Abstract—Robotic simulations can be carried out using several goal oriented programs still researchers prefer to use the limited SimMechanics environment because of its popularity in the scientific community. One big disadvantage of SimMechanics is that only fixed ground contacts can be realized with its toolboxes, while mobile robotics projects require a variable ground contact. The aim of this paper is to introduce a SimMechanics model of a 3D ground contact intended to be used with uneven ground.

I. INTRODUCTION

A number of robotic simulation environments are at the disposal of developers and researchers. Among them, the most appropriate and elaborate programs for embedded robotics research are perhaps the Gazebo, V-rep, Microsoft Robotics Developer Studio and Webots.

As an example the V-rep supports 3 physical engines, thanks to which it can simulate the real world physics and the interaction of objects. It contains all the necessary kinematic and inverse kinematic calculation algorithms. It is making it even easier to develop mobile robots since it’s capable of simulating real sensors and it disposes for example with a routing module as well. [9]

Despite the readiness of the mentioned tools, the researchers however prefer the more limited Matlab/SimMechanics environment. The reason for this is the overall acceptance and popularity of Matlab in the scientific community, and the possibility to have all the parameters of the model under control (all the computations/formulas can be checked or overwritten). [10]

SimMechanics primarily provides a “multibody” simulation environment for the mechanical systems. Robots fall into this category (as well as vehicle suspensions). The simulated system can be put together using solids, joints and elements representing constraints and forces. The software can generate and solve the equations describing the mechanical systems.

In the case of fixed (non-mobile) robots the SimMechanics environment provides a great foundation for different robot arm kinematics and dynamics testing, but for a basic simulation of mobile robots we can run into some difficulty. SimMechanics does not consider the interactions of objects, it can work only on bodies permanently bound together by joints.

Mobile robot simulations make it necessary to simulate the relationship between the robot and the ground (ground contact). Because SimMechanics lacks this feature, this has to be achieved by means of a separate Simulink tools.

II. DEFINITION OF THE TASK

The simulation of a 6 legged walking robot, which forms the core of our research [21], due to the earlier mentioned influences, was built in SimMechanics [16]. The model used a limited ground contact with limited functionality, which calculated the force between the ground and the legs which were perpendicular to the ground. However, due to a progressive expansion of the simulation conditions (uneven ground, not horizontal robot) it was necessary to achieve a more complex contact with the ground.

The new ground contact had to fulfill the following tasks:
- Definition of a strong interaction between any directions of the body (eg the robot feet) and a surface of any orientation (ground).
- The breakdown of the speed of the feet into components perpendicular and in the plane of the ground.
- Definition of perpendicular ground reaction force (F_n) from the velocity of the foot perpendicular to the ground and the distance of the two, using the damped mass-spring model.
- Determination of ground reaction forces acting in the plane of the ground from the velocity of the foot and the normal force using the constant friction model (Karnopp, Coulomb models).

III. CURRENT DEVELOPMENTS, LITERATURE

The implementation of the new model was preceded by studying of the works and literature of the others.

Many of the researches use kinematic models, where the complex interactions appearing at ground contact are simply ignored. An example is article [17], which deals with the energy consumption minimization of SILO-6 robot. Using the simpler kinematic model the researchers were able to determine the energy consumption with sufficient precision.

A detailed and well-documented kinematic model built in SimMechanics was used in [18]. The parallel robot in this article realizes the movements of an aircraft simulator. Because parallel robots are connected to the ground with one fixed point, the implementation of the complex ground was not needed.

A fine Hexapod SimMechanics simulation is shown in [1]. The work contains among others a functional ground contact, but even in this case only a horizontal ground contact has been worked out.
It is much easier to realize the ground contact with the previously mentioned target software. [19] deals with walking on uneven ground. In this article, the authors used Microsoft Robotics Developer Studio to create the robot model. Spring-damper model was used for surface normal reaction forces and Coulomb friction model was used for the ground friction forces.

A number of Simulink/SimMechanics models are available on the World Wide Web which include a ground contact implementation. Table I summarizes the achieved realizations found by us in the last couple of years.

Based on past research, three-dimensional, sloping ground developed ground contact was not found.

IV. IMPLEMENTATION

A. Description of the system

The model used in the ground contact test is shown in Fig. 1. The connection is implemented between a spherical body (Ball) and polygons (ground) loaded from an external file. If required, the body may be replaced by any other element, such as at any time it could be for example interchangeable with a robot leg.

SimMechanics is able to visualize simple shapes (spheres, cylinders, rectangular...) and it can compute their parameters (inertia, mass,...), but as the ground is a complex surface, it must be read in from an external source. In this case, it is a "stl" (STereoLithography) file, in which the uneven ground is made up from polygons generated by matlab functions. The external file reader "Solid" module is only used for displaying and is not able to return the position of the polygons. The latter must be addressed separately.

The ground and the body (hereinafter ball) are connected using a six degrees of freedom joint (6 DOF Joint) that has no constraints. As the position of the ground defined in the model is the world coordinate zero, therefore the connection between the two can be specified with respect to the world coordinate system.

The position and speed of the center of the ball can be obtained using a sensor element (Transform Sensor). For the computation of the vertical distance between the ball and the ground, and the orientation of the ground, the x, y and z coordinates of the ball have to be known. In order to compute the reaction forces acting on the ball, the vx, vy and vz velocities need to be known.

B. Ground orientation determination, transformations

As has been previously noted, the ball velocity vectors are defined in the world coordinate system. However, in order to compute the forces acting on the ball, these velocities need to be transformed in the plane of the touched polygon of the ground. To do this as a first step a rotation between the ground and the world coordinate system has to be determined.

<table>
<thead>
<tr>
<th>Model name</th>
<th>Description</th>
<th>Type of ground contact</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>bouncing_ball_heat</td>
<td>Vertically bouncing ball.</td>
<td>Spring damper, no friction.</td>
<td>[3]</td>
</tr>
<tr>
<td>HoppingRobot</td>
<td>One legged jumping robot in 3D.</td>
<td>Spring damper, no friction.</td>
<td>[5]</td>
</tr>
<tr>
<td>LandingGear</td>
<td>Highly complex flight simulator.</td>
<td></td>
<td>[6]</td>
</tr>
<tr>
<td>Mini_Golf</td>
<td>2D golf simulator.</td>
<td>2D collision, no friction.</td>
<td>[7]</td>
</tr>
<tr>
<td>MyRover</td>
<td>Wheeled robot simulator, prismatic damping.</td>
<td>Spring damper, viscous friction, without vertical force.</td>
<td>[8]</td>
</tr>
</tbody>
</table>
The height of the ground can be read-out from an external matrix on the basis of x and y coordinates. This matrix has to be generated together with the stl file. The columns and rows of the matrix correspond to the x and y coordinates of the ground, the stored value is the height. The matrix is read using a "2-D Lookup Table" which interpolates and returns the proper height result.

The rotation of the ground along the X axis can be determined by measuring the changes along the Y axis. The Z coordinate must be defined in two nearby points, in the (X, Y-YE) and (X, Y+YE) coordinates, where YE is an arbitrary unit of distance.

The rotation along the X-axis is equal to the Z height difference and the double distance (2xYE) ratio arc tangent (1).

$$\alpha = \arctan \frac{\Delta Z}{2Y_E}$$  \hspace{1cm} (1)

The rotation along the Y axis can be calculated similarly (2):

$$\beta = \arctan \frac{\Delta Z}{2X_E}$$  \hspace{1cm} (2)

The determination of the orientation angles is carried out by the "Get Orientation" Subsystem (Fig. 2).

Knowing the orientation of the ground, using rotational matrices (Equation (3) shows the rotation along the X-axis, and equation (4) shows the rotation along the Y-axis) it is possible to transform the ball velocity into ground plane coordinate system. The velocity has to be multiplied using matrices which rotate along the X and Y axis. \cite{11}

The determination of the orientation angles are done by the "Rotate Along" subsystems (Fig. 4).

The two matrices:

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix}$$  \hspace{1cm} (3)

$$R_y(\alpha) = \begin{bmatrix} \cos \alpha & 0 & \sin \alpha \\ 0 & 1 & 0 \\ -\sin \alpha & 0 & \cos \alpha \end{bmatrix}$$  \hspace{1cm} (4)

C. Normal forces, friction

The ground and the body in contact with it (regardless of whether it is a ball or for an example a robot leg) is considered to be a spring damper system. For this reason, the reaction force acted by the ground perpendicular on the ball can be determined by the physical model of the damped free vibration. Summarizing the forces acting on the weight of the ball the following equation can be described (5) \cite{12}, where m is the mass of the body, a is the acceleration, c is the damping coefficient, v is the speed of the body, k is the stiffness of the spring, and x represents the amount the spring is compressed:

$$ma + cv + kx = 0$$  \hspace{1cm} (5)

It can be expressed that the normal force acting on the ball is; the sum of the stiffness multiplied by the ball penetration, and damping coefficient multiplied by the speed of the ball (6). Of course, the direction of the forces must be taken into account.

$$F = ma - (cv + kx)$$  \hspace{1cm} (6)

The proportionality constants c and k have to be defined by us as a function of the knowledge of the body; the calculation of the velocity perpendicular to the body has been calculated previously; the penetration of the ball (which corresponds to the compression of a flexible body in a "rigid" environment) has been calculated using the world coordinates of the ball based on the vector containing the ground heights (lookup table).

The computation of the perpendicular forces acting on the ball is carried out by the "Spring Damper" subsystem (Fig. 3). The forces acting on the ball that are present in the ground plane can be obtained from the velocity of the ball (vx, vy), and from the impact strength of the ball perpendicular to the ground.
The reaction forces in the directions of X and Y can be calculated on the basis of different friction models [13]. The most accurate models often use simultaneously Stribeck (Fs), Coulomb (Fc), and viscose (Fv) components as well (Fig. 7) [14].

If the case of the ball it is sufficient to assume a non-slip connection, so in this case the friction can be neglected. In contrast, a real robot model could be much more detailed.

For the six-legged robot – to which this simulation was made for - the situation is more complex. The actual robot simulation for example uses primarily a Karnopp friction model. Fortunately, changing the friction models are not a problem because in the ground contact model, the module responsible for friction can be freely exchanged.

V. CONCLUSION

The operation of the system was also tested on two types of grounds (environment).

The first ground (Fig. 5) simulates a real, natural environment, while the second is an inverted bell (Fig. 6), which neatly tracks the bouncing of the ball.

The formula of the natural imitative ground (7):

\[
S=\text{abs(}\text{double(}\text{ifftn}}((\text{hypot(meshgrid(-50:49), meshgrid(-50:49)))+1e-1,}^{\text{-1.7}})\text{.* exp(rand(100)*1i*2*pi))))\]

(7)

The formula of the bell shaped environment (8):

\[
S=\text{exp}(-(X.^2+Y.^2)/2)
\]

(8)

In the simulation setup, the ball in both different environments realistically bounced (at the expected angles), before the movement gradually calmed down. Based on this the model is functional and it certainly copes with different circumstances.

In the next step, this will be built into the previously mentioned robot model to provide an opportunity to carry out more complex verification and testing.

REFERENCES


