

Motion Simulator for a Large-Scale Textile Plotter

Vesselin Gueorguiev
 Technical University of Sofia
 Sofia, Bulgaria
 veg@tu-sofia.bg

Ivan Evgeniev Ivanov
 Technical University of Sofia
 Sofia, Bulgaria
 iei@tu-sofia.bg

Abstract—This paper presents the design, implementation and results of creation of computerised model of a large-scale textile plotter. The presented simulator has to generate graphics output and to support logs for the controller inputs and outputs (the model outputs and inputs). It was used to tune all software components of the embedded controller of a large-scale textile plotter. Motion machine non-linearities are implemented.

Keywords- motion simulator; X-Y machine; program generation.

I. INTRODUCTION

The presented work describes the design, solutions and results of motion simulator of a large-scale textile plotter. All presented here solutions were successfully implemented. The plotters of this type draw drafts for hand-driven cloth cutting. Required draw accuracy is 0.05 mm corresponds to the cutting accuracy.

Implementation of a control system for a machine of this type and size is impossible without preliminary use of a system model. The described below machine specifics eliminate the possibility to use scientific environment as MATLAB for all design and implementation stages. For design and tuning purposes is implemented very specific parallel connection of both real machine and simulator to the controller.

A specific approach of that type is discussed in [1]. Other analyses and conclusions are presented in [2][3].

At the end of this article is presented the approach of use of a program generating framework (PRGEN) to build both control and simulation systems.

II. SIMULATION OBJECT

The simulator presented here was first designed for the plotter of this type with the following parameters:

- Drawing field of 1000 × 2000 mm length/wide (Fig. 1). Technical movement dimensions are 1100 × 2100 mm. The bridge carries drawing head or cutting devices either.
- Both axes are driven by DC servomotors with permanent magnets. A rotational movement is reduced by gears and transformed to a translational movement of a drawing/cutting head by steel cords. The drawing head is placed on a bridge. The bridge is the dominant load of 10kg in Y direction.
- Technical parameters:

$$\begin{aligned} V_{X_{\max}} &= V_{Y_{\max}} = 10 \text{ m/s}; \\ a_x^+ &= 5 \text{ m/s}^2; a_x^- = 4.5 \text{ m/s}^2; \\ a_y^+ &= 3 \text{ m/s}^2; a_y^- = 2 \text{ m/s}^2. \end{aligned}$$

Machine's axes are not independent. The kinematics scheme and position feedback elements are briefly shown on Fig. 2a.

A rotational movement is reduced by gears and transformed to a translational movement of a drawing head by wire. The wires are wound up on a drum shown on Fig. 2b. The distance $l_1=2100$ mm. The distance l_2 varies because this is the distance to the bridge. The height of the drum is $h_1=150$ mm. The deviation of the wire from the horizontal line generates nonlinearity in angle-to-length transformation. The linear error ϵ is function of two variables – the absolute angle φ of the drum and the coordinates of the drawing head $\langle X, Y \rangle$. The position feedback sensor is incremental sensor, coupled directly to the motor shaft and thus – does not

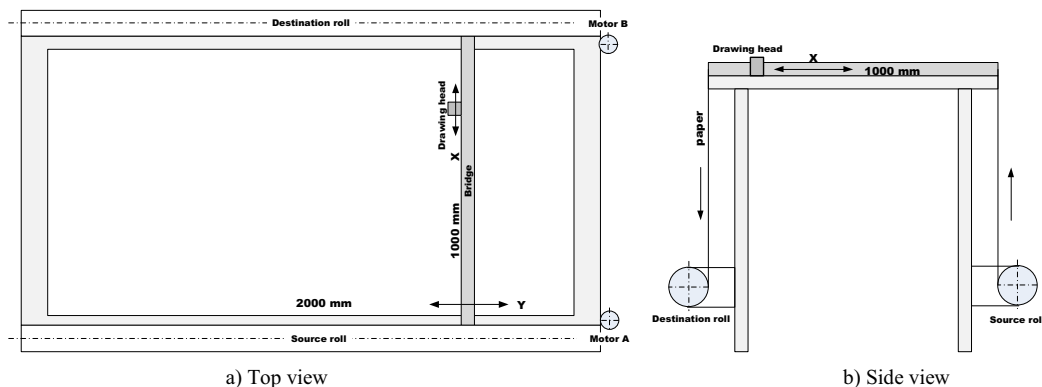


Figure 1. Flat-bed textile plotter

measure exact head position.

III. EASE OF USE

The closed loop kinematics shown on Fig. 2a generates dynamics problems based on the fact that movements on axis X or Y need both motors to work but movement in direction with $\text{tg } \alpha = \pm 1$ engages only one of the motors which has to move both the bridge and the head. Simulation of such a dynamics is rather complicated. The weight of the object to be moved varies also – it is the head only or both head and bridge.

The motor's dynamics is represented with a first order equation:

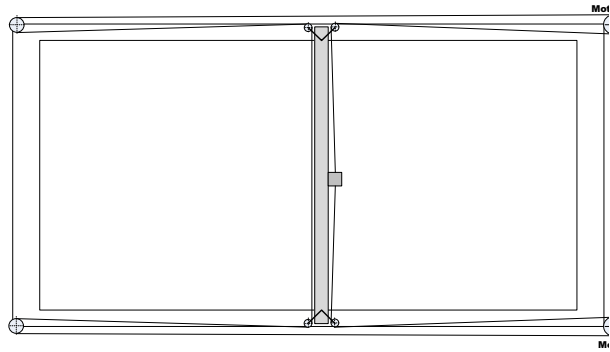


Figure 2a. Kinematics scheme

$$J \frac{d\Omega}{dt} + \frac{K_L \cdot K_r}{R} \cdot \Omega = \frac{K_m \cdot U}{R} - M_c, \quad (1)$$

what is a representation of a first order dynamic object.

Each motor is simulated separately.

The varying element here is the J , representing the moment of inertia. It depends on the angle between motion direction and coordinate axes according the machine kinematics. The exact explanation is out of scope of this material.

The simulator is implemented using the PRGEN program generator [4][5]. The simulator structure consists of several dynamic elements (first order objects) and a set of elements

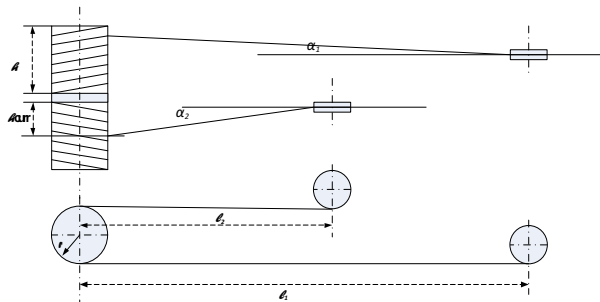


Figure 2b. Wiring scheme

calculating parameterisation of the above elements according kinematic scheme and movement direction.

The output of the dynamic elements is integrated resulting to round angles and to linear movements.

As it was described in the previous section the wired mechanism generates additional linear error dependent on I_1 ,

I_2 and the actual height of the wrapped cylinder (h_1) on Fig. 2b for both upper and lower parts of the drum. Transformation module calculating real coordinates from ideal coordinates is implemented, either.

IV. REAL-TIME IMPLEMENTATION

Simulation procedure is implemented in two different types of complexity. The first one is simple computer simulation. The second is parallel run of the simulator and real machine (see Fig. 3a and 3b). The possibility to run both types of simulation is the reason to avoid usage of MATLAB and SIMULINK for real-time test-bed.

The simulator is implemented on a 32-bit computer able

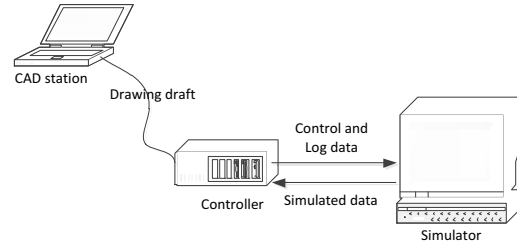


Figure 3a. Simulation environment structure - simple Controller-Simulator connection scheme

to run the HARTEX real-time operating system used for the embedded controller and equipped with a display with the appropriate dimensions and resolution.

The communication between controller and simulator

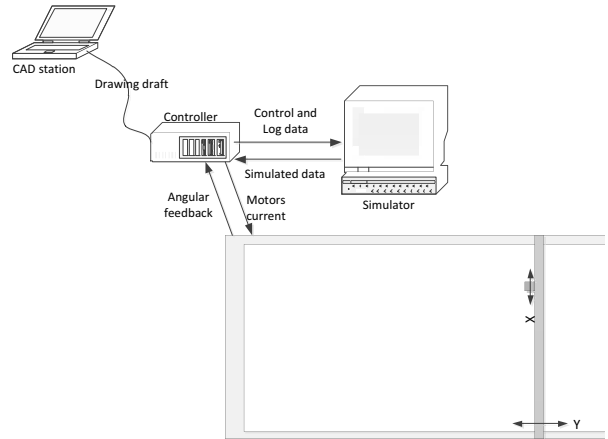


Figure 3b. Simulation environment structure - Controller-Machine-Simulation connection scheme

computers is high speed duplex serial line. It is able to transfer all the information for simulation and for the system log in a single control time interval.

The exchanged information consists of the following data:

- controller-to-simulator: required angular velocity ω for each motor, $\langle X, Y \rangle$ couple from the interpolator
- simulator-to-controller: $\Delta\phi$ (change of the axis angle) for the each motor.

Visualisation module of the simulator draws both required and actual trajectories.

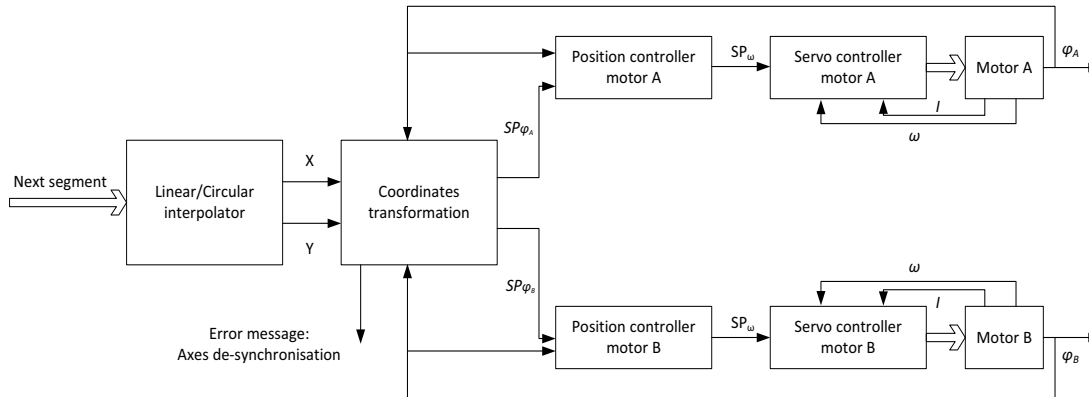


Figure 4. Interpolation and trajectory control – general structure

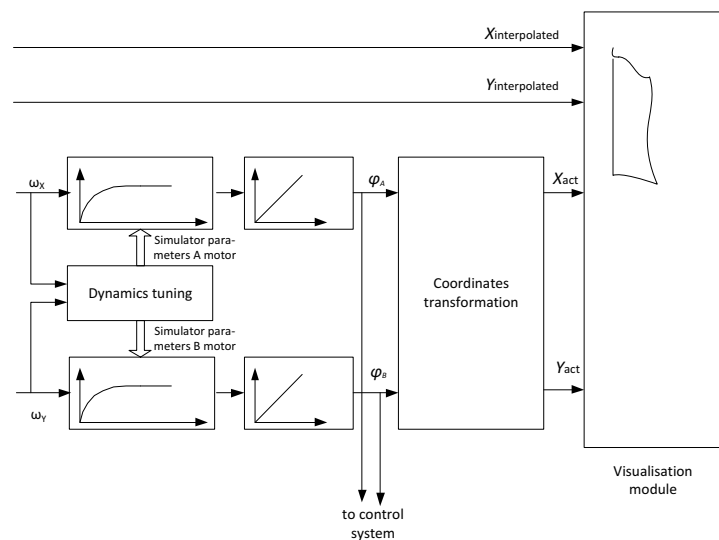


Figure 5. Simulator structure

The general structure of the control system is presented at Fig. 4.

The general structure of the simulator is represented at Fig. 5.

Simulation procedure is event-driven. On every packet received from the controller the simulator calculates all object variables, sends them to the controller (for the next sampling period), visualises movements on the screen and saves log data.

Simulation, visualisation and file operations are implemented as different tasks of the operating system. They are configured using pre-designed modules from the PRGEN modules library.

V. RESULTS AND CONCLUSION

The presented motion simulator was implemented to tune the controller of the presented flat-bed textile plotter. After its successful implementation it was adapted for a number of similar machines. Its internal structure representing machine's kinematics and dynamics was respectively adapted.

The possibility to connect both the real machine and the simulator to the controller and to compare real object's data and simulated behaviour enabled to tune the simulator to represent the machine very exactly.

REFERENCES

- [1] Xie Yue, Vilathgamuwa, M., Tseng, K.J., Nagarajan, N. Modeling and robust adaptive control of a 3-axis motion simulator, Industry Applications Conference, 2001. Thirty-Sixth IAS Annual Meeting. Conference Record of the 2001 IEEE, vol.1, pp. 553 - 560
- [2] O. Benedettini, B. Tjahjono, "Towards an improved tool to facilitate simulation modeling of complex manufacturing systems". 2008, International Journal of Advanced Manufacturing Technology 43 (1/2): 191–9
- [3] Gabriel A. Wainer, Discrete-Event Modeling and Simulation: a practitioner's approach. CRC Press, 2009.
- [4] Emanuil Markov, Ivan Evg. Ivanov, Vesselin Gueorguiev, Program Generator Architecture, DESE 2011, Dubai, UAE, December 2011
- [5] Nikolay Baldzhiev, Velizar Bodurski, V. Gueorguiev, I. E. Ivanov, Implementation of Objects Simulators and Validators using Program Generation Approach, DESE 2011, Dubai, UAE, December 2011