Faster than real-time simulation of mobile crane dynamics using digital twin concept

V Zhidchenko¹², I Malysheva¹, H Handroos¹ and A Kovartsev²

¹Laboratory of Intelligent Machines, Lappeenranta University of Technology, Skinnarilankatu 34, Lappeenranta, Finland, 53850
²Samara National Research University, Moskovskoe shosse 34, Samara, Russia, 443086

e-mail: vzhidchenko@yandex.ru

Abstract. The paper discusses the problem of the mobile crane movement prediction in real-time during the operation of the crane. Dynamic and kinematic models of the crane are created to solve this problem. The models form the digital twin that can be used to facilitate the crane operation. As the crane can move rather fast, the crane dynamics, rather than kinematics, becomes more important in its movement prediction. However, in order to be calculated faster than real-time the crane model should be simplified. The paper considers an example of the mobile crane for which two models are developed: the detailed reference model and the simplified model for faster than real-time calculations, which includes the dynamic and kinematic equations. We implemented both models as C programs. Experiments are made in order to compare the accuracy and the calculation speed of the simplified model with respect to the reference model.

1. Introduction
The Digital Twin concept provides the tool for improving the features of various systems by analyzing the output of the system’s model – digital twin. The use of simulation models is not a new idea but the capability of feeding them with the real time data concurrently with the actual system has recently become a very popular approach. This capability arises on the basis of the advantages of networking and computing speed. The wide and natural adoption of the concept is through the Internet of Things (IoT). Many applications of the digital twin have been recently developed [1]. They can be divided into the two large areas: analysis of very complex systems (like plants or transportation systems) and real-time analysis of relatively small systems (like a vehicle or a human body). The first area deals with a huge number of processes that take place in a large and complicated model. This approach is very close to the well-known simulation modelling. The second approach uses simpler models but often requires real-time interactions. It is sensitive to the networking speed and computational speed [2].

This paper discusses the possibility for usage of digital twin for the making real-time decisions in controlling of mobile machines on the basis of the faster-than-real-time simulation. The problem of controlling mobile machines is considered. The digital twin uses the model of the machine and its
environment to facilitate the control task. For example, the use of the crane or excavator in a limited
environment carries a risk of collisions. The digital twin can help the operator to avoid them by fast
trajectory computation based on the control input made by the operator. The less experienced is the
operator and the more responsible is the work or environment the more useful is the help.

Much research has been already done about the operation of cranes. When dealing with collision
prevention only the kinematics of the crane is usually considered [3]. The reason is that collision
prevention is especially important on large construction sites where the loads of high dimensions and
weight are operated and where many people and technical devices work. The cranes being used in
such cases have large inertia and large response times and move the loads to long distances. That is
why the kinematics of the crane substantially determines its trajectory and dynamics is of small
importance. In contrast, when small mobile cranes are considered the distances are short and are
comparable with the size of the load. In such situation the dynamics of the crane play more significant
role as the response time is small and the movement can be fast. It is easier for the operator to make a
mistake with short distances and fast movement. The consequences of collisions in small cranes
operation are not so important but we consider this problem because it gives the working example for
investigation of faster than real-time calculations. The results could also be used in other applications
like cars collision prevention or remote control of robot actuators.

As an example, in this work, the mobile crane PATU 655 is considered. This crane is hydraulically
actuated, and has the maximum load of 850 kg and 500 kg at the maximum boom extension. The crane
consists of the following parts: slew mechanism, pillar, lifting boom, system of four interconnected
side links, outer boom, and extension boom. The crane is actuated through five hydraulic cylinders of
different dimensions. The first two are included in double cylinder rack and pinion actuator of slew
mechanism. The mechanism provides the crane rotation around the vertical. The next two cylinders
operate through the joints the lifting and outer booms. The last cylinder is mounted inside the outer
boom. It controls the motion of the extension boom along outer boom.

In the work, the two dynamic models of the example crane are developed. The first one is created
using commercially available software such as MATLAB/Simulink. The model captures in details the
structure of the real-life crane. This model is used as a reference for accuracy estimation of the second
model that is intended for faster than real-time calculations. Moreover, it provides the first
approximation for the time needed for such model calculation. The second model is built from scratch
using dynamic equations and is aimed to be calculated by software specially created for this task. The
first goal of the research is to find the level of abstraction of the crane model suitable for improving
the calculation speed while retaining the accuracy. The second goal is to investigate the ways of
development and execution of the simulation program using different software and hardware platforms
to run it faster than real time.

2. Reference model

For the research purposes, a detailed reference model of mobile crane is developed (figure 1). The
model captures the dynamics of the crane and is created using visual programming language
MATLAB/Simulink including the features provided by Simscape Multibody. Figure 2 shows the view
of developed Simulink model. Here the constructive parts of the crane are modeled as separate blocks:
pillar, lifting boom, system of four side links, outer boom, extension boom, and four hydraulic
actuators including one in slew mechanism. Block port names depict the parts they are connected to
either directly or via the joints. All the parts are interconnected either with revolute or prismatic joints
as shown in figure 2. The inertia properties of the crane parts modeled in the blocks are imported from
CAD drawings of the example crane developed in [4]. The slew mechanism modelled as double
cylinder rack and pinion actuator using two Rack and Pinion Constraint blocks. In order to make two
models comparable, a number of assumptions are made. The constituting parts of the crane are
modelled as rigid bodies and the friction forces in the joints are neglected. The dynamics of the slew
mechanism is neglected. The load mass is located at the end of the extension boom and is rigidly
fixed.
The actuation of the model is implemented by applying cylinder forces to the translational joints that represent hydraulic cylinders. The pressure in the cylinder in plus and minus chambers and the geometry of the cylinder are used as the input parameters to calculate the cylinder force:

\[ F_c = p_1 A_1 - p_2 A_2 - F_\mu \]  

where \( p_1, p_2 \), and \( A_1, A_2 \) are the pressures and the cross-section areas of the piston in plus and minus chamber respectively, \( F_\mu \) is the friction force. In equation (1) the dimensions of hydraulic cylinders correspond to those of the example crane.

The case when the cylinder is in a fixed position corresponds to fully closed directional control valve and compressible hydraulic fluid. This kind of states can be modelled using forces provided by equivalent hydraulic springs. A stiffness of such a spring for each cylinder is estimated based on the following equation [5]:

\[ C_{eq} = B_{oil}(A_1^2/(V_1 + V_{1d}) + A_2^2/(V_2 + V_{2d})) \]

where \( B_{oil} \) is effective bulk modulus of hydraulic fluid, \( A_1 \) and \( A_2 \) are the effective cross-section areas of the piston in plus and minus chamber respectively, \( V_1 \) and \( V_2 \) are the dynamic volumes of these chambers depending on the piston position, \( V_{1d} \) and \( V_{2d} \) are the dead volumes of the fluid lines connected to the plus and minus chamber respectively.

**Figure 1.** Structural view of the reference model in Simscape Multibody Mechanics Explorer.

**Figure 2.** Block diagram of the reference model.

Hydraulic spring damping for each cylinder is chosen in a range 5000…10000 Ns/m. The spring parameters can be considered constant as the cylinders are assumed to perform only small motions around their fixed positions when the control valve is closed. The output of the model is presented as coordinates of the crane tip position with respect to the global coordinate system.
After the described reference model had been fully built in Simulink, the C source code was generated using MATLAB Embedded Coder. Such a code is independent and can be compiled and run outside MATLAB. However, it is not intended for modification. The generated code is optimized for the execution efficiency. The code has input points for hydraulic pressure assignment and enables logging to the file of the crane tip location coordinates.

3. Mobile crane model for faster than real-time calculations

Using the same assumptions as in the reference model, the model with the higher level of abstraction is developed that should be calculated faster but with the accuracy enough for decision making in the task of controlling the crane.

The model of the crane consists of two parts: the dynamic model and the kinematic model. The kinematic model is quite simple and utilises the techniques traditionally used in multibody simulation[6]. The crane is represented as a set of rigid bodies interconnected through the revolute and prismatic joints. The numbering of the joints is indicated in the figure 3.

![Figure 3. The numbering of the joints of the crane used in kinematic model.](image)

The global origin is located at the base of the crane in the joint 0. Every point of the crane in a global space is represented by a 3x1 position vector \( r = [x, y, z]^T \). The origins of the local coordinate systems associated with the crane’s booms are located in the points representing the joints between the booms. For example, the origin of the first boom’s coordinate system is located in the joint 1. The numbering of joints is used to enumerate the local coordinate systems of the booms. Therefore, for the points of the first boom the local coordinate system associated with the joint 1 is used.

The orientation of the body relative to some coordinate system is defined with 3x3 rotation matrix \( A \):

\[
A = \begin{bmatrix}
    r_{11} & r_{12} & r_{13} \\
    r_{21} & r_{22} & r_{23} \\
    r_{31} & r_{32} & r_{33}
\end{bmatrix}
\]  

(3)

where \( r_{ij} \) is the cosine of the angle between the unit vector number \( j \) of the reference frame associated with the body and the unit vector number \( i \) of the coordinate system in which the orientation is defined (direction cosine). For the revolute joints that are located between the booms the rotation matrix will be defined as follows:

\[
A = \begin{bmatrix}
    \cos(\theta) & -\sin(\theta) & 0 \\
    \sin(\theta) & \cos(\theta) & 0 \\
    0 & 0 & 1
\end{bmatrix}
\]  

(4)

where \( \theta \) is the angle between the booms.
The position of any point of the boom number \( j \) can be represented in the local coordinate system associated with the joint number \( i \) as follows:

\[
r^{j} = R^{j} + A^{j}u^{j}
\]  

(5)

where \( r^{j} \) is the position vector of the point relative to the joint \( i \), \( R^{j} \) is the position vector of the origin of the local coordinate system associated with the joint \( j \), relative to the joint \( i \), \( A^{j} \) is the rotation matrix of boom \( j \) relative to boom \( i \), and \( u^{j} \) is the position vector of the point relative to the joint \( j \).

Translation and rotation together can be represented by a 4x4 transformation matrix \( T_{ij} \):

\[
T_{ij} = \begin{bmatrix}
A^{j} & R^{j} \\
0 & 1
\end{bmatrix}
\]  

(6)

Using the transformation matrix the position vector of the point can be represented as follows:

\[
r^{j}_4 = T_{ij}u^{j}_4
\]  

(7)

where \( r^{j}_4 = [r^{j}_x, r^{j}_y, r^{j}_z, 1]^T \) and \( u^{j}_4 = [u^{j}_x, u^{j}_y, u^{j}_z, 1]^T \).

The notation described above allows us to define the transformation matrices between the coordinate systems associated with the joints:

\[
T_{01} = \begin{bmatrix}
\cos(\theta_1) & -\sin(\theta_1) & 0 & R^{p1}_x \\
\sin(\theta_1) & \cos(\theta_1) & 0 & R^{p1}_y \\
0 & 0 & 1 & R^{p1}_z \\
0 & 0 & 0 & 1
\end{bmatrix},
\]

\[
T_{12} = \begin{bmatrix}
\cos(\theta_2) & -\sin(\theta_2) & 0 & r^{lp2}_x \\
\sin(\theta_2) & \cos(\theta_2) & 0 & r^{lp2}_y \\
0 & 0 & 1 & r^{lp2}_z \\
0 & 0 & 0 & 1
\end{bmatrix},
\]

\[
T_{24} = \begin{bmatrix}
1 & 0 & 0 & r^{lp4}_x \\
0 & 1 & 0 & r^{lp4}_y \\
0 & 0 & 1 & r^{lp4}_z \\
0 & 0 & 0 & 1
\end{bmatrix}
\]  

(8)

where \( R^{p1} = [R^{p1}_x, R^{p1}_y, R^{p1}_z] \) is the global position vector of the joint 1, \( r^{lp2} = [r^{lp2}_x, r^{lp2}_y, r^{lp2}_z] \) is the position vector of the joint 2 in the coordinate system associated with the joint 1 and \( r^{lp4} = [r^{lp4}_x, r^{lp4}_y, r^{lp4}_z] \) is the position vector of the joint 4 in the coordinate system associated with the joint 2.

Using these matrices it is easy to calculate the global position vector of any point of the crane. For example, the global position vectors of the joint 2 and joint 4 are calculated as follows:

\[
R^{lp2} = T_{01} \cdot T_{12} \cdot \bar{Z},
\]

\[
R^{lp4} = T_{01} \cdot T_{12} \cdot T_{24} \cdot \bar{Z}
\]  

(9)

where \( \bar{Z} = [0,0,0,1]^T \).

The angle \( \theta_1 \) can be calculated from the stroke of the first cylinder:

\[
\theta_1 = \epsilon_1 + a \cos \left( \frac{L_{16}^2 + L_{17}^2 - s_1^2}{2 \cdot L_{16} \cdot L_{17}} \right) + \epsilon_2 - \frac{\pi}{2}
\]  

(10)

where \( \epsilon_1 \) is the constant angle between the axis OY and the line from joint 6 to joint 1, \( L_{16} \) and \( L_{17} \) are the distances between the joint 1 and joints 6 and 7, \( s_1 \) is the stroke of the cylinder 1 and \( \epsilon_2 \) is the angle between the line from joint 7 to joint 1 and the axis OX in the position corresponding to such cylinder stroke \( s_1 \) that the first boom is horizontal.

The angle \( \theta_2 \) is calculated from the stroke of the second cylinder and four-bar mechanism that moves the second boom:
\theta_2 = 2 \cdot \tan \left(\frac{-B - \sqrt{B^2 - 4 \cdot A \cdot C}}{2 \cdot A}\right) - (\pi - \theta_{128} - \theta_{235}), \quad (11)

\begin{align*}
A &= \cos(\theta_{2810}) - \frac{L_{82}}{L_{810}} - \frac{L_{82} \cdot \cos(\theta_{2810})}{L_{32}} + \frac{L_{810}^2 - L_{103}^2 + L_{32}^2 + L_{82}^2}{2 \cdot L_{810} \cdot L_{32}} \\
B &= -2 \cdot \sin(\theta_{2810}) \\
C &= \frac{L_{82}}{L_{810}} - \left(\frac{L_{82}}{L_{32}} + 1\right) \cdot \cos(\theta_{2810}) + \frac{L_{810}^2 - L_{103}^2 + L_{32}^2 + L_{82}^2}{2 \cdot L_{810} \cdot L_{32}} \\
\theta_{2810} &= \pi - a \sin \left(\frac{Y_{89}}{L_{89}}\right) - a \cos \left(\frac{L_{89} + L_{810}^2 - S_2^2}{2 \cdot L_{89} \cdot L_{810}}\right) + a \sin \left(\frac{Y_{28}}{L_{28}}\right)
\end{align*}

In expression (11), $L_{NM}$ is the length of the line segment between joint N and joint M, $Y_{NM}$ is the projection of this line segment to the axis $OY$ in the coordinate system associated with the joint 1, $s_2$ is the stroke of the second cylinder. The equations mentioned above could be simplified but they are given in that form for better comprehension of their derivation.

The dynamic model of the crane is built using the well-known iterative Newton-Euler dynamic formulation [7]. This formulation is used for modelling dynamics of multiple interconnected bodies that have common joints and form the serial chain like a robotic arm does. Iterative Newton-Euler dynamic formulation has $O(N)$ complexity and is one of the fastest and most popular formulations for robot dynamics. Review of different dynamic formulations and their computational efficiency can be found in [8]. Iterative Newton-Euler formulation solves inverse dynamics problem computing joint torques or forces that are required to produce the known manipulator’s trajectory (position, velocity, and acceleration). It provides the way to acquire equations for forces or torques in joints in the form:

$$
F = \tau(\theta, \dot{\theta}, \ddot{\theta})
$$

where $T = [\tau_1, \tau_2, ..., \tau_N]^T$ is the vector representing the torques or forces in joints ($\tau_i$ is the torque for revolute joint number $i$ or the force for prismatic joint), $\theta = [\theta_1, \theta_2, ..., \theta_N]^T$ is the vector of independent coordinates that are represented by angle of rotation for revolute joints and by displacement for prismatic joints.

In this work, iterative Newton-Euler dynamic formulation is used for derivation of equations for forward dynamics. The crane booms are considered as bodies connected with each other by revolute or prismatic joints. In the system of equations (12) $F$ is the vector of linear functions of accelerations $\ddot{\theta}_1$, so the system can be solved for the vector $\dot{\theta} = [\dot{\theta}_1, \dot{\theta}_2, ..., \dot{\theta}_N]^T$. This gives the system of differential equations of independent coordinates. The system is solved by the Runge-Kutta fourth-order method with the following initial conditions:

$$
\dot{\theta}_i(t_0) = \ddot{\theta}_i(t_0) = 0
\quad \theta_i(t_0) = f_i(s_i), \quad i = 1, N
$$

where $f_i(s_i)$ are the functions that calculate initial values for independent coordinates in each joint. Equations (10), (11) define these functions for the angles of booms from the initial strokes of the cylinders.

In the case when the pillar is not rotating the problem can be assumed planar and the differential equations for the angles $\theta_1$ and $\theta_2$ are defined as follows:
\[ \dot{\theta}_2 = \left[ \tau_2 - \dot{\theta}_1 \cdot D - P_{2cm}^x \cdot m_2 \cdot B - P_{2cm}^y \cdot m_2 \cdot C - m_4 \cdot g \cdot \left( c_2 \cdot r_{lp2}^y + s_2 \cdot r_{lp2}^x \right) \right] \left( I_2^z + m_2 \cdot \left( (P_{2cm}^x)^2 + (P_{2cm}^y)^2 \right) \right)^{-1} \]

\[ \dot{\theta}_1 = \left[ \tau_1 - A \left( \tau_2 - P_{2cm}^x \cdot m_2 \cdot B - P_{2cm}^y \cdot m_2 \cdot C - m_4 \cdot g \cdot \left( c_2 \cdot r_{lp2}^y + s_2 \cdot r_{lp2}^x \right) \right) \right] \left( I_2^z + m_2 \cdot \left( (P_{2cm}^x)^2 + (P_{2cm}^y)^2 \right) \right)^{-1} \]

\[- \tau_2 + m_4 g \left( c_1 \cdot P_{1cm}^x + s_1 \cdot P_{1cm}^y \right) + m_2 \left( s_2 \cdot r_{lp2}^y - c_2 \cdot r_{lp2}^x \right) \right] \left( \dot{\theta}_1 + \dot{\theta}_2 \right) \left( P_{2cm}^x \right)^2 + m_2 \cdot \left( c_1 \cdot r_{lp2}^y + s_1 \cdot r_{lp2}^x \right) \right] - \left( \dot{\theta}_1 + \dot{\theta}_2 \right) \left( c_2 \cdot r_{lp2}^y + s_2 \cdot r_{lp2}^x \right) + m_2 \left( c_1 \cdot r_{lp2}^y + s_1 \cdot r_{lp2}^x \right) \left( \dot{\theta}_1 + \dot{\theta}_2 \right) \left( c_2 \cdot r_{lp2}^y + s_2 \cdot r_{lp2}^x \right) + g \left( s_2 \cdot s_1 - c_2 \cdot c_1 \right) \right] \]

\[ \left[ M - A \cdot D \cdot \left( I_2^z + m_2 \cdot \left( (P_{2cm}^x)^2 + (P_{2cm}^y)^2 \right) \right)^{-1} \right] \]

where

\[ A = P_{2cm}^x \cdot m_2 \cdot \left( c_1 \cdot r_{lp2}^y - s_1 \cdot r_{lp2}^x \right) + P_{2cm}^y \cdot m_2 \cdot \left( c_1 \cdot r_{lp2}^x + s_1 \cdot r_{lp2}^y \right) \]

\[ B = \left( \dot{\theta}_1 \right)^2 \cdot \left( s_2 \cdot r_{lp2}^y - c_2 \cdot r_{lp2}^x \right) - g \left( s_2 \cdot s_1 - c_2 \cdot c_1 \right) \]

\[ C = \left( \dot{\theta}_1 \right)^2 \cdot \left( s_2 \cdot r_{lp2}^y + c_2 \cdot r_{lp2}^x \right) - g \left( s_2 \cdot s_1 + c_2 \cdot c_1 \right) \]

\[ D = I_2^z + m_2 \cdot \left( P_{2cm}^x \right)^2 + \left( P_{2cm}^y \right)^2 \right) + P_{2cm}^x \cdot m_2 \cdot \left( s_2 \cdot r_{lp2}^y + c_2 \cdot r_{lp2}^x \right) + P_{2cm}^y \cdot m_2 \cdot \left( c_2 \cdot r_{lp2}^y - s_2 \cdot r_{lp2}^x \right) \]

\[ M = I_2^z + m_2 \cdot \left( (P_{2cm}^x)^2 + (P_{2cm}^y)^2 \right) \right) - r_{lp2}^y \cdot m_2 \cdot \left( s_1 \cdot P_{2cm}^y + s_1 \cdot c_2 \cdot r_{lp2}^x - s_1 \cdot s_2 \cdot r_{lp2}^y \right) \]

\[ + r_{lp2}^y \cdot m_2 \cdot \left( c_1 \cdot P_{2cm}^y + c_1 \cdot c_2 \cdot r_{lp2}^x + c_1 \cdot s_2 \cdot r_{lp2}^y \right) + m_2 \cdot \left( c_1 \cdot r_{lp2}^y + s_1 \cdot r_{lp2}^x \right) \]

\[ \left( P_{2cm}^x + s_1 \cdot r_{lp2}^y + c_2 \cdot r_{lp2}^x \right) \]

\[ I_N \] is the inertia tensor of the boom \( N \) and \( I_N^z = I_N (3;3) \), \( m_1 \) is the mass of the boom 1, \( m_2 \) is the mass of the boom 2, \( m_4 \) is the mass of the load, \( c_1 = \cos(\theta_1) \), \( s_1 = \sin(\theta_1) \), \( c_2 = \cos(\theta_2) \), \( s_2 = \sin(\theta_2) \), \( P_{2cm}^x \) and \( P_{2cm}^y \) are projections of the center of mass of the boom \( N \) to the axes \( OX \) and \( OY \) in the coordinate system associated with joint \( N \)-1, \( g \) is the gravity. Inertia tensors and coordinates of the centers of mass in the local coordinate systems of the booms are taken as constants from the reference model. \( \tau_1 \) and \( \tau_2 \) are the torques being applied to the joint 1 and joint 2 respectively. They are calculated from the forces produced by the cylinders.

### 4. Experimental results and discussion

This section presents the results of simulation with the faster-than-real-time (FTRT) model described above compared against the results of reference model. The mass of the load being simulated is 250 kg. The simulated time period is 5 s. In the first experiment, all cylinders were fixed in the middle position. The fixed position of the cylinders was simulated in the FTRT model by the spring force caused by compression of hydraulic fluid in each cylinder and described in section 2. The pillar was rotated with the constant torque 1000 Nm applied to it.

Figure 4 (a), (b), (c) shows the \( X \), \( Y \) and \( Z \) coordinates of the boom tip in the global coordinate frame with respect to time in the first experiment. Figure 4 (d) shows the trajectory of the boom tip in axonometric projection. The variation of the \( Y \) position is caused by the spring force acting in the cylinders.

Figure 5 shows the relative difference between the tip coordinates calculated with two models. The relative difference \( d \) is calculated as the absolute difference between the tip position vectors obtained from two models divided by the length of the position vector calculated by the reference model:
In the second experiment, the cylinder of the first boom provided the constant force 62000 N in addition to the rotation of the pillar caused by the constant torque 1000 Nm applied to it. Figure 6 (a), (b), (c) shows the $X$, $Y$ and $Z$ coordinates of the boom tip in the global coordinate frame with respect to time in the second experiment. Figure 6 (d) displays the trajectory of the boom tip in axonometric projection. Figure 7 shows the relative difference between the tip coordinates calculated with different models in the second experiment. The relative difference is calculated in accordance with the equation (15).
Analysing obtained results in both experiments it can be concluded that the relative difference between the tip coordinates increases with time and it does not exceed 2.5% for the simulated time period of 5 seconds. Since the models being discussed are intended for real-time simulations of shorter time periods (2-3 seconds), the period of 5 seconds can be considered as sufficient for error estimation. The absolute difference that corresponds to the maximum relative difference between the tip coordinates is 0.12 m that is close to the deviation of the tip coordinates due to fluid compression inside the cylinders. Therefore, the relative difference of 2.5% can be considered as acceptable for the prediction of tip location.

Figure 8 shows the duration of execution for the programs that implement the discussed models. For the simulated time period of 5 seconds the program execution that takes less than 5 seconds can be considered as faster than real-time. Since it is assumed that the simulation outcome is used by some interpretation algorithm (for example, for the purposes of predicted movement display, collision...
detection, etc.) the overall success of this approach depends on the sum of execution times of both the simulation program and the program implementing the interpretation algorithm. This work does not consider any particular interpretation of the simulation results. Its main purpose is a minimization of execution time of simulation program. The best solution should execute faster than real time on any hardware platform.

The program implementing the reference model was compiled from the C source code generated by MATLAB Embedded Coder. FTRT model was implemented by the program compiled from the manually created C source. The programs were executed on the desktop personal computer with Intel Core i5-4590 3.30 GHz processor. Both programs run faster than real time but the program implementing FTRT model runs 6.7 times faster than the program generated from MATLAB model.

![Figure 8. Duration of execution for the programs that implement different models.](image)

The program implementing FTRT model was also compiled and executed on the Orange Pi 2G-IOT single-board computer with ARM Cortex-A5 1GHz CPU and 256Mb of RAM. This computer is compatible with Raspberry Pi platform that is widely used in IoT applications. At the time of the paper’s writing the majority of Raspberry Pi compatible computers available on the market have CPU with higher frequency and larger amounts of memory. Thus, the tested computer can be considered as relatively slow among the similar systems. The program implementing FTRT model executes faster than real-time on the Raspberry Pi platform consuming less than a half of real time for calculations.

5. Conclusions and future work

Experimental results show that the proposed simplified model can estimate the movement of the crane. The accuracy of movement prediction depends on the complexity of the model. The simplifications introduced in the model provide the accuracy that is enough for the boom’s trajectory prediction for real-time decision making support. The simplified model can be applied for a wide range of hydraulically actuated mobile cranes.

The proposed simplified model allows faster than real time simulation of crane dynamics even on low cost and low power single-board computers that are commonly used in IoT. Based on the obtained results it can be concluded that the developed model can be used as a basis for a Digital Twin that can be run on-board of mobile crane. Such Digital Twin is suitable for the real-time crane’s trajectory prediction as well as for calculation and transfer of data about crane dynamics in IoT applications.

In the research, the pressure values in the cylinders were used as the input parameters for the forces and torques calculation. It is also possible to consider the input voltage of hydraulic directional control valves that is proportional to the signal from the control device used by the operator. In such a case, the hydraulic system should be modelled and included in simulation. The time needed for this simulation increases the total execution time but the benefits are that dynamics of hydraulic system is taken into account and that movement and fixed position of hydraulic cylinders are simulated in a common way. Moreover, it is essential to perform experiments with the real-life cranes in order to find
the most suitable input parameters to minimize the time of calculations and maximize the accuracy of simulation.

6. References
[1] Grievs M and Vickers J 2017 Digital twin: mitigating unpredictable, undesirable emergent behavior in complex systems Transdisciplinary Perspectives on Complex Systems (Cham: Springer) 85-113
[2] Hu P 2017 Survey on fog computing: architecture, key technologies, applications and open issues J. of Netw. and Comput. Appl. 98 27-42
[3] Zhang C 2012 Improving lifting motion planning and re-planning of cranes with consideration for safety and efficiency Adv. Eng. Inform. 26(2) 396-410
[4] Luostarinen L, Åman R and Handroos H 2014 Development of control interface for HIL simulation of electro-hydraulic energy converter Int. Rev. on Model. and Simul. IREMOS 7(4) 653-660
[5] Khalil M k B 2009 Interactive analysis of closed loop electro-hydraulic control systems 13th Int. Conf. on Aerospace Sciences & Aviation Tech. ASAT-13-HC-01
[6] Craig J J 2005 Introduction to Robotics: Mechanics and Control (N.J.: Pearson)
[7] Luh J Y S, Walker M W and Paul R P 1980 On-line computational scheme for mechanical manipulators ASME. J. Dyn. Sys., Meas., Control 102(2) 69-76
[8] Featherstone R and Orin D 2000 Robot dynamics: equations and algorithms Proc. IEEE Int. Conf. Robotics & Automation ICRA 826-834

Acknowledgments
This work was partially funded by the Russian Federation Ministry of Education and Science and Russian Foundation of Basic Research. Grant #16-41-630637.