SIMULATION AND VISUALISATION OF A LABORATORY MODEL BY SUPPORT SOFTWARE TOOLS CONNECTION

Jiri Zatopek

Department of Automatic and Control Engineering, Faculty of Applied Informatics, Tomas Bata University in Zlin
Nad Stranemi 4511, 760 05 Zlin, Czech Republic

Abstract

This article introduces the possibilities of the simulation and visualisation of the "Twin-Rotor MIMO System" laboratory model outputs by means of various support software tools. The 3D model of the system, (used for simulation and visualisation), is designed in SolidWorks 3D CAD software. Matlab/Simulink with extension libraries like Simscape and 3D Animation - (formerly Virtual Reality Toolbox), is used for 3D visualisation and simulation. The 3D Animation toolbox is only used for the visualisation of the mathematical and real models. The Simscape library - on the other hand, is used for the validation of the reverse control of the derived mathematical model´s correctness and for simulation and analysis purposes as a suitable substitution for real models. As a result of this, these supporting software tools streamline the overall suggested controls - from analysis to presentation of the results.

Keywords: Motion Control; Simulation; Visualisation; Matlab/Simulink; Simscape; 3D Animation

1. Introduction

Today, a wide range of support software tools are available. These tools facilitate work in analysing, modelling, simulating, and designing control laws for complex mechanical structures. By interconnecting these resources, a considerable simplification of the synthesis can be achieved - as well as clarity, by using visualisation in 3D space directly in the regulation process, or by creating a physical model in Simulink directly from CAD software (SolidWorks), without the need to derive the motion equations [3, 10].

The "Twin Rotor MIMO System" laboratory model is the subject of interest. It contains Teaching Toolboxes with a derived mathematical model, a Simulink model and 1DOF and 2DOF PID controllers [7, 11], which are used as a link between the laboratory model, the physical (Simscape) model, and the graphic (3D Animation) model.

First of all, the 3D model will be designed in the 3D CAD software – SolidWorks environment. The precision of the 3D CAD model should be a compromise between the complexity of the model and the actual impact on the overall dynamics of the system – i.e. it must adequately reflect the behaviour of a real model. The model contains masses of

individual parts and the resulting dynamic properties of the entire assembly [9, 2]. The model, designed using 3D SolidWorks, is exported directly into a suitable format for the 3D Animation toolbox and the Simscape/SimMechanics library [10].

The 3D Animation toolbox - intended for visualisation, uses only the appearance properties of the SolidWorks model and is applied to the mathematical and laboratory model outputs´ visualisation. The mathematical model´s outputs or captured laboratory model variables (for virtual animation purposes) are used as 3D Animation model inputs [6, 9]. These parameters are converted into the appropriate format by a transformation matrix using Euler parameters [1, 3, 4].

The Simscape – i.e. the SimMechanics specifically, uses the SolidWorks model as a whole (e.g. its material properties, physical links, etc.) to build a physical model. The model can be used for analysis and simulation purposes, (the mathematical model does not need to know), as well as to validate the mathematical model´s correctness [10, 8, 5].

The results of this work could be used for: inductive identification, design of control laws, mathematical model correctness validation, or the presentation of results.

2. Laboratory model

The laboratory model is used for educational purposes as an example of a non-linear motion system. It represents a model similar to a tethered helicopter, with orthogonally disposed axes of rotation. The system is controlled by the voltage change of a pair of DC motors, in the 0 to 5V range. The laboratory model output is information about rotations in the vertical (i.e. elevation); and horizontal (azimuth) directions. Measurements of the rotation information is realised by incremental sensors [7, 11].
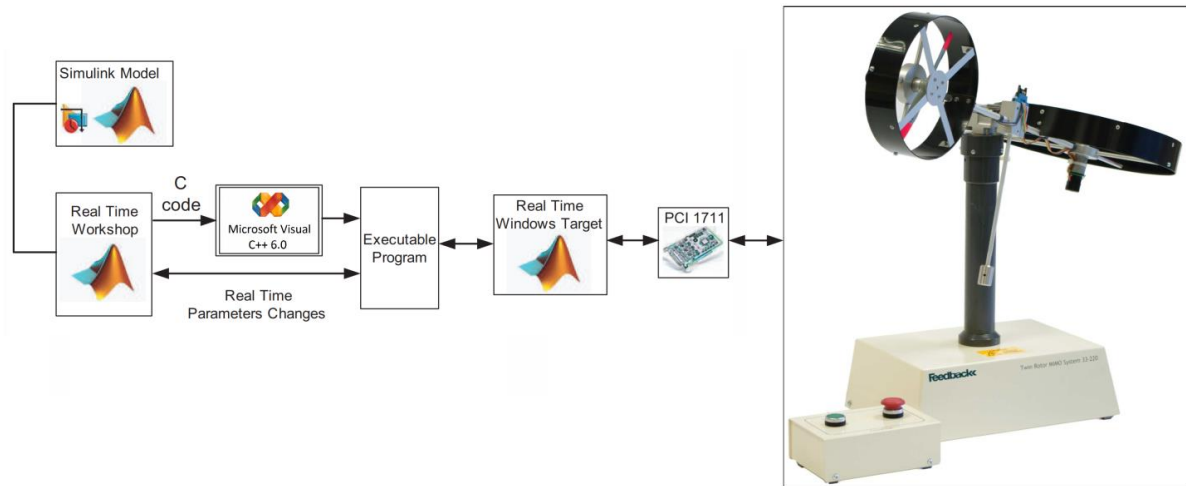

Fig. 1 Laboratory model in a block chart [7, 12]

3. SolidWorks model

The SolidWorks model consists of 9 parts - as shown in Fig. 2.b. Each part has an associated local coordinate system, mass, and material, and is bounded by structural bonds to the other parts of the assembly (Fig. 2.a). Working directly in SolidWorks, it is possible to determine the centres of mass positions, the inertia matrices or even to execute a physical analysis for example.
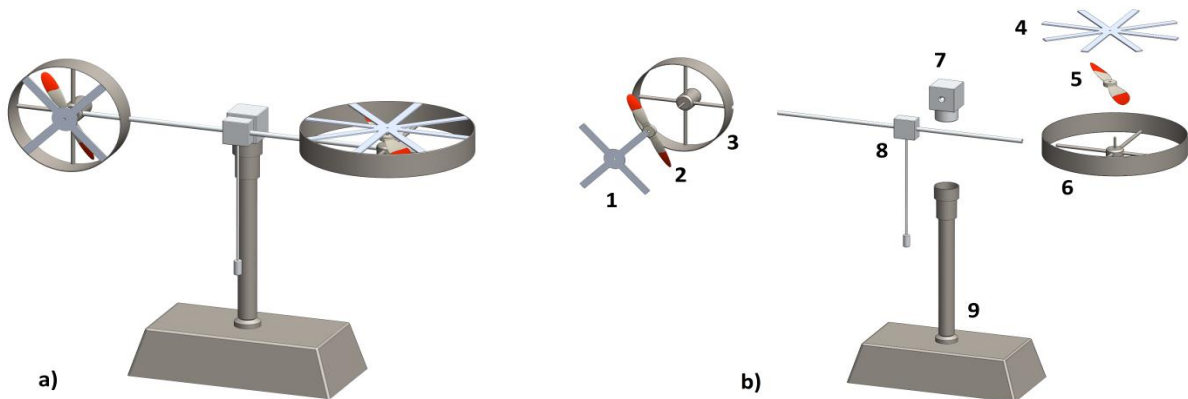

Fig. 2 SolidWorks model (a); and exploded view (b)
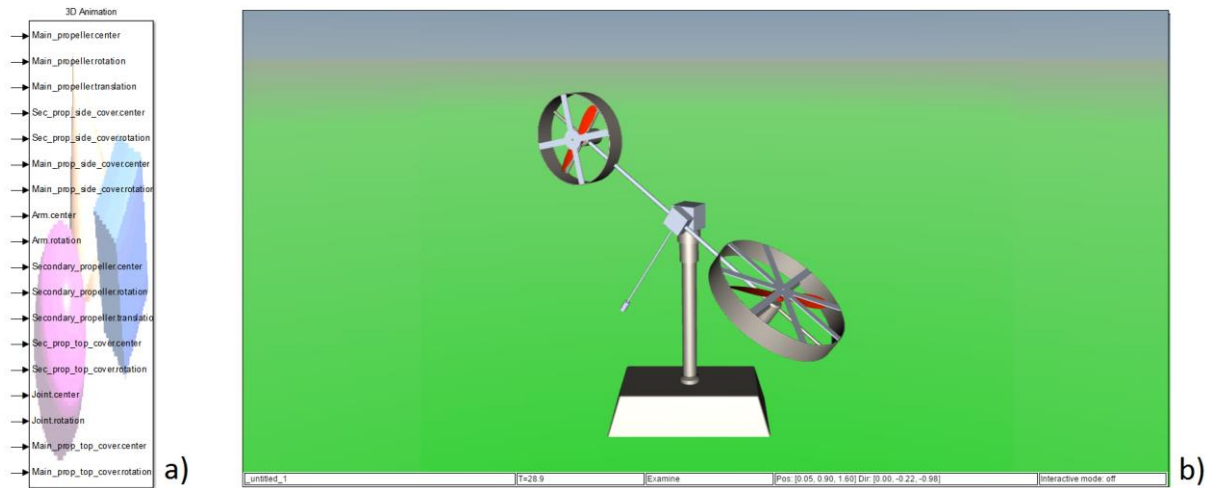
4. 3D Animation

Fig. 3 VR Sink block (a), and Web-browser visualisation (b), in 3D Animation

The 3D CAD model cannot be exported into a format for 3D Animation with linkages; thus all nine bodies are freed up. Their position in 3D space is determined by a direction vector with its origin in a global coordinate system, and its orientation is determined by a unit vector and the angle of rotation around this unit vector. However, the only parameters available are those of the azimuth and elevation - by means of which it is possible to calculate the desired parameters (transformation matrices and Euler's Theorem will be used for this) [9].

4.1. Transformation of coordinate systems

The transformation matrices are used to determine the position and orientation of a local coordinate system firmly connected with the component in the overall global coordinate system. In this case, we need to know only 3 transformation matrices since Part 9 in Fig. 2.b is fixed (i.e. connected to the ground); while Part 7 rotates azimuthally around a global axis (i.e. there is no need for a transformation matrix); and Parts 1, 3, 4, 6, 8 in Fig. 2.b are connected by mutual firm bonds - they are only relatively displaced (i.e. a mutual rotation transformation matrix).
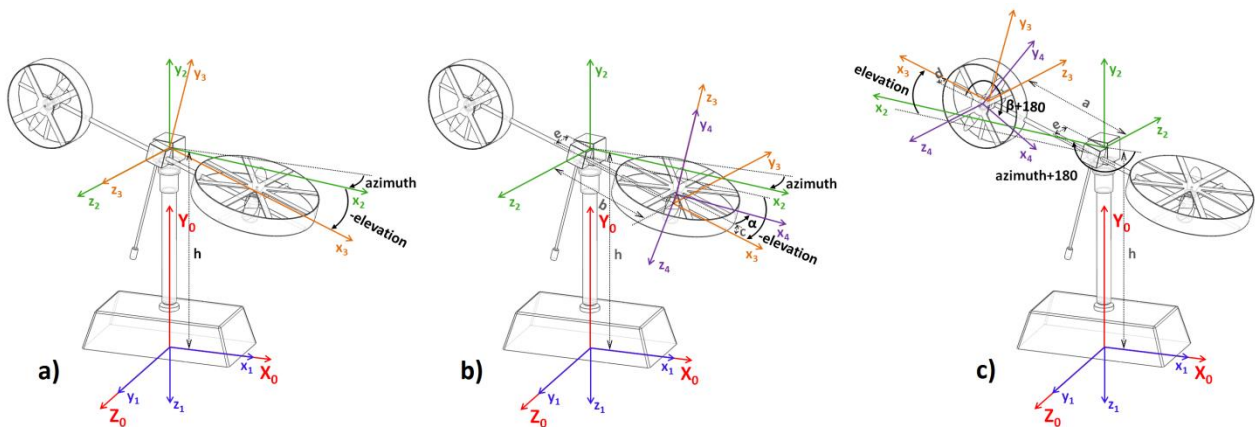

Fig. 4 Model with the selected coordinate system placement

The coordinate system placement for the determination of the transformation matrix is shown in Fig. 4. Fig. 4.a has the last local coordinate system linked with an arm-mounting point, and the resulting transformation matrix is used to determine the orientation of Parts 1, 3, 4, 6, 8 (shown in Fig. 2.b) in 3D space. Fig. 4.b has the last local coordinate system linked with the centre of rotation of the main propeller, and the resulting transformation matrix is used for determining the orientation of Part 5. Fig. 4.c has the same use as Fig. 4.b - but is for the secondary propeller.

The Denavit-Hartenberg notation/method of coordinate systems placement (DH) [1, 9] is used for the derivation of the transformation matrix. The interface for determining the general (i.e. symbolically specified) transformation matrix was created in Matlab. It has a graphical interface, and uses the Symbolic Math Toolbox (Fig. 5).

The resulting transformation matrices are transformations from the local coordinate system into the global coordinate system ($^{G}T_{B}$) [1,9] and are listed in (1) for Fig. 4.a, (2) for Fig. 4.b and (3) for Fig. 4.c.
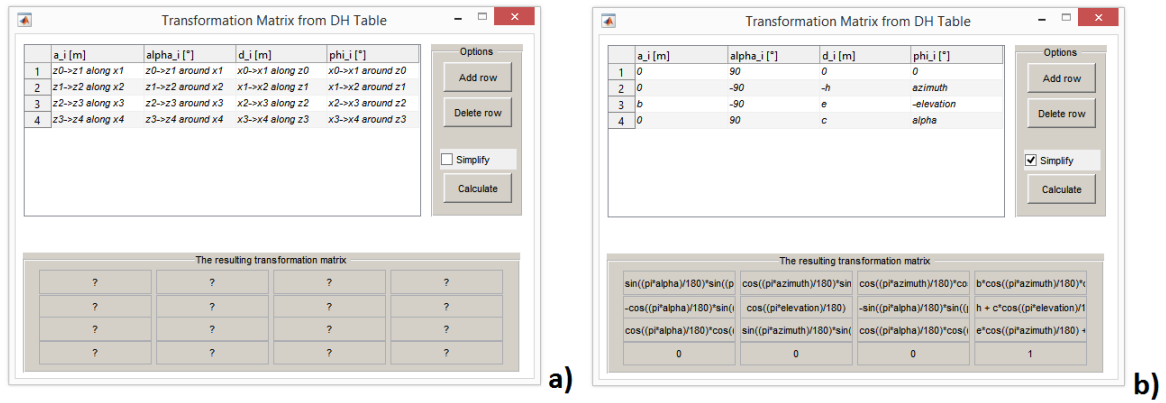
Fig. 5 Matlab program - designed for determining the transformation matrix

$$
^{G}T_{1_{B}} = \begin{bmatrix} cos(\varepsilon)cos(\varphi) & cos(\varepsilon)sin(\varphi) & -sin(\varepsilon) & 0 \\ -sin(\varphi) & cos(\varphi) & 0 & h \\ sin(\varepsilon)cos(\varphi) & sin(\varepsilon)sin(\varphi) & cos(\varepsilon) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\tag{1}
$$

$$
^{G}T_{2_{B}} = \begin{bmatrix} sin(\alpha)sin(\varepsilon)+cos(\alpha)cos(\varepsilon)cos(\varphi) & cos(\varepsilon)sin(\varphi) & sin(\alpha)cos(\varepsilon)cos(\varphi)-cos(\alpha)sin(\varepsilon) & b \cdot cos(\varepsilon)cos(\varphi)-e \cdot sin(\varepsilon)+c \cdot cos(\varepsilon)sin(\varphi) \\ -cos(\alpha)sin(\varphi) & cos(\varphi) & -sin(\alpha)sin(\varphi) & h+c \cdot cos(\varphi)-b \cdot sin(\varphi) \\ cos(\alpha)sin(\varepsilon)cos(\varphi)-sin(\alpha)cos(\varepsilon) & sin(\varepsilon)sin(\varphi) & cos(\alpha)cos(\varepsilon)+sin(\alpha)sin(\varepsilon)cos(\varphi) & e \cdot cos(\varepsilon)+b \cdot sin(\varepsilon)cos(\varphi)+c \cdot sin(\varepsilon)sin(\varphi) \\ 0 & 0 & 0 & 1 \end{bmatrix}
\tag{2}
$$

$$
^{G}T_{3_{B}} = \begin{bmatrix} -cos(\beta+\varphi)cos(\varepsilon) & -sin(\beta+\varphi)cos(\varepsilon) & -sin(\varepsilon) & -d \cdot sin(\varepsilon)-e \cdot sin(\varepsilon)-a \cdot cos(\varepsilon)cos(\varphi) \\ sin(\beta+\varphi) & -cos(\beta+\varphi) & 0 & h+a \cdot sin(\varphi) \\ -cos(\beta+\varphi)sin(\varepsilon) & -sin(\beta+\varphi)sin(\varepsilon) & cos(\varepsilon) & d \cdot cos(\varepsilon)+e \cdot cos(\varepsilon)-a \cdot sin(\varepsilon)cos(\varphi) \\ 0 & 0 & 0 & 1 \end{bmatrix}
\tag{3}
$$

In which:

ε ... azimuth
φ ... elevation

In order to determine the orientation of the individual parts of the system, the Axis-angle representation was predicted using Euler's Rotation Theorem [9].

4.2. Axis-angle representation - Euler's Rotation Theorem

Euler's Rotation Theorem dictates that any rotation or sequence of rotations of a rigid body in a three-dimensional space is equivalent to a pure rotation about a single, fixed, axis. Using the transformation matrix ($^{G}R_{B}$) – i.e. Rodriguez' Rotation Formula, and Euler's Rotation Theorem, the following relations can be written [9]:

$$
\Phi = arccos\left(\frac{tr\left(^{G}R_{B}\right)-1}{2}\right)
\tag{4}
$$

$$
\tilde{u} = \begin{pmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{pmatrix} = \frac{1}{2 \cdot sin(\Phi)} \cdot \left[^{G}R_{B} - \left(^{G}R_{B}\right)^{T}\right]
\tag{5}
$$

Where:

(u_1, u_2, u_3) ... is the unit vector that determines the direction of the rotation axis
Φ ... is the rotation angle around the unit vector

After the substitution of the derived transformation matrix into (4) and (5), a transcription into the form of (u_1, u_2, u_3, Φ), we derive a vector which will be connected directly to a *VR Sink* block *rotation* parameter. The *centre* and *translation* parameters are also calculated using transformation matrices. The *centre* parameter is the origin of the unit vector, and the *translation* parameter is the position of the local coordinate system in the global coordinate system – a transformation

matrix multiplied by the vector (0, 0, 0, 1) [9]. It is possible to finalize the animation at this moment by using Euler's Rotation Theorem and transformation matrices above-mentioned. All parts are virtually bounded over the transformations connected to VR Sink Block shown in Fig. 3.a. Final 3D animation of the laboratory model is shown in Web-browser in Fig. 3.b.

5. The Simscape/SimMechanics

The SolidWorks 3D model is then exported into the SimMechanics Second Generation scheme, along with all linkages and masses - and therefore contains the dynamics of the entire assembly, and is suitable for simulation purposes. Damping coefficients and spring stiffness can be added to the contact surfaces and joints, the parts can be modified (e.g. by changing the masses, moments of inertia, etc.), or the whole assembly´s kinematic structure can be changed. Also, it is possible to simulate the torques based on the generated kinematic movements, for each joint.
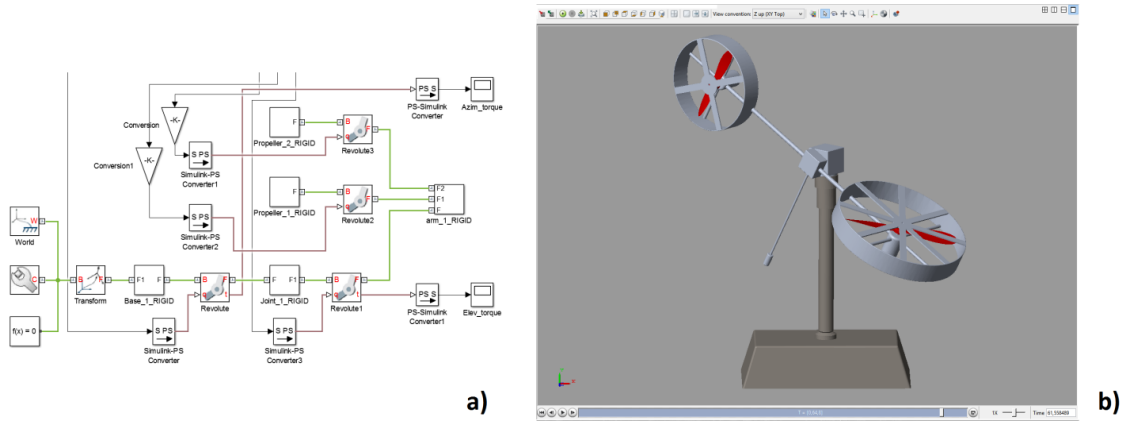

Fig. 6 SimMechanics Second Generation scheme (a) and its simulation (b)

The SimMechanics scheme is shown in Fig. 6.a. It includes 4 rotational kinematic pairs, which determine the types of linkages between the five independent parts (i.e. Base - Fig. 2.b.9; Joint - Fig. 2.b.7; Arm - Fig. 2.b.1, 3, 4, 6, 8; Propeller_1 - Fig. 2.b.2; Propeller_2 - Fig. 2.b.5). The gravitational field vector is in the negative direction of the global axis Y_0. Fig. 6.b shows a 3D simulation of a SimMechanics model. Thus, it is possible to observe dynamic processes occurring in the system in real time - with the possibility of slowing down, speeding up, or pausing the simulation.

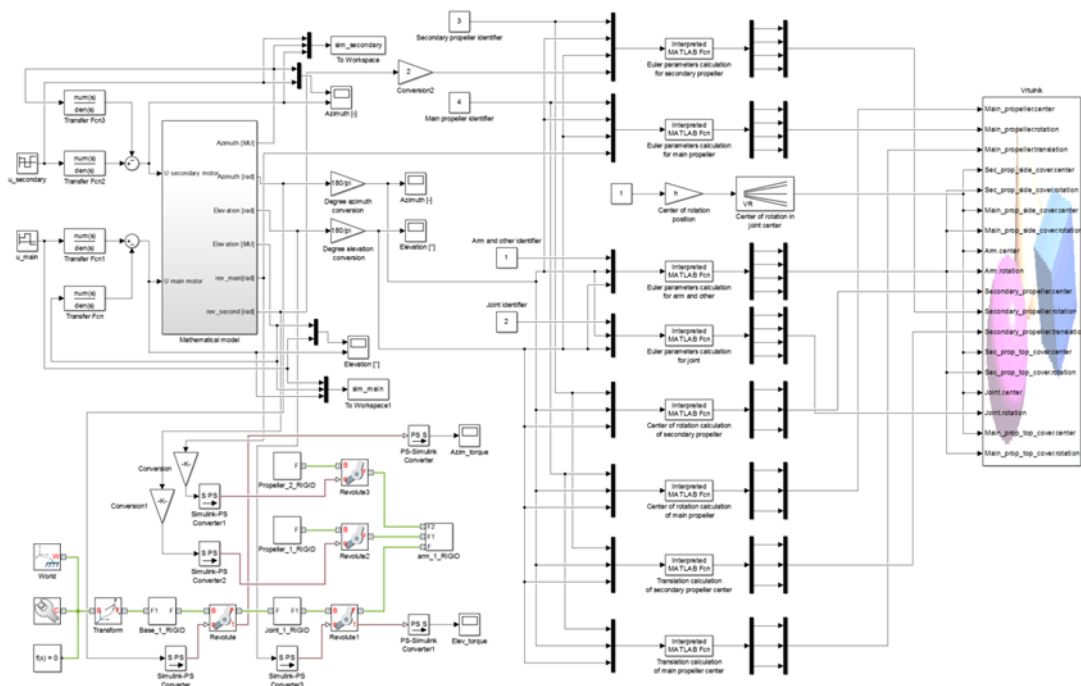## 6. The Matlab/Simulink interconnection

Fig. 7 Complete Simulink scheme

Fig. 7 depicts the complete Simulink scheme, which contains a Mathematical model part, the SimMechanics model, and the 3D Animation model with Euler Parameter Recalculations.
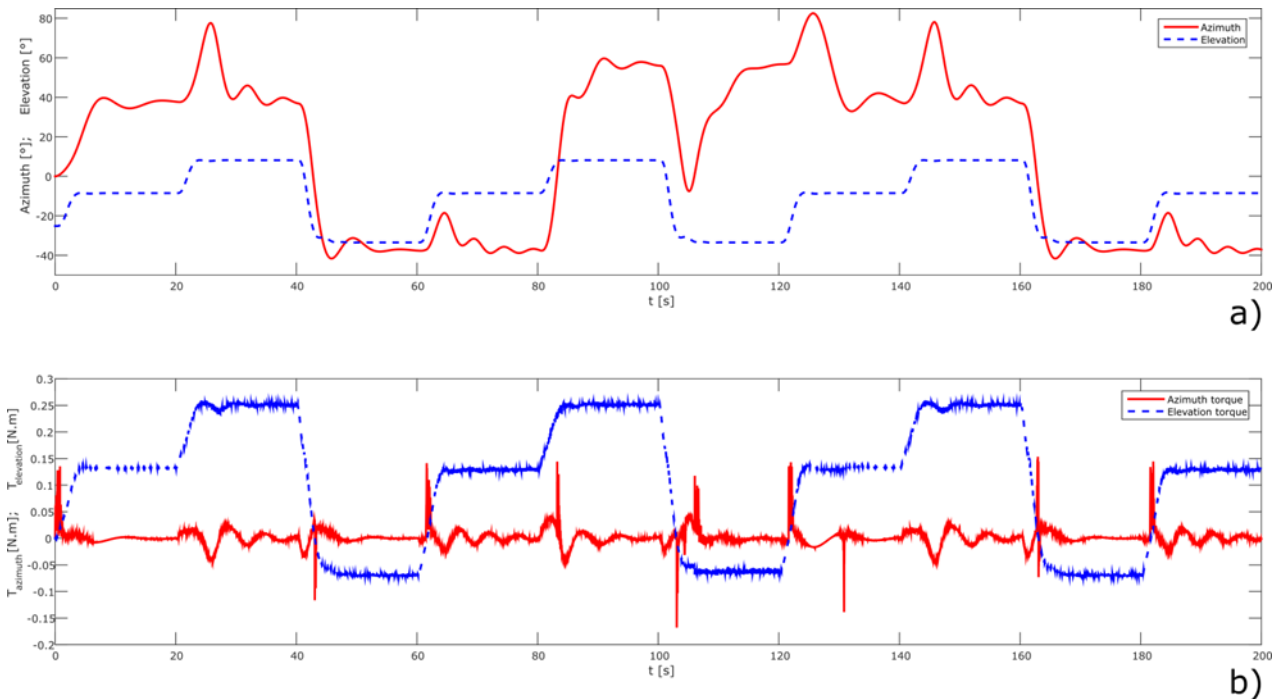

Fig. 8 Reference waveforms of laboratory model orientation (a) and required torque (b) needed to achieve them

The reference waveforms in Fig. 8.a are the controlled mathematical model outputs, which are then applied as required inputs into the SimMechanics model. The SimMechanics model is capable of generating the required moments (viz. Fig. 8.b), to perform the waveforms in Fig. 8.a. This is useful for instance, for the selection of suitable actuators for the system. If it is known - to what extent, and what dynamics the system requires, it is possible to use the SimMechanics model to determine what the requirements for the torque characteristics of the motor are.

In connection with the CAD model, the mathematical model, and the measurable state of the real model, the SimMechanics environment has many uses; this being only one of them.

7. Conclusion

Attention in this work was devoted to the simulation and visualisation of the "Twin Rotor MIMO System" laboratory model by using support software tool connections, which was the main intention of this paper. The whole assembly was designed in the SolidWorks 3D CAD system, which is used throughout this paper. The precision of the 3D CAD model should be a compromise between the complexity of the model and the actual impact on the overall dynamics of the system.

The first use of the 3D CAD model is for the visualisation of the mathematical model outputs using the 3D Animation toolbox. The position and orientation of each part is determined by The Denavit-Hartenberg Notation and Euler's Rotation Theorem. The Matlab interface was designed for the transformation matrix determined from the Denavit-Hartenberg table, and works with the Symbolic Math Toolbox. The 3D Animation toolbox is also applicable to visualise real model outputs and use of a web-browser could be helpful for the real-time presentation of a real model´s behaviour.

Another part relates to simulation by a physical model. This is also exported from the SolidWorks model, with all masses and links. The SimMechanics model could be used as a partial substitute for the real model - in the case where it is constructed with sufficient precision. The SimMechanics model was used as an aid for the selection of the appropriate actuators in this paper, by generating the required torque.

The significance of using the above-mentioned support software tools is their interconnection into one suitable software programme - Matlab. Simulation, visualisation, and other operations are solved by this interconnection; the Simulink model is prepared to further work with the "Twin Rotor MIMO System" laboratory model.

Further work will be focused on using the SimMechanics model for dynamic analysis purposes – e.g. the influence of the propellers´ rotation on the azimuth/elevation parameters from the perspective of centrifugal forces, the correction of friction coefficients by similarities observed between the real and physical models, or the online visualisation of real model outputs.

Support software tools connection is a popular theme today, especially 3D simulation/visualisation, not only for academics but also for students as a tool for the understanding of complex processes.

8- References

[1] Denavit, J. & Hartenberg, R. S. (1955). ASME Journal of Applied Mechanics", A kinematic notation for lower-pair mechanisms based on matrices, Vol. 77, No. 22, June 1955, pp. 215-221.

[2] R. N. Jazar. (2010). Theory of Applied Robotics, Springer US, ISBN: 978-1-4419-1750-8, Boston, MA

[3] Damic, V. & Cohodar, M. (2016). Dynamic Analysis and Visualization of Spatial Manipulators with Closed Structure, Proceedings of the 26th DAAAM International Symposium, pp.0109-0117, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-07-5, ISSN 1726-9679, Vienna, Austria, DOI: 10.2507/26th.daaam.proceedings.016

[4] Shala, A., Likaj, R., Bruqi, M. & Bajrami, X. (2014). Propulsion Effect Analysis of 3Dof Robot under Gravity, Proceedings of the 25th DAAAM International Symposium, pp.206-212, B. Katalinic (Ed.), Published by Elsevier Ltd, ISSN 1877-7058, Vienna, Austria, DOI: 10.1016/j.proeng.2015.01.359

[5] Sudharsan, J. & Karunamoorthy, L. (2016). International Journal of Simulation Modelling", Path Planning and Co-Simulation Control of 8 DOF Anthropomorphic Robotic Arm, Vol. 15, No. 2, June 2016, pp. 302-312., ISSN: 1726-4529

[6] Flanders, M. & Kavanagh, R. C. (2015). Computer Applications In Engineering Education", Build-A-Robot: Using virtual reality to visualize the Denavit–Hartenberg parameters, Vol. 23, No. 6, Apr 2015, pp. 846-853., DOI: 10.1002/cae.21656

[7] http://www.feedback-instruments.com/products/education/control_instrumentation/twin_rotor_mimo, (2016). Feedback, Twin Rotor MIMO System, Accessed on: 2016-09-23

[8] Urednicek, Z. (2009). Electromechanical actuating devices, (In Czech) T. Bata university in Zlin, ISBN: 978-80-7318-835-1, Zlin

[9] Urednicek, Z. (2012). Robotics, (In Czech) T. Bata university in Zlin, ISBN: 978-80-7454-223-7, Zlin

[10] Zatopek, J. (2016). XLI. Seminar ASR '2016 "Instruments and Control", States of Nonlinear Unstable System Simulation, Connections between SolidWorks-Matlab/SimulinkSimMechanics-3D Animation, Apr 2016, pp. 163-172., ISBN: 978-80-248-3910-3, VSB-TUO, Ostrava

[11] Chalupa, P., Prikryl, J. & Novak, J. (2014). Modelling of Twin Rotor MIMO System, Proceedings of the 25th DAAAM International Symposium, pp.249-258, B. Katalinic (Ed.), Published by Elsevier Ltd, ISSN 1877-7058, Vienna, Austria, DOI: 10.1016/j.proeng.2015.01.365

[12] Belmonte, L. M., Morales, R., Fernandez-Caballero, A. & Somolinos, J. A. (2016). Sensors, Robust Decentralized Nonlinear Control for a Twin Rotor MIMO System, Vol. 16, No. 8, July 2016, pp. 1160-1181., DOI:10.3390/s16081160