

# HUMAN-INSPIRED JUMPING ROBOT - DEVELOPMENT AND CONTROL OF DYNAMIC MODEL

**Damir Omrčen, Jan Babič**

Jožef Stefan Institute  
1000 Ljubljana, Jamova 39, Slovenia  
*damir.omrcen@ijs.si (Damir Omrčen)*

## **Abstract**

In the paper we describe the complete design process of a real robotic system inspired by the anatomic properties of the human body. At the beginning we describe the starting points and the requirements that we wanted to achieve. Then we describe the CAD model of the robot and the construction of the real robotic system. The CAD model data were compared to the data of the manufactured robot parts and the final mechanical properties of the robot were derived.

In order to simulate and to control the real robot we have set the dynamic and the kinematic model of the robot. To obtain dynamic equations of the system we have used SD/FAST tool for modeling of mechanical systems, which is very computationally efficient and provides dynamic equations of the system. Since the robot behaves differently when it touches the ground and when it is in the air, we have derived two models. One for the cases, when the robot is in the air and has six DOF and one for the cases when the robot is on the ground and has four DOF. Using dynamic equations that we obtained using SD/FAST we have developed Simulink blocks for direct and inverse dynamics for both model types.

Afterwards, we have shown in the paper the development of the dynamic model of robotic systems with some uncontrollable joints, these are the joints, where it is not possible to apply any torque by a motor, e.g. in the x and y direction when the robot is in the air.

The derived dynamic model(s) are used for simulation of the jump and for the control of the real robot. In the last section of the paper, the controller of the robot is described and the results of the vertical jumps are presented.

**Keywords: robot modeling, dynamic model, robot control, human-inspired robot, control of balance and stability**

## **Presenting Author's biography**

Damir Omrčen received his B.Sc. degree in automatics from the University of Ljubljana, Ljubljana, Slovenia, in 2000. From the same University he received in 2005 the Ph.D. degree in electrical science. He is currently a research assistant at the Jozef Stefan Institute, Department for Automatics, Biocybernetics and Robotics. His research interests include robot control, modeling and simulation, especially in bio and mobile robotics, redundant systems, mobile manipulators and robot assistants.



## 1 General

The jump is an example of a fast explosive movement that requires quick and completely harmonized coordination of all segments of the robot, for the push-off, for the flight and, finally, for the landing.

In the past, several research groups developed and studied jumping robots but most of these were simple mechanisms not similar to humans. Probably the best known hopping robots were designed by Raibert and his team [1]. They developed different hopping robots, all with telescopic legs and with a steady-state control algorithm. Later, De Man et al. developed a trajectory generation strategy based on the angular momentum theorem which was implemented on a model with articulated legs [2].

The purpose of our research is to propose a new human inspired structure of the lower extremity mechanism by which humanoid robots will be able to efficiently perform fast movements such as running and jumping. For this purpose we built an efficient dynamic model of the humanoid robot and designed and built a human inspired half sized humanoid robotic mechanism that is capable of performing vertical jump and long jump.

In this article we describe the complete process of the modeling a jumping robot system. The dynamic model is acquired using SD/FAST modeling tool. We show in details the development of the dynamic models of robotic systems that have some uncontrollable joints. Besides, we describe the controller of the robot and show the results of the vertical jump performed in the simulation environment.

## 2 Real jumping robot system

### 2.1 Design of the robot

We have designed the robot using I-DEAS CAD modeling tool. The I-DEAS model of the jumping robot is shown in Fig. 1. The model of the jumping robot is planar and is composed of four segments which represent foot, shank, thigh and trunk (see Fig. 1). The segments are connected by rotational hinges whose axes are perpendicular to the sagittal plane. As we can see, the structure of the robot and the link lengths are approximation of a human's structure. The robot has three electromotors, which actuate ankle, knee and hip joint. One uncontrolled rotational joint is at the foot end, where foot touches the ground. Additionally, one passive joint is in the toe joint and is actuated by a spring, however, this DOF will be neglected in modeling.

To achieve a jump of the robot besides suitable control algorithms very high torques in the robot's joint are required in addition to low mass of the whole system. To achieve high torque with a lightweight electromotor high gear ratio is usually necessary. However, high gear ratio is not desired, since we want

to maintain low friction in the joint and the ability to move the joint by an external force easily (backdrivable joint). An alternative is to use a motor with low gear ratio and overload the motor. This allows us to get a sufficient torque for a shorter time.

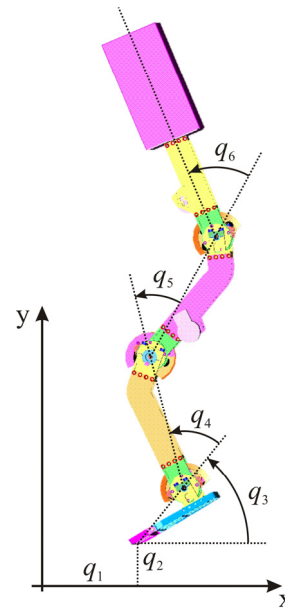


Fig. 1 I-DEAS model of the robot

Since we want to perform a jump, which usually takes a very short time, the overloading is acceptable. We used Maxon RE 40 DC motors with stall torque 2.500 Nm and weight of only 480 g. In our case the gear ratio is 8, which ensures low friction and backdrivability of the joint. Using this motor/gear combination the joint torque is theoretically 20 Nm, which is, based on simulations, sufficient for performing a jump.

Motors are mounted in the shank, thigh and trunk segments. However, the most of the robot weight is situated in the last-trunk segment. In this segment besides a motor a computer, a motion controller and a motor's power amplifiers are situated. In the same way the weight in humans is arranged, i.e. heavy trunk and light legs. From the control part of view it is not desired to have most of the weight in the last segment, since the system act as an inverse triple pendulum. However, it can be shown that this kind of weight distribution can be useful to improve efficiency of the jump and many other dynamic movements.

### 2.2 CAD model

The real robot system was designed considering I-DEAS CAD modeling software. The real system is shown in the right side of Fig. 2. We have used aluminium for all robot parts, only the gears were made of titanium, due to the high torques in the gears.

To control a robot we have to determine the mathematical dynamic model of the robot. We have approximated the robot by an articulated system of four rigid bodies connected by three joints. To define

the dynamic model of the robot corresponding properties of all links has to be known.

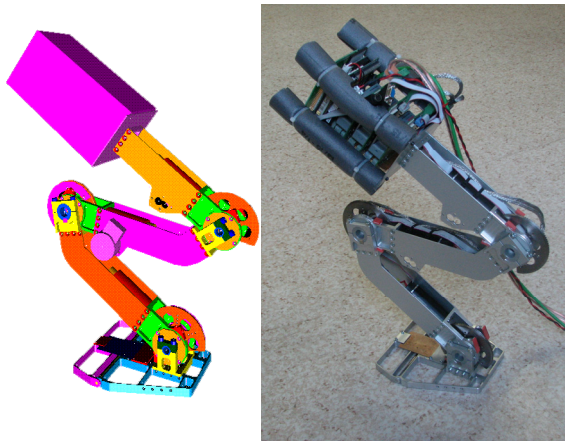


Fig. 2 Model and real jumping robot

Using I-DEAS we have defined lengths and mass properties (i.e. mass, position of the center of gravity COG and inertia tensor) of each robot part and also of each segment. The weights from the model and the weights of the manufactured robot parts were compared and the modifications of the mass properties were made to increase the accuracy of the model.

The final properties of all four links including motors, computer, power amplifiers, wiring, etc. are shown in Tab.1:

Tab.1 Properties of the robot links

link	mass/ kg	length/ mm	pos. of COG/mm	inertia/ kg mm <sup>2</sup>
foot	0.399	93.5	70.7, -3.4	1057
shank	0.963	255	127.8, 10.5	7487
thigh	1.033	254	119.2, -12.0	7440
trunk	2.031	370	186.4, -12.0	20366

### 2.3 Dynamic model of the robot

The dynamic model of the robot will be used in the simulation of the robot system and also in the control of the simulated and the real system.

The actual model is a combination of two models. The first model is when the robot is in the air and the second model is when the robot touches the ground with the tip of the foot. The point of the contact is modeled as a rotational hinge joint. We did not model other contacts of the robot with the ground. The model on the ground has, therefore, four degrees of freedom (DOF) during stance (with the assumption that the foot tip of the robot does not slip and does not bounce back). Additionally, the robot has two more DOFs in the air (i.e. six DOFs during flight). The generalized coordinates used to describe the motion of the robot are denoted by  $q_i$  (see Fig. 1).

The mathematical dynamic model of the robot is given in the following form:

$$\mathbf{H}(q)\ddot{q} + \mathbf{C}(q, \dot{q}) + \mathbf{g}(q) = \boldsymbol{\tau}, \quad (1)$$

here,  $\mathbf{H}$ ,  $\mathbf{C}$  and  $\mathbf{g}$  denote the inertia matrix, the vector of Coriolis and centrifugal forces and the vector of gravity forces, respectively.  $\boldsymbol{\tau}$  is the vector of joint torques and  $q$  is the vector of joint positions.

There are many tools to define the dynamic model of the robot. A raw comparison of different tools used in robotics was done by Žlajpah [3]. The dynamic modeling tool should be numerically very efficient, since the control algorithm will run in the real time on the real robotic system. Additionally, we should be able to transfer the result of the modeling to different computer platforms, since we are very limited in the selection of the robot operation system.

Based on these requests and based on the comparison shown in [3] we have chosen SD/FAST modeling tool (see <http://www.sdfast.com/>).

SD/FAST provides physically-based simulation of mechanical systems by taking a short description of an articulated system of rigid bodies (bodies connected by joints) and deriving the full nonlinear equations of motion for that system. The equations are then output as C or Fortran source code, which can be compiled and linked into many simulation or animation environments. The symbolic derivation of the equations provides the fastest possible simulations. Many substantial systems can be executed in real time on modest computers. SD/FAST is free for an evaluation period.

We have compared few packages for dynamic simulation regarding numerical efficiency. The comparison was done for the planar robotic system with 2, 3, ... 10 DOFs. Fig. 3 shows the comparison of Robotics toolbox [4], Planar Manipulator Toolbox PMT [5], SD/FAST and Matlab SimMechanics. As we can see SD/FAST is some order faster than the other packages, since it generates code of symbolic equations.

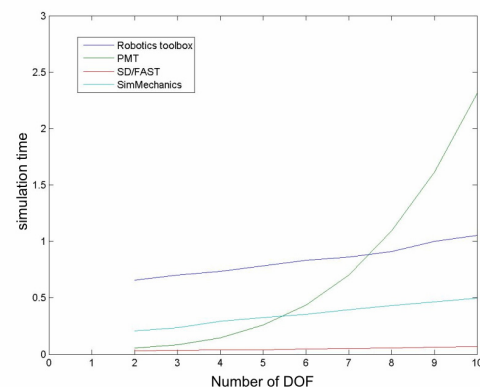


Fig. 3 Comparison of different modeling tools

### 2.4 Description of a model using SD/FAST

SD/FAST prescribes description of the system. For each rigid body in the system we have to define the following parameters (an example is shown in Fig. 4):

- name of the body
- definition of the body, to which the body is connected by a joint
- type and orientation of the interconnection joint
- distance from the COG of the body to the joint (bodytojoint)
- distance from the COG of the body to which the body is connected to the joint (inbtojoint)
- mass and inertia of the body

With definition of all bodies in the system, the structure of a whole system is defined.

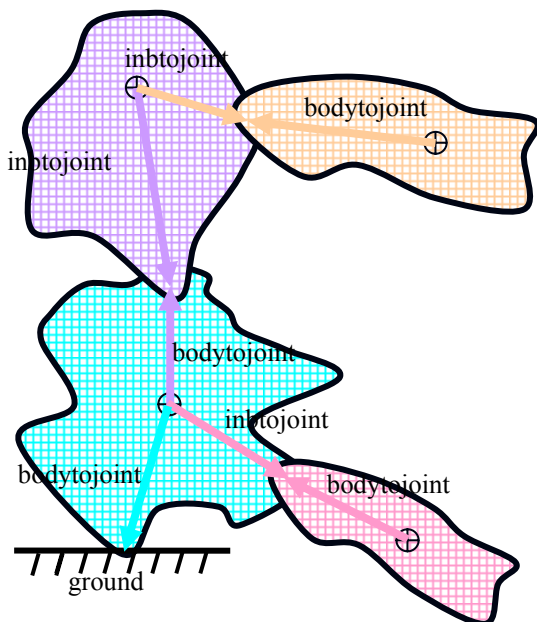


Fig. 4 Description of a system using SD/FAST

As described above we built two models of the jumping robot. Tab.2 shows the description of both models:

Tab.2 Description of the jumping robot

		Model on the ground	Model in the air
link	connected to	type of joint	type of joint
foot	ground	rot	2 trans + 1 rot
shank	foot	rot	rot
thigh	shank	rot	rot
trunk	thigh	rot	rot
Number of DOFs		4 ( $q_3-q_6$ )	6 ( $q_1-q_6$ )

When the system is completely described, SD/FAST generates the equations of motion for both models. These equations can be used in other programs to simulate or to control the robot system.

## 3 Simulation and control of the jumping robot

### 3.1 Dynamic model

The equations of motion which are generated by SD/FAST should be used in some other programming package, which provides integration of the equations. We have used Matlab/Simulink for the simulation of the jumping robot. The C-code of the equations of motion, which is the output of SD/FAST, is used in Simulink as an S-function. We have designed two Simulink blocks (two S-functions), one for the direct dynamics and one for the inverse dynamics.

#### 3.1.1 Inverse dynamics

Depending on the current state of the robot ( $q, \dot{q}$ ) the Simulink block for inverse dynamics defines the matrices  $H$ ,  $C$  and  $g$  from the dynamic model (1) (see Fig. 5). Additionally, the block defines also the position of the COG and the Jacobian matrix of the COG, which are used in control and simulation. The block considers also the model type (ground or air), which is the input in the block.

To get the requested outputs of the block, we have called the following functions, which are provided by SD/FAST: `sdpos` (position), `sdrel2cart` (Jacobian), `sdmassmat` ( $H$ ), `sdequivht` ( $g$ ), `sdfrcmat` ( $C$  and  $g$ ).

#### 3.1.2 Direct dynamics

On the other hand the direct dynamic model performs the integration of the equation of motion (Eq. (1)) and outputs the states of the robot ( $q, \dot{q}$ ) depending on the input torque  $\tau$ . In our case Matlab/Simulink performs the integration, therefore, we only have to define the derivatives of the states and to apply the input torque the following functions, which are provided by SD/FAST, are used: `sdstate` (get states), `sdderiv` (get derivative of states) and `sdhngt` (set torque).

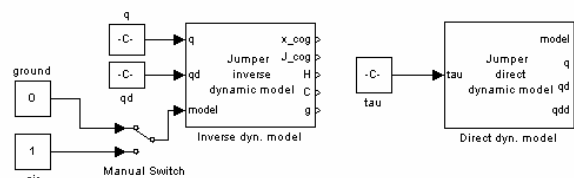


Fig. 5 Inverse and direct dynamic model blocks in Simulink

In the case of the direct dynamics special attention should be paid to switching between both dynamic models (ground and air). For example, system starts

from the ground. To switch from *the ground* to *the air model*, the ground reaction forces of the robot in the vertical direction have to be positive. In the moment of the switch all joint states maintains its value (are continuous).

On the contrary, the switch between *the air* to *the ground model* appears in the moment when the robot lands (the foot tip touches the ground). At this time an impact occurs and therefore, the joint velocities change its value momentarily (discontinuity appears). The joint velocities after the impact are calculated depending on the robot configuration and depending on the change of velocity in the point of the impact using the following equation [6]:

$$\Delta \dot{\mathbf{q}} = \mathbf{H}_{\text{air}}^{-1} \mathbf{S}^T (\mathbf{S} \mathbf{H}_{\text{air}}^{-1} \mathbf{S}^T)^{-1} \Delta \mathbf{v}, \quad (2)$$

where  $\mathbf{H}_{\text{air}}$  represents the  $\mathbf{H}$  matrix of the system in the air,  $\Delta \mathbf{v}$  is the change of velocity in the moment of impact in the point of the impact and  $\Delta \dot{\mathbf{q}}$  is the resulting change of the joint velocities before and after the impact. Matrix  $\mathbf{S}$  defines the constraints of the impact and is in our case equal to:

$$\mathbf{S} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (3)$$

To get the ground reaction force and the position of the foot tip the following functions are used: `screac` (reac. force), `sdpos` (position).

Special attention should be paid also on the numerical problems during the switching between models. However, this is out of scope of this paper.

### 3.2 Control of the system

To control the system two different controllers have to be used. In the same way, one controller has to be used for the ground phase and one for the air phase.

Furthermore, in the case of the jumping robot few DOFs are not controllable. The robot has only three controllable DOFs ( $q_4$ - $q_6$ ), while the others are uncontrollable (there is no actuator). In the ground case the rotation about the joint at the foot tip ( $q_3$ ) is not controllable, while the position of the foot tip ( $q_1$  and  $q_2$ ) is fixed. Similarly, in the air case the rotation about the foot tip joint as well as the vertical and the horizontal position of the joint are not controllable ( $q_1$ ,  $q_2$ ,  $q_3$ ). In the uncontrollable DOFs the control torque is zero.

Let us divide the dynamic model to the controllable and the uncontrollable part:

$$\begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_1 \\ \ddot{\mathbf{q}}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\tau}_1 \\ \boldsymbol{\tau}_2 \end{bmatrix}. \quad (4)$$

Let us presume that first few DOFs in kinematic chain are not controllable, i.e.  $\boldsymbol{\tau}_1 = 0$ . This condition leads to:

$$\begin{aligned} \mathbf{H}_{11} \ddot{\mathbf{q}}_1 + \mathbf{H}_{12} \ddot{\mathbf{q}}_2 + \mathbf{C}_1 + \mathbf{g}_1 &= 0, \\ \mathbf{H}_{21} \ddot{\mathbf{q}}_1 + \mathbf{H}_{22} \ddot{\mathbf{q}}_2 + \mathbf{C}_2 + \mathbf{g}_2 &= \boldsymbol{\tau}_2. \end{aligned} \quad (5)$$

Here, acceleration in uncontrollable joints  $\ddot{\mathbf{q}}_1$  is unknown and can be eliminated from the above equations:

$$\ddot{\mathbf{q}}_1 = -\mathbf{H}_{11}^{-1} (\mathbf{H}_{12} \ddot{\mathbf{q}}_2 + \mathbf{C}_1 + \mathbf{g}_1) \quad (6)$$

and

$$-\mathbf{H}_{21} \mathbf{H}_{11}^{-1} (\mathbf{H}_{12} \ddot{\mathbf{q}}_2 + \mathbf{C}_1 + \mathbf{g}_1) + \mathbf{H}_{22} \ddot{\mathbf{q}}_2 + \mathbf{C}_2 + \mathbf{g}_2 = \boldsymbol{\tau}_2 \quad (7)$$

and finally we get

$$\begin{aligned} &\underbrace{(-\mathbf{H}_{21} \mathbf{H}_{11}^{-1} \mathbf{H}_{12} + \mathbf{H}_{22})}_{\mathbf{H}_{\text{new}}} \ddot{\mathbf{q}}_2 + \\ &\underbrace{(-\mathbf{H}_{21} \mathbf{H}_{11}^{-1} \mathbf{C}_1 + \mathbf{C}_2)}_{\mathbf{C}_{\text{new}}} + \\ &\underbrace{(-\mathbf{H}_{21} \mathbf{H}_{11}^{-1} \mathbf{g}_1 + \mathbf{g}_2)}_{\mathbf{g}_{\text{new}}} = \boldsymbol{\tau}_2. \end{aligned} \quad (8)$$

We have acquired new model with three controllable DOFs. The other DOFs are not controllable directly, however, they can be controlled indirectly as can be seen from (6).

### 3.3 Example of a vertical jump

We have performed a simulation study of a vertical jump. To assure the verticality of the jump, the robot's COG has to move in the upward direction above the support polygon during the push-off phase of the jump. The second condition, which refers to the balance of the robot during the push-off phase, is the position of the zero moment point (ZMP). Both conditions are described in details in [7].

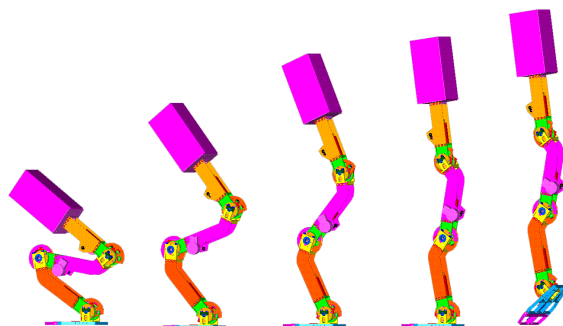


Fig. 6 Vertical jump of the robot

To control a model we have to switch the ground and the air model used in the controller. In the push-off phase the robot stands on the ground. Here, we use the ground model for the control. In the push-off phase we

control the position of the COG and ZMP in such a way that the robot performs a vertical jump. In the flight phase we use the air model for the control. Here, the robot should only stop the motion in all joints and prepare for the landing.

Since, we can only control the controllable DOFs, the following controller was used:

$$\mathbf{H}_{new}\ddot{\mathbf{q}}_{2c} + \mathbf{C}_{new} + \mathbf{g}_{new} = \boldsymbol{\tau}_{2c}, \quad (9)$$

where  $\boldsymbol{\tau}_{2c}$  is the control torque and  $\ddot{\mathbf{q}}_{2c}$  is the control acceleration of the controllable joints.  $\ddot{\mathbf{q}}_{2c}$  is defined as:

$$\ddot{\mathbf{q}}_{2c} = \ddot{\mathbf{q}}_{2d} + K_p \mathbf{e} + K_d \dot{\mathbf{e}}, \quad \mathbf{e} = \mathbf{q}_{2d} - \mathbf{q}_2. \quad (10)$$

Here,  $\mathbf{q}_{2d}$  and  $\mathbf{q}_2$  are the desired and the actual joint position of the controllable joints, respectively. The desired joint position is defined considering position of the COG and ZMP and is defined in [7].

Fig. 6 schematically shows the complete system, i.e. controller and direct dynamic model of the robot. Fig. 7 shows the controller, which merges two controllers one for the ground and one for the air phase.

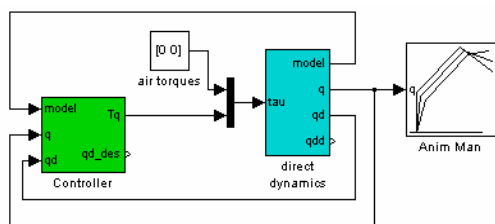


Fig. 7 Block diagram of the controlled jumping robot system

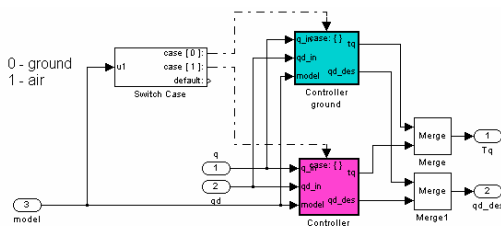


Fig. 8 Block diagram of the controller merged from ground and air controllers

The results of the experiment are shown in Fig. 8, where configurations during the jump are shown. Fig. 9 shows the position of the COG.

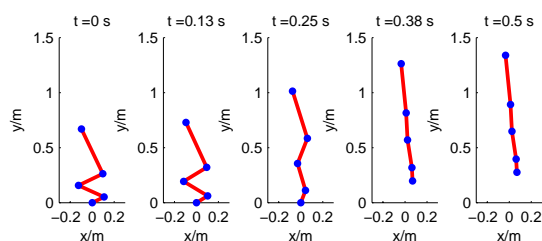


Fig. 9 Configurations of the robot during vertical jump

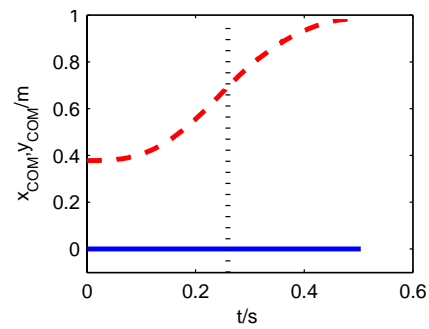


Fig. 10 Position of COG during vertical jump

## 4 Conclusions

We described the complete process of the modeling of the human-inspired jumping robot system. At the beginning we described the CAD model, then we acquired the dynamic model of the system using SD/FAST modeling tool, which has shown as a good and computationally very efficient tool, however, not very user friendly. We demonstrated the switching between two different models (ground and air), which can sometimes lead to numerical instability. Afterwards, we showed in details the development of the dynamic model of robotic systems with some uncontrollable joints. Simulink blocks of the inverse and direct dynamics were developed and used in experiment of a vertical jump.

## 5 References

- [1] M. Raibert, "Legged Robots That Balance" MIT Press, 1986.
- [2] H. De Man, D. Lefeber, F. Daerden, and E. Fagniet, "Simulation of a new control algorithm for a one-legged hopping robot (using the multibody code mechanics motion)", In Proc. International Workshop on Advanced Robotics and Intelligent Machines, pp.1-13, 1996.
- [3] L. Žlajpah, "Simulation in robotics", 5th Vienna Symposium on Mathematical Modeling, 2006
- [4] P.I. Corke, "A Robotics Toolbox for MATLAB", IEEE Robotics and Automation Magazine, vol. 3, pp. 24-32, 1996
- [5] L. Žlajpah, "Simulation of n-R Planar Manipulators", Simulation Practice and Theory, vol. 6, pp. 305 - 321, 1998
- [6] Y.F. Zheng and H. Hemami, "Mathematical modeling of a robot collision with its environment", Journal of Robotic Systems, vol.2, pp.289-307,1985.
- [7] J. Babič, D. Omrčen and J. Lenarčič, "Balance and control of human inspired jumping robot". Advances in Robot Kinematics.