k$, $k \le \mathrm{N}$ for factorized representation (4);
- to create (step by step) the solving matrices $\mathbf{M}^{1k}$, $k \le \mathrm{N}$;

So we can separate all submodules into four levels. The zero level group – the kernel level – is intended for manipulations by noncommutative variables and for realization of matrix algebra procedures (extended by the Kronecker product and sum [5]). The first level group of submodules is intended for preparing of all matrices which are necessary for next steps. The second level submodules are meant for symbolic manipulation for preparing corresponding databases and program packages. Here the basic objects are matrices from which we build desired solutions for a beam line considered as a dynamical system. These three groups of submodules can be realized with the help of computer algebra codes (i.e. such as $REDUCE$, $MAPLE\,V$, $MuPAD$). The third level submodules play the main role for our problem. Namely, the final result of this submodules group is a map generated by the transport system under study. There are some differences from usual approaches for this problem. According to this approach a particular map can be extracted from a corresponding database. If the desired solving matrices are absent in this database than one must turn to corresponding calculation submodules and make necessary calculations with the help of the two first group of submodules.

3.3 Organization of Calculations

Naturally, all above mentioned operations must be provided with a suitable interface. Here we should note that the interface problem is created not only for convenience of a user. The main role of this interface to optimize the modelling process, to provide adaptivity and efficiency of all manipulations [11]. Modern systems of visual programming (what is known as Rapid Application Development systems) such as $DELPHI$, $C^{++}\,Builder$ are given necessary tools for this. It is obviously that an expert system can be designed only by a researchers group which consists of adepts in different scientific domains. Only their co–operation can be lead to the desired goal: creation of an expert system (more exactly a prototype of the expert system). The complexity of the modelling process and of physical behaviour of particles in beam lines (first of all in nonlinear systems) make very difficult a formalization process and creation of a rules set.

4 REFERENCES

[1] S.N.Andrianov: 'Computer Algebra and Object–Oriented Programming', Abstr. of the Int. Conf. on Interval and Comp. Algebra in Sci. and Engineering', SPb, 1994, p.41.

[2] S.N.Andrianov: 'Algebraic Approach to Perturbation Theory and Computer Algebra', Proc. of the Int. Workshop "New Computer Technology in Control Syst.", Pereslavl–Zalesky, Russia, 1994, p.10.

[3] A.J.Dragt, J.Finn: 'Lie Series and Invariant Functions for Analytical Symplectic Maps', J.of Math. Phys., **17**, 1976, p. 2215.

[4] P.V.Koseleff: 'Formal Calculus for Lie Methods in Hamiltonian mechanics', Ph.D. Thesis, Ecole Polytechnique, 1993.

[5] S.N.Andrianov: 'Group–Theoretical and Algebraic Modelling of Control Systems for Particle Beam', Problems of Mechanics and Control Processes, issue 15, S.-Petersburg, SPbSU, 1992, p.7 (in Russian).

[6] S.N.Andrianov:'Construction of Approximate Symmetries and Invariants for Dynamical Systems', Proc. of the Second Int. Workshop on Beam Dynamics & Optimization – BDO'95, SPb, 1996, p.25.

[7] S.N.Andrianov: 'σ– Matrix Application for Solving of Problems of Charged Particle Beams Transport', Mathematical Theory for Technology Objects, Leningrad, LSU, 1982, p.114 (in Russian).

[8] S.N.Andrianov: 'The Explicit Forms for Lie Transformations', Proc. of the Fifth European Particle Conference EPAC'96, SITGES (Barcelona), Spain, 1996, p. 998.

[9] S.N.Andrianov: 'Analytical Simulation of Space Charge in Ion–Optical Systems', Proc. of the First Int. Workshop on Beam Dynamics & Optimization – BDO'94, SPb, 1995, p.19.

[10] S.N.Andrianov, A.I. Dvoeglazov: 'A Solving Module for Hamiltonian Dynamical Systems', Proc. of the Third Int. Workshop on Beam Dynamics & Optimization – BDO'96, SPb, will be published.

[11] S.N.Andrianov: 'Dynamic Modeling in Beam Dynamics', Proc. of the Second Int. Workshop on Beam Dynamics & Optimization – BDO'95, SPb, 1996, p.25.