Computer Assisted Mechanics and Engineering Sciences, 3: 289–296, 1996.

On derivation of equations of dynamics for a certain class of models of mechanical systems

Witold Grabysz
Silesian University of Technology, Engineering Mechanics Department (RMT4)
Konarskiego 18a, 44-100 Gliwice, Poland

(Received October 10, 1995)

Mechanical systems are quite often modelled as sets of stiff and flexible elements. If some conditions concerning connections of these elements are fulfilled, then we get a physical model having an interesting feature: it is quite easy to derive equations of motion from a diagram representing in a natural, intuitive manner the structure of the system. The algorithm for such derivation, including creation of a simple graphical user interface is presented in the paper. The algorithm takes advantage of the software package MATLAB/SIMULINK.

1. Introduction

During the recent decades *the modelling* has become a widely approved technique for investigating new products before introducing them into the market, as well as for improving the existing ones. The modelling removes the need for making experiments (testing) on real existing objects (prototypes) and in such a way, it may accelerate the course "from a concept to the production", lower its cost, sometimes even save human health and life.

The process of modelling can be roughly divided into two stages:

- preparing of the mathematical model, i.e. a set of mathematical equations which sufficiently (with respect to the assumed requirements) represent the investigated object;
- manipulation on the equations produced in the first stage.

Both stages can be performed manually but now we observe a tendency to employ computers.

For the reasons which will become clear in the remaining part of the paper, let us introduce (remind) two concepts: the *physical model* and the *system.*

1.1. Physical model

It is useful to divide the preparation of mathematical model into two substages: creation of a physical model and the assembly of the equations. What is the physical model? It is a simplified representation of the real object which consists only of *elementary models.* The elementary models are delivered and investigated by appropriate branches of physics. For example, in the domain of mechanics these elementary models are: material point, rigid body, linear spring, viscous damper, rod, beam and many others. It seems very sensible to implement the following idea:

A researcher (designer) prepares only a physical model. The equations are derived from the physical model automatically.

Owing to such an organization, a researcher is freed from the burden of thinking about the mathematical representation of the elements and processes he is working on. Instead of this, he thinks about the evaluated problem in terms of the notions from the field which is known to him very well. Additionally it is desirable to facilitate the creation of the physical model by introducing *a graphical interface.* It is a user-friendly program which translates a graphical representation of the object made by the researcher on the screen of a computer into such a description which can be later transformed into the equations.

The above mentioned idea is implemented very nicely by the well-known multi-body analyzer ADAMS.

1.2. System

It is very important for a correct modelling to fulfill *the principle of extraction.* It is assumed that the environment can affect the object only by *input variables*; the object in turn can answer by *output variables.* There is no feedback in the environment between output and input variables or — in other words — we assume that the output variables can not affect the input ones. It is why authors speak very often about *modelling of systems.* Let us remind that *a dynamic, linear, stationary, continuous system* is a system (not defining the term "system") of which the input $\mathbf{u}$, output $\mathbf{y}$ and *state variables* $\mathbf{x}$ are related by the following equations:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)\,, \tag{1}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t)\,, \tag{2}$$

where matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ are constant.

Afterwards in the paper we will focus just on continuous mechanical systems. Continuous not in terms of material domain but as far as time is concerned. Of course, during numerical solution, differential equations are changed into discrete differences, but it has nothing in common with the fact that the original problem was continuous.

If equations (1) and (2) model a mechanical system, then eigenvalues of the matrix $\mathbf{A}$ should correspond to the natural frequencies.

2. Physical models of the MaSp type

Let us divide all elementary models into two classes: *dynamic elements* and *static elements.* As far as mechanics is concerned, such elementary models as material point and rigid body belong to the first class when spring (linear or not) and damper represent the latter. Let us assume, that elementary models can exchange information one with another through *ports*, or — in other words — they can be connected only by ports. What does it mean that two elements are connected by a port? It means that an element receives a portion of information via a port and answers also via the port. So, a port is a logical interface to build a whole structure from elements. It must be emphasized that a port is a two-way interface: an element receives and sends data. Of course, the output and input are related by the function realized by the element.

As an example, let us have a closer look at elementary model "material point". It has only one port. (It is possible to assume it has several ports, but in such a case they are identified as the same one.) The element receives a value of force $\Sigma\mathbf{F}$ acting on it and answer with a value of displacement $\mathbf{x}$ (not to be confused with the vector of state variables). The values are connected by the formula

$$\Sigma\mathbf{F} = m\ddot{\mathbf{x}}\,. \tag{3}$$

The mass m is a parameter of the element.

The elementary model "linear spring" is an example of a two-port element. One port receives displacement of one end of the spring, the second port — displacement of the second end. The values of forces are the output variables of the element. The values sent by ports differ only in sign. Their absolute value is calculated according to the formula

$$F = k(\lambda - \lambda_0) \tag{4}$$

where λ and λ_0 are the actual and neutral length of the spring.

Let us notice that the discussed elements can

- either receive values of displacement (possibly also values of velocity as in case of a damper) and send values of force,
- or receive values of force and send values of displacement (and velocity).

So it seems logical to propose the following physical model:

Definition 1 (of the MaSp model)

The physical MaSp type model of a system is a model which

1. *Consists of elements being elementary models which are connected by clearly defined ports.*
2. *The port of a dynamic element can be connected with every number of ports of static elements. In such a case the inputs received by the port are to be summed.*
3. *The port of a static element can be connected with one and only one port of a dynamic element.*

The proposed name "MaSp" is derived from "mass-spring". In the next chapter it will be shown, that in linear case it is quite easy to derive equations of dynamics from MaSp models or at least to perform numerical simulation without deriving these equations *explicite*.

Let us here state it once more: a port is a two-way communication interface. It sends as well as receives data. It is of no use to discriminate the input from the output in the port. It is also why connections via ports are symmetrical: one can not attribute a direction to the connection.

We can imagine a situation where in order not to sink in details, we have built the model from *submodels*. Submodels can consist of submodels of the second order and so on, but eventually they are built just from elementary models, however this knowledge is not important for the user of the submodels. If we use complex submodels having several ports, it makes no sense to limit ourselves to the case that external ports (to be connected with other submodels) belong to elements of the same type. It is practical to shift the concept "dynamic–static" from the element to the port itself. If so, let us improve the Definition (1):

Definition 2 (of the MaSp model, improved)

The physical MaSp type model of a system is a model which

1. *Consists of submodels which are connected by clearly defined ports of two possible types: dynamic or static.*
2. *The dynamic port can be connected with every number of static ports. In such a case the inputs received by the port are to be summed.*
3. *The static port can be connected with one and only one dynamic port.*

A question arises if MaSp models are useful. In author's opinion they are not a general solution for every kind of technical problems, but in some areas it is a good idea to try to use them. Especially they are used in the case of driving systems, which can be treated as a chain of stiff masses (toothed wheels) connected by springs (flexibility of meshings) [4, 6, 7]. The application of MaSp models in investigation of dynamics of *multi-body systems* was proposed by Zeid [8, 9], but its usefulness is controversial. MaSp models were also investigated by the author [1, 2, 3].

Perhaps it would be also possible to apply the described methodology in other fields: electricity, hydraulics or in modelling of mixed (e.g. electro-mechanical) systems, but it requires further investigation.

3. Use of MaSp models in automatic derivation of equations of motion

In this section we will present how to derive equations of motion for some mechanical systems represented by MaSp models. A software package MATLAB/SIMULINK [5] will be used. MATLAB ("Matrix Laboratory") is a program for matrix and vector calculations and SIMULINK is a MATLAB's extension for investigation of dynamic systems. The use of MATLAB/SIMULINK is not obligatory. Other general-purpose simulation programs or even house-developed ones can also be used.

It is obvious, that prior to transforming a physical model into a set of equations, it is necessary to record the structure of the investigated system in the form understandable by the computer. Of course, feeding the computer with the data must be as convenient for the user as possible. Using the possibilities of SIMULINK we will build a kind of a simple user interface.

3.1. Linear examples

Let us define in SIMULINK *a superblock* "Mass". It is shown on the screen as a rectangle with one input and one output. After double-clicking the rectangle reveals its interior, shown in Fig. 1

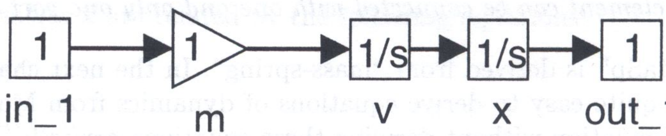

Fig. 1. Superblock "Mass"

Let us define also a superblock "Spring" for modelling a linear spring. It has two inputs and two outputs corresponding to the ports described earlier. Being equipped with these two superblocks, we can describe the structure of the simple mechanical system shown in Fig. 2 as a SIMULINK block diagram presented in Fig. 3. We can see arrows in this figure which collides with the feature of symmetry of connection, but let us notice that these arrows appear always in pairs (well, there is no arrow in Fig. 3 pointing from the spring to the constant value which stands for the base, but formally it is no reason it could not be), so it would be better to describe the structure as shown in Fig. 4, but it is not feasible in SIMULINK to represent two reversed arrows as a plain line.

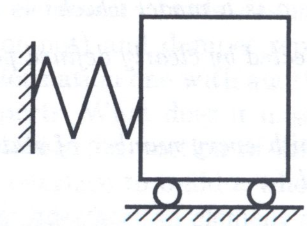

Fig. 2. A system with one DOF

Using copies of the defined superblocks we can record in the format digestible by SIMULINK another, a little more complicated system shown in Fig. 5. The record itself is visible in Fig. 6.

Having described the structure of the system in the form of a block diagram, it is possible to order SIMULINK to simulate the system and monitor the chosen variables (compare [3]). Here however we obtain equations of motion according to formulation (1) and (2). In order to do it, let us type in the SIMULINK window the following command:

```
[A,B,C,D] = linmod('masp3')
```

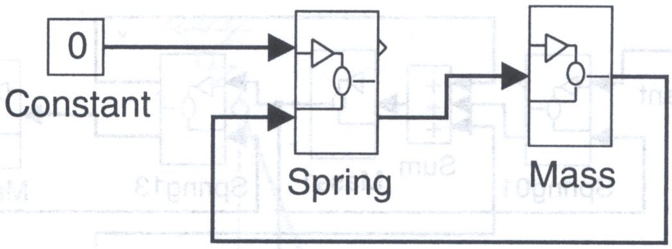

Fig. 3. The block diagram describing the system shown in Fig. 2

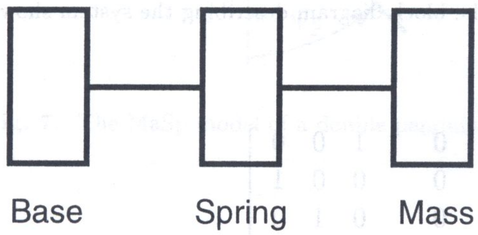

Fig. 4. An improved but not supported by SIMULINK block diagram describing the system shown in Fig.2

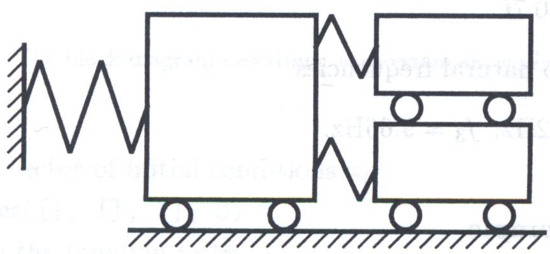

Fig. 5. A system with three DOFs

The function `linmod` obtains a linear model from the SIMULINK block diagram titled (in this case) "masp3". "`linmod` perturbs the states around the operating point to determine the rate of change in the state derivatives and outputs. This result is used to calculate the state–space matrices" [5]. In the considered case we assume that masses and stiffness coefficients of springs are

$$m_1 = 1\,\text{kg}, \qquad k_{01} = 10000\,\text{N/m},$$
$$m_2 = 0.5\,\text{kg}, \qquad k_{12} = 10000\,\text{N/m},$$
$$m_3 = 1\,\text{kg}, \qquad k_{13} = 40000\,\text{N/m},$$

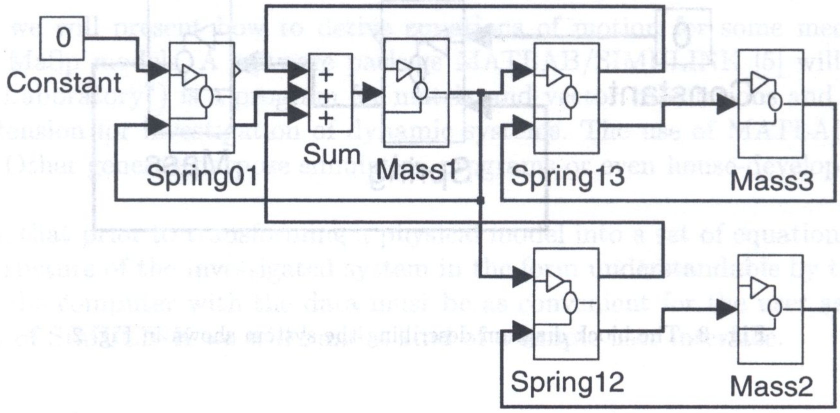

Fig. 6. The block diagram describing the system shown in Fig.5

and `linmod` produces

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ -60000 & 10000 & 40000 & 0 & 0 & 0 \\ 40000 & 0 & -40000 & 0 & 0 & 0 \\ 20000 & -20000 & 0 & 0 & 0 & 0 \end{bmatrix}$$

with **B**, **C** and **D** being zero matrices because inputs and outputs of the whole system have not been defined. Now it is possible — for example — to find natural frequencies of the system. Typing a command `eig(A)` we obtain eigenvalues of matrix **A**

$$\pm 304.89i;\ \pm 152.84i;\ \pm 60.7i$$

which should correspond to natural frequencies

$$f_1 = 48.52\text{Hz};\ f_2 = 24.32\text{Hz};\ f_3 = 9.66\text{Hz}.$$

3.2. Quasi-nonlinear example

The function `linmod` can be used also in case of a nonlinear system. Then it performs a linearization and it is necessary to give the point in which the linearization must be done. Let us trace the following example concerning a double pendulum, for which we will find natural frequencies for small displacements.

A double pendulum belongs to multi-body systems. In order to apply MaSp methodology to such systems we must assume, that some rigid components of the system are flexible. They can be very, very stiff, however flexible. In our example let us assume the links are just springs, as shown in Fig. 7.

We have to define a two-port superblock "rigid body". After doing this we can build the block diagram shown in Fig. 8.

The function `linmod` is called

```
[A,B,C,D] = linmod('pendulum', x)
```

with the additional argument **x** — the point of the linearization. It can be set manually, but it is

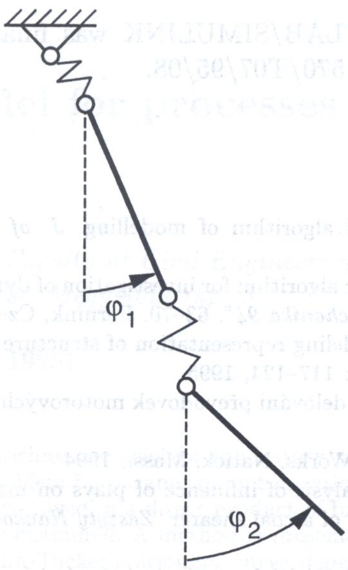

Fig. 7. The MaSp model of a double pendulum

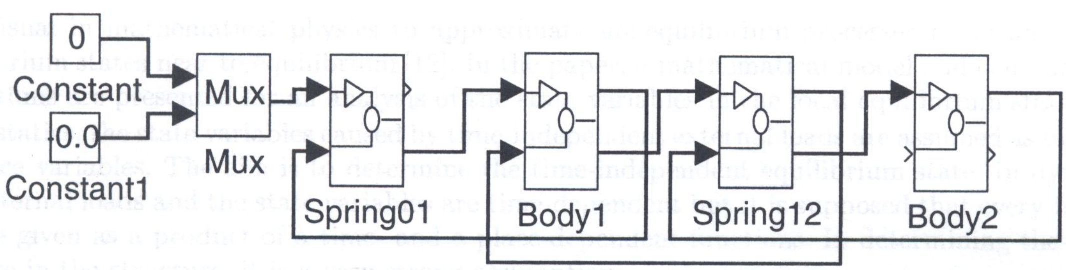

Fig. 8. The block diagram describing the system shown in Fig.7

more convenient to find the vector of initial conditions $\mathbf{x}_0$

```
[sizes, x0] = pendulum([], [], [], 0)
```

and afterwards, to pass it to the function `trim`

```
[x,u,y] = trim('pendulum', x0).
```

The function `trim` "tries to find an equilibrium point such that the maximum absolute value of $[\mathbf{x}; \mathbf{u}; \mathbf{y}]$ is minimized" [5]. "Finding" the vector of initial conditions requires an explanation: of course, the initial conditions are set by the user in superblocks, but the order in which they appear in vectors $\mathbf{x}$ and $\mathbf{x}_0$ depends on many factors, so the easiest way is to apply the procedure given above.

The eigenvalues of matrix $\mathbf{A}$ are

$$\pm 161.8i;\ \pm 97.16i;\ \pm 66.67i;\ \pm 61.8i;\ \pm 7.16i;\ \pm 2.66i.$$

The greatest four values correspond to comparatively high frequencies caused by the presence of stiff ($k = 10000$ N/m) springs. The remaining two values comply with natural frequencies of a double pendulum built of two rods having length 1 m and mass 1 kg both.

4. Acknowledgement

The purchase of the package MATLAB/SIMULINK was financed by the State Committee for Scientific Research as the grant PB 570/T07/95/08.

References

[1] W. Grabysz. Information flow based algorithm of modelling. *J. of Theoretical and Applied Mechanics*, **32**: 945–957, 1994.

[2] W. Grabysz, E. Świtoński. Yet another algorithm for investigation of dynamics of multi-body mechanical systems. In: *Proc. of the Conf. "Výpočtová mechanika 94"*, 63–70. Pernink, Czech Republic, 1994.

[3] W. Grabysz. A convenient in the modeling representation of structure of mechanical systems. *Zeszyty Naukowe Politechniki Śląskiej, Mechanika*, **121**: 117–124, 1995.

[4] V. Kotek, P. Heriban. Přispěvek k modelováni převodovek motorových vozidel. *Inženýrská mechanika*, **1**: 39–46, 1992.

[5] *SIMULINK. User's guide.* The MathWorks, Natick, Mass., 1994.

[6] E. Świtoński, D. Dubiel, Z. Rak. Analysis of influence of plays on magnitudes of dynamic forces in kinematic pairs of a driving system of gearheads of a coal shearer. *Zeszyty Naukowe Politechniki Śląskiej, Mechanika*, **113**: 117–124, 1993.

[7] J. Vondřich. Dynamická analýza mechanických systémů. *Inženýrská mechanika*, **1**: 23–30, 1992.

[8] A. Zeid, D. Chang. A modular computer model for the design of vehicle dynamic control systems. *Int. J. of Vehicle System Dynamics*, **18**: 201–221, 1989.

[9] A. Zeid. Bond graph modeling of planar mechanisms with realistic joint effects. *Transactions of the ASME, J. of Dynamic Systems, Measurement, and Control*, **111**: 382–388, 1989.