Helicopter Lands on Uneven Terrain by means of Articulated Robotic Legs – Modelling, Simulation and Control Approach

Daniel Melia Boix[1], Keng Goh[2] and James McWhinnie[3]
School of Engineering and Built Environment, Edinburgh Napier University, Edinburgh, EH105DT, United Kingdom
[1]d.meliaboix@napier.ac.uk, [2]k.goh@napier.ac.uk, and [3]j.mcwhinnie@napier.ac.uk

Abstract - In this paper, a 3D mathematical model of a helicopter attached to a robotic landing gear is proposed to investigate into helicopter landing on uneven terrain. The landing gear consists of four articulated legs connected to the helicopter fuselage. Each leg has two revolute joints governed by PID joint controllers to generate motion with two degrees of freedom (*DoF*). A combination of Lagrange and Newton-Euler techniques is used to model the system dynamics, its motion and joint torques. A contact model is introduced to simulate ground interaction forces and a level controller that uses a PI algorithm to maintain the main body desired attitude when landing in two-axes slopes. Simulation results are compared to a second model designed using the SimMechanics toolbox for validation purpose.

Keywords—PID control, robotic landing gear, nonlinear dynamic model, contact model, level controller, simulations

I. INTRODUCTION

Helicopters are widely used in search and rescue missions, disaster areas or mountainous environments because of their ability to access remote or unprepared areas that are not accessible by any other means. This is due mainly to their ability to perform hovering and vertical landing/take-off manoeuvres. However, they usually incorporate fixed landing gear systems like skids or wheels, which limit their ability to land on irregular terrain.

A number of early publications suggested several solutions to cope with the problem of landing on sloped surfaces using hydraulic or mechanical systems to adapt the position of wheels/legs to the ground conditions [1], [2], [3], but with limited study available since then. A recent example of robotic landing gear was developed by the Georgia Institute of Technology under the DARPA's Mission Adaptive Rotor (MAR) program [4] consisting of four articulated robotic legs to ensure the helicopter stays level during landing. A practical demonstration was reported in [5].

Dynamics modelling techniques are generally divided into two main categories, the ones that are based in Newton-Euler equations, and the ones based on analytical mechanics approaches, like Lagrange method [6]. When modelling multibody systems, a common practice is to use an abstract, simplified model of the robot to reduce the model's complexity. A mixed approach is found in [7] and [8] where the authors use a modelling technique for quadruped robot locomotion based on decoupling the body and legs. For body position and attitude, a point-mass model of the *CoM* is analysed and the external forces are computed. Then the joint torques are calculated using a single leg dynamics model. The whole body model is constructed by coupling the equations of the floating-base body and each of the legs attached to it.

In this paper, a nonlinear dynamic model of a landing gear with four articulated legs and the control strategy to keep the helicopter's main body stable when landing on irregular terrain is presented. In Section II, the position and attitude of the main body are computed applying the Newton-Euler equations to obtain the *CoM* dynamics. In Section III, a kinematic and dynamic model of a single leg are derived using the Lagrange method. The entire model is built by coupling the *CoM* dynamics and the geometry of the main body with four single-leg models.

In Section IV, a level controller is designed to adapt the position of the legs to the ground conditions and maintain the desired attitude of the helicopter body. In Section V, a contact model to simulate a two-directional slope and the ground-leg reaction and friction forces is presented. Finally, in Sections VI and VII, a second model of the landing gear is built using the SimMechanics toolbox for multibody simulation and the results of both models are compared.

II. CENTER OF MASS DYNAMICS

The complete landing gear-helicopter model, is a multibody system formed by four robotic legs connected to the helicopter body. All four legs actuate parallel to the XY plane of the helicopter body-fixed frame and are situated two on each side of the landing gear as shown in Figure 1. The main body has a size and mass proportional to the dimensions and mass of the legs.

The position of the main body is defined by the distance of its *CoM* to the ground in the x, y and z directions and its attitude by the roll (θ_z), pitch (θ_x) and yaw (θ_y) angles. A point-mass model of the landing gear is used, to calculate the effect produced by the different forces in the motion of the system's centre of mass. All the different bodies that form the system are reduced to a single point that has a rotational inertia, *I*, and a total mass, *m*, which emulate those of the whole system and is located at the *CoM* of the whole system. For simplification purposes, the *CoM* is considered to be at the centre of the main body, as the mass of the legs is negligible compared to the mass of the body.

The forces considered for the calculations include the gravity acceleration, a thrust force model to control the descent rate and the ground reaction forces acting on the feet when they make contact with the ground. Newton-Euler equations are used to calculate the *CoM* both linear and angular motion.

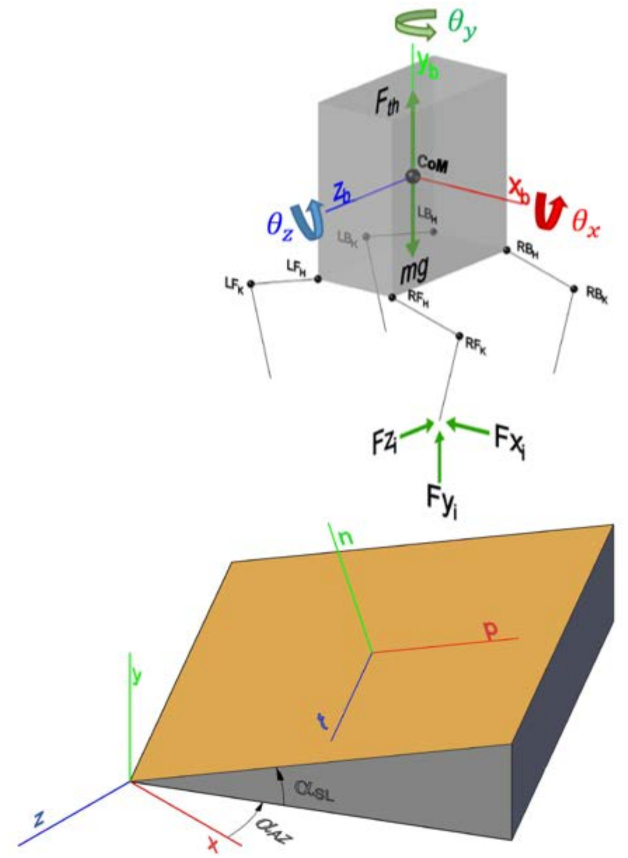

Figure 1. Sketch of the landing gear. The eight joints are Left Front Hip and Knee (LF$_H$, LF$_K$), Left Back Hip and Knee (LB$_H$, LB$_K$), Right Front Hip and Knee (RF$_H$, RF$_K$), and Right Back Hip and Knee (RB$_H$, RB$_K$). It shows the inertial frame (x,y,z), the body-fixed frame (x$_b$,y$_b$,x$_b$) and the ground slope frame (p,n,t)

The linear motion in each of the x, y and z directions of the inertial frame is governed by Newton's second law:

$$m\ddot{x}_{CoM} = \sum F_{x_i} \qquad (1)$$

$$m\ddot{y}_{CoM} = \sum F_{y_i} + F_{th} - mg \qquad (2)$$

$$m\ddot{z}_{CoM} = \sum F_{z_i} \qquad (3)$$

where m is the system's total mass, g is the gravity acceleration, F_{th} is the thrust controller force, and F_{x_i}, F_{y_i} and F_{z_i} are the resultant ground reaction forces in each foot in the respective directions.

The angular motion on the respective axes is computed using Euler's equations of motion

$$\sum M_x = I_{xx}\dot{\omega}_x - (I_{yy} - I_{zz})\omega_y\omega_z \qquad (4)$$

$$\sum M_y = I_{yy}\dot{\omega}_y - (I_{zz} - I_{xx})\omega_x\omega_z \qquad (5)$$

$$\sum M_z = I_{zz}\dot{\omega}_z - (I_{xx} - I_{yy})\omega_x\omega_y \qquad (6)$$

where I_{xx}, I_{yy}, and I_{zz}, are the system's moments of inertia on each axis, and ω_x, ω_y, ω_z, are angular velocity terms of the body-fixed frame.

The sum of moments on each axis is calculated as follows

$$\sum M_x = \sum (F_{y_i} \cdot z_i) + \sum (F_{z_i} \cdot y_i) \qquad (7)$$

$$\sum M_y = \sum (F_{x_i} \cdot z_i) + \sum (F_{z_i} \cdot x_i) \qquad (8)$$

$$\sum M_z = \sum (F_{x_i} \cdot y_i) + \sum (F_{y_i} \cdot x_i) \qquad (9)$$

x_i, y_i and z_i are the distances from the *CoM* to each foot in the x, y and z directions of the body-fixed frame.

The thrust controller is a simplified version of the one used in [5] where the only aim is to control the descent rate

$$F_{th} = C \cdot (\dot{y}_{ref} - \dot{y}) + mg \qquad (10)$$

where $\dot{y}_{ref}$ is the desired descent rate and C is a constant.

III. Single-Leg Model

This section presents the mathematical model of a single robot leg. Every leg used in the landing gear has the same structure. It has 2 links and two revolute joints at the hip and knee with its axes of rotation perpendicular to the XY body-fixed plane, therefore the leg-system has 2-*DoF*. The hip and knee angles of the leg *i* are represented by θ_{h_i} and θ_{k_i} respectively with its origin as shown in Figure 2 and positive in the anti-clockwise direction. The length of the upper and lower links is represented by l_{U_i} and l_{L_i} respectively and their masses are m_{U_i} and m_{L_i}. For simplification, the masses are considered to be concentrated at the end of each link.

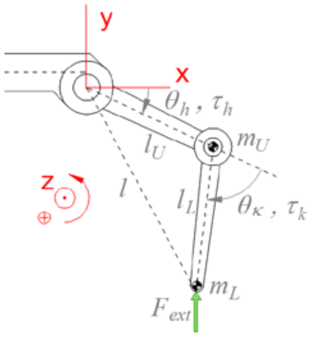

Figure 2. Sketch of a robotic leg with relevant parameters. It shows dimensions and masses of links, joint angles and torques and external forces.

The equations of motion of the single leg are derived using the Lagrange-Euler formulation as it presents a more systematic derivation and provides a closed-form expression [9].

The equations of motion are obtained by solving the Euler-Lagrange equation:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}_n}\right) - \frac{\partial \mathcal{L}}{\partial \theta_n} = \tau_i \qquad (11)$$

where θ_n is the joint angle of the link n, and τ_i represents the torque applied to the joint i.

The net torque acting on the hip and knee joints is given by the expressions [10]:

$$\begin{aligned}\tau_{T_h} = \big((m_U + m_L)l_U^2 + m_L l_L^2 \\ + 2m_L l_U l_L cos\theta_k\big)\ddot{\theta}_h \\ + (m_L l_L^2 + m_L l_U l_L cos\theta_k)\ddot{\theta}_k \\ - m_L l_U l_L \big(2\dot{\theta}_h\dot{\theta}_k + \dot{\theta}_k^2\big)sin\theta_k \\ + (m_U - m_L)g l_U cos\theta_h \\ + m_L g l_L cos(\theta_h + \theta_k)\end{aligned} \tag{12}$$

$$\begin{aligned}\tau_{T_k} = m_L l_L^2\ddot{\theta}_k + (m_L l_L^2 + m_L l_U l_L cos\theta_k)\ddot{\theta}_h \\ + m_L l_U l_L \dot{\theta}_h^2 sin\theta_k \\ + m_L g l_L cos(\theta_h + \theta_k)\end{aligned} \tag{13}$$

The total joint torque, τ_{Tn}, is the sum of the torque induced by the external forces acting on the leg, F_{ext}, and the joint actuator torque, τ_n, produced by a PID joint controller. The external forces that are considered here are the ground reaction and friction forces. The torque induced in the joints by the external forces is computed using the transpose of the Jacobian matrix, J^T.

$$\tau_{T_n} = \tau_n + J^T \cdot F_{ext} \tag{14}$$

The inverse kinematics equations provide the reference joint angles for the joint controllers

$$\theta_{h_r} = tan^{-1}\left(\frac{x}{y}\right) + cos^{-1}\left(\frac{l_U^2 + l^2 - l_L^2}{2l_U l}\right) - 90° \tag{15}$$

$$\theta_{k_r} = cos^{-1}\left(\frac{l_U^2 + l_L^2 - l^2}{2l_U l_L}\right) - 180° \tag{16}$$

$$l = \frac{y}{\cos(tan^{-1}\left(\frac{x}{y}\right))} \tag{17}$$

where the value of x is fixed, and the value of y has a fixed part and a variable part depending on the level controller output. This way, the terrain slope is overcome by adjusting the vertical position of the foot while maintaining the horizontal distance.

The hip-foot distance is calculated using forward kinematics, based on the measured joint angles

$$x_{h-f} = l_U cos\theta_h + l_L cos(\theta_h + \theta_k) \tag{18}$$

$$y_{h-f} = l_U sin\theta_h + l_L sin(\theta_h + \theta_k) \tag{19}$$

The foot position with respect to the ground is calculated using the *CoM* position and the hip-foot distance.

IV. Level Controller

The level controller (*LC*) adjusts the position of the legs to maintain the main body in a stable position even when landing on a sloped terrain. It takes the roll and pitch angles as inputs and generates a signal to increase/decrease the height (y coordinate) of all legs in order to level the system ($\theta_x = \theta_z = 0$).

A proportional-integral (PI) controller is chosen to control the height of the legs.

$$\begin{bmatrix} y_{inc\ roll} \\ y_{inc\ pitch}\end{bmatrix} = \begin{bmatrix} K_P e_z + K_I \int_0^t e_z dt \\ K_P e_x + K_I \int_0^t e_x dt\end{bmatrix} \tag{20}$$

where y_{inc} is the variation of the leg height (during roll and pitch motion), e_i is the error in the respective angle between the desired value and the measured one ($e_i = \theta_d - \theta$) and K_P, and K_I are proportional and integral gains respectively.

The height of each leg is adjusted by adding/subtracting the *LC* outputs to the initial height of each leg:

$$y_{LF} = y_0 + y_{inc\ roll} + y_{inc\ pitch} \tag{21}$$

$$y_{LB} = y_0 + y_{inc\ roll} - y_{inc\ pitch} \tag{22}$$

$$y_{RF} = y_0 - y_{inc\ roll} - y_{inc\ pitch} \tag{23}$$

$$y_{RB} = y_0 - y_{inc\ roll} + y_{inc\ pitch} \tag{24}$$

Overall, the simulation starts with the legs in initial position and the *CoM* at the initial height (Figure 3). The aircraft descends at a controlled speed and the position and attitude of the body are computed. Using kinematics, the feet position are determined and when ground contact is made, the ground contact forces are simulated. As the landing gear tilts in the direction of the slope, the level controller measures the pitch and roll angles and corrects the desired position of each leg. With the inverse kinematics the reference joint angles are calculated and the PID joint controllers move the legs to the desired position.

V. Ground Contact Model

A. Terrain definition

The ground modelled in this simulation consists on a slope defined by two parameters: α_{AZ}, which is the azimuth angle or rotation about y-axis, and α_{SL}, which is the inclination of the slope (Figure 1). The azimuth parameter is introduced in order to create a 2-axes slope which direction is not aligned with the axes of the body-fixed frame.

The ground slope coordinate system is defined by the Cartesian system formed by the normal direction to the ground, n, the parallel direction to the slope, p, and the transversal direction to the slope, t. To calculate ground reaction forces, the foot coordinates need to be referred in the ground slope coordinate system. The conversion from *XYZ* coordinates to the Parallel-Normal-Transversal (*PNT*) system is given by the expression:

$$(p, n, t)^T = R(z, \alpha_{SL}) \cdot R(y, \alpha_{AZ}) \cdot (x, y, z)^T \tag{25}$$

where R(u,α) is a 3x3 rotation matrix of an angle α, around the axis u.

B. Normal Force

To simulate the ground contact forces during landing, a compliant contact model composed by a spring-damper system has been used. The main advantages of this approach are its good performance and straightforward implementation. Numerous implementations of this technique have been reported [11], [12], [13].

Equation (26) represents the normal force to the ground when a foot lands. The ground reaction force resists the leg tip penetrating the ground surface and it is proportional to the amount of ground penetration (spring

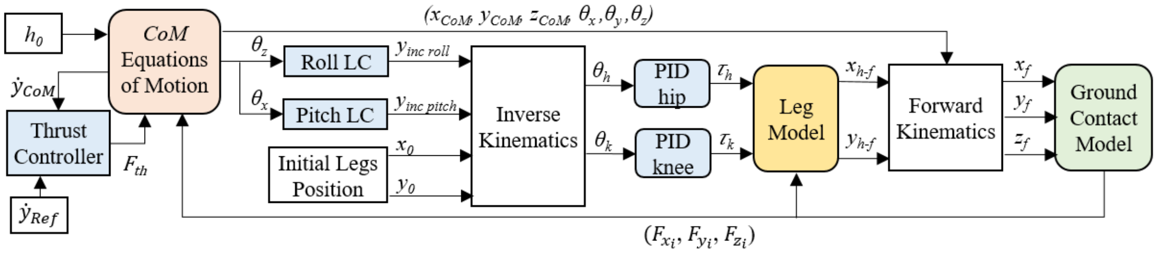

Figure 3. System Block Diagram of a single leg. Initially, the rotorcraft starts at height, h₀, and the legs in initial position (x₀, y₀). The thrust controller regulates the rate of descent and the roll and pitch level controllers adjust the position of the legs. The joint angles are calculated using inverse kinematics and PID joint controllers control the position of the legs. The ground reaction forces on each leg are calculated and fed back into the CoM equations of motion and the single-leg models.

component) and the rate of penetration (damper component).

$$F_n\begin{cases} -n_i\cdot k - \dot{n}_i\cdot b & if\ n_i\le 0\\ 0 & if\ n_i>0\end{cases} \tag{26}$$

Where n_i is the coordinate on the i^{th} foot in the n direction), k is the spring elastic constant, and b is the damping ratio. The normal force is zero before the impact and starts to make effect at the moment of touchdown, when $n_i \le 0$. To avoid 'sticking forces' only positive values of F_n are allowed.

C. Friction force

For the model, friction force exists between the foot and the slope surface (p and t axis) after touchdown. A number of frictional force models have been reviewed and considered [11], [13]. The frictional model used in this paper is a variation of the Coulomb friction model for its simplicity.

Friction force has opposite direction to the sliding velocity, and is a non-linear function of the relative velocity and position of the two contacting surfaces. At the moment of touchdown, the friction force is in the "sticking" region, and it is modelled as a spring-damper system. It is proportional to the velocity and displacement of the foot in each direction. If the velocity and displacement keep growing until the friction force exceeds the maximum static friction force, F_{f_S}, then the contact model switches to "slip" mode and the friction force is equal to the dynamic friction force, F_{f_D}.

$$F_p\begin{cases} -(p_i - p_{0_i})\cdot k - \dot{p}_i\cdot b & if\ F_p < F_{f_S}\\ F_{f_D} & if\ F_p \ge F_{f_S}\end{cases} \tag{27}$$

$$F_t\begin{cases} -(t_i - t_{0_i})\cdot k - \dot{t}_i\cdot b & if\ F_t < F_{f_S}\\ F_{f_D} & if\ F_t \ge F_{f_S}\end{cases} \tag{28}$$

$$F_{f_S} = \mu_s F_n \tag{29}$$

$$F_{f_D} = \mu_D F_n \tag{30}$$

where μ_S and μ_D are the static and dynamic friction coefficients, and p_{0_i} and t_{0_i} are the p and t coordinates of the i^{th} foot at the moment of ground contact. The distances $(p_i - p_{0_i})$ and $(t_i - t_{0_i})$ are the displacement of the foot on each direction after touchdown.

D. Resultant Ground Reaction Forces

The normal and friction forces have been calculated with respect to the terrain slope coordinate system (*PNT*). They need to be converted to the world coordinate system (*XYZ*) before they can be introduced in the *CoM* equation of motion and the single-leg model

$$\left(F_{x_i}, F_{y_i}, F_{z_i}\right)^T = R(n, -\alpha_{AZ})\cdot R(t, -\alpha_{SL}) \cdot \left(F_{p_i}, F_{n_i}, F_{t_i}\right)^T \tag{31}$$

VI. SimMechanics Model

Due to the increasing level of complexity of the final model derived, a second model has been constructed using a multibody system modelling tool to validate the first model.

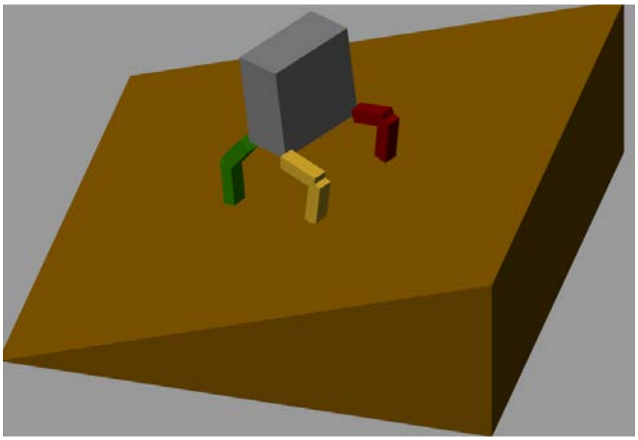

Figure 4. SimMechanics model. Mechanics Explorer view of the system landing on a slopped terrain

The second model of the landing gear is modelled and simulated using the SimMechanics toolbox from the MATLAB/Simulink package. Using simulation blocks representing bodies, joints, constraints, sensors and actuators, and forces, it formulates and solves the equations of motion for the intended system. The model can also interact with control blocks developed in MATLAB/Simulink [14]. Figure 4 shows a view of the Mechanics Explorer, a part of the SimMechanics toolbox that allows the visualisation of the physical system that is being modelled so the dynamics and behaviour of the system can be easily observed.

VII. Simulation Results And Comparison

A. Simulation parameters

The mathematical model is implemented in MATLAB/Simulink and the results are compared to a second model built in SimMechanics. The weights, dimensions and motors' maximum torque used in the simulations have been chosen to match with those of a laboratory-built prototype that will be developed in future work. The parameters used in the simulations are summarised in Table 1.

Table 1. Prototype physical parameters

Parameter	Symbol	Value
Upper/Lower leg mass	m_U/m_L	0.1/0.15 kg
Total system mass	m	3.5 kg
Upper/Lower leg length	l_U/l_L	0.0935/0.1045 m
Main body dimensions	$D_x/D_y/D_z$	0.1/0.2/0.2 m
Motor max torque	T_{max}	18kfg·cm / 1.76Nm
Spring coefficient	k	1500 kg/s^2
Damper coefficient	b	40 kg/s
Friction coefficients	μ_S/μ_D	0.8/0.5
Descent rate	$\dot{y}_{Ref}$	-0.25 m/s
Terrain slope	α_{SL}/α_{AZ}	20°/30°

The PID controllers are tuned manually by trial and error to obtain the desired performance. The controller parameters are summarised in Table 2.

Table 2. Controllers' parameters

Controller Gain	Joint controllers	Roll/Pitch LC
K_P	50	0.14
K_I	30	1.4
K_D	4.5	--

B. Simulation results

The results of the simulations are shown in Figures 5-9 where both models are tested and compared when landing on a slope at 20° of inclination and 30° of azimuth, at 0.25 m/s.

Figure 5 shows the roll and pitch angles of the main body. It shows how, during touchdown the main body starts to tilt in the roll and pitch direction. Without the level controller the system would settle down at an inclination of about 19° and 11° respectively due to the slopped terrain. But with the level controller the maximum body inclination is reduced considerably and quickly settles to 0°. The variation of the yaw angle due to the action of the level controller is negligible.

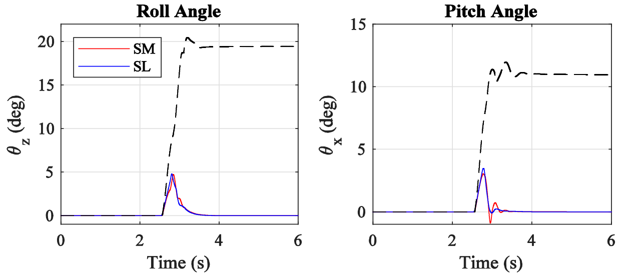

Figure 5. Roll and pitch angles of the main body. Blue plots represent the Simulink model and red ones represent the SimMechanics one. Dotted line without the level controller

Figure 6 shows the translational movement of the *CoM*. As it can be seen, as the level controller corrects the attitude of the body, it also reduces the displacement of the *CoM*, giving more stability to the system. Both models evolve at different paces but settle down at similar distances. The remaining displacement is mainly due to the displacement of the whole system along the terrain slope.

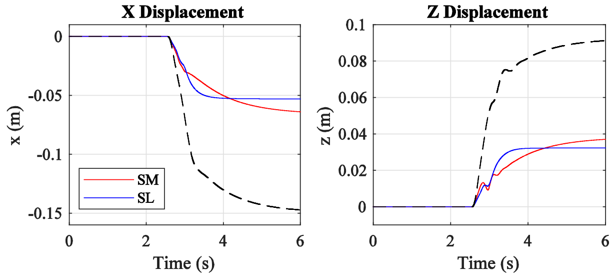

Figure 6. Centre of Mass displacement in the XZ plane. Blue plots represent the Simulink model and red ones represent the SimMechanics one. Dotted line without the level controller.

To correct the inclination of the helicopter body the level controller adapts the height of the robotic legs by adjusting the joint angles. Figure 7 shows the joint angles in all 4 legs where it can be seen how all 4 legs start in the same position and how, after touchdown, the right legs retract by increasing the hip angle and reducing the knee angle and opposite for the left legs.

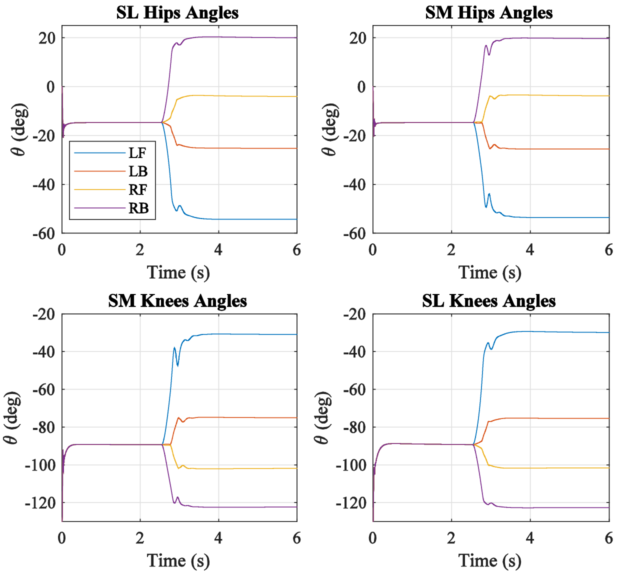

Figure 7. Joint angles is the Simulink and SimMechanics models

Figure 8 shows the ground reaction forces. It can be seen how the forces at all legs converge to the same value as the roll and pitch angles settle back to 0°.

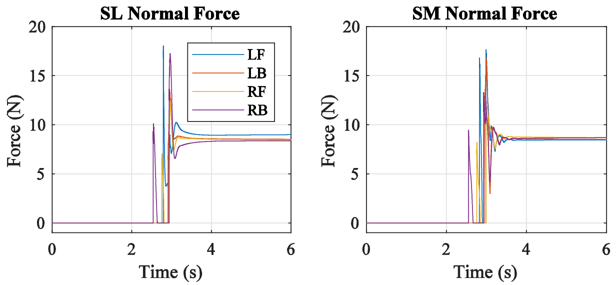

Figure 8. Normal ground reaction force

Figure 9 shows the joint torques in both models where it can be seen how the weight of the system is distributed between all four legs after landing. The values obtained in both models are very similar.

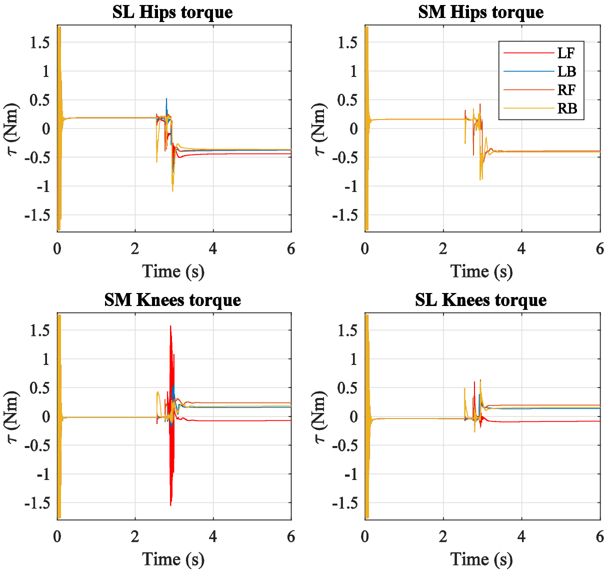

Figure 9. Joint torques in the SimMechanics (SM) and Simulink (SL) models

C. Results Analisis

The comparison between the mathematical and SimMechanics models shows that both models behave in a very similar manner. However there are several factors that can affect system performances. For example, when deriving the dynamic model of the single leg, it is assumed that the mass of each link is situated at the end of the link, and for the system's *CoM*, it is assumed that is located at the centre of the main body. By contrast, SimMechanics computes the centre of mass of each individual body. Another source of error can come from the system's total inertia, which is not calculated for every link. Despite the differences produced in the result comparison in the preceding section, both models demonstrate close correlation in their performances.

The level controller can be seen to reduce the peak inclination angle of the main body and ensure that the system is levelled. In this paper, the system has been tested on a slope of 20° of inclination which is beyond what a standard helicopter can handle. The maximum inclination that the system can handle will depend on the chosen parameters like the geometry of the legs and the terrain properties. In terms of leg geometry, the length of each leg segment and the separation between the feet will determine the maximum distance that the legs can extend/retract and therefore the maximum terrain slope that it can be overcome. Regarding the type of terrain, the friction coefficients will determine the maximum inclination before the system starts to slide down the slope. By changing these coefficients, different terrain types can be modelled for further analysis.

VIII. Conclusions

This paper has presented a methodology to obtain the dynamic model of multibody systems, in this case, a landing gear for helicopters consisting of four robotic legs and a main body. Apart from the nonlinear dynamic equations, the model also includes the kinematic equations, a landing scenario, a controller to regulate the rate of descent, a ground contact model to simulate the ground-leg interaction, and a level controller to maintain the stability of the main body when landing on uneven terrains. The model has been designed in MATLAB/Simulink environment and its performance has been compared with a SimMechanics built model. The results have shown a very similar behaviour from the two models. The performance of the level controller in both cases provides good correlation and maintains the stability of the helicopter while landing on slopped terrain.

The dynamic model presented in this paper constitutes a flexible tool that can be used to test different landing conditions.

IX. References

[1] S. H. Mason, "Helicopter self-leveling landing gear". United States Patent US 3857533 A, 1974.

[2] D. W. Felder, "Slope Landing Compensator System". Patent US4062507 A, 1977.

[3] S. M. Calvert, "Telescoping Landing Leg System". US Patent US9033276 B1, 19 May 2015.

[4] DARPA, "Robotic Landing Gear could enable future helicopters to take off and land almost everywhere," October 2015. [Online]. Available: http://www.darpa.mil/news-events/2015-09-10.

[5] V. Manivannan, J. P. Langley, M. F. Costello and M. Ruzzene, "Rotorcraft Slope Landings with Articulated Landing Gear," in *AIAA Atmospheric Flight Mechanics (AFM) Conference*, 2013.

[6] E. Leylek, M. Ward and M. Costello, "Flight Dynamic Simulation for Multibody Aircraft Configurations," *Journal of Guidance, Control and Dynamics*, vol. 35, pp. 1828-1842, 2012.

[7] X. Zhang, L. Lang, J. Wang and H. Ma, "The Quadruped Robot Locomotion Based on Force Control," in *27th Chinese Control and Decision Conference (CCDC)*, 2015.

[8] B. U. Rehamn, M. Focchi, J. Lee, H. Dallali, D. G. Caldwell and C. Semini, "Towards a Multi-legged Mobile Manipulator," in *IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden, 2016.

[9] M. W. Spong, S. Hutchinson and M. Vidyasagar, Robot Modelling and Control, John Wiley & Sons, Inc, 2006.

[10] K. Goh, D. Melia, J. McWhinnie and G. Smith, "Control of Rotorcraft Landing Gear on Different Ground Conditions," in *IEEE International Conference on Mechatronics and Automation*, Harbin, China, 2016.

[11] A. Ehsaniseresht and M. Mohammadi Moghaddam, "A new ground contact model for the simulation of bipeds' salking running and jumping," in *RSI International Conference on Robotics and Mechatronics*, Tehran, Iran, 2105.

[12] D. Naef, "Quadruped walking/running simulation," Swish Federal Institute of Technology, Zurich, 2011.

[13] B. Raiszadeh and D. Way, "Ground Contact Model for Mars Science Laboratory Mission Simulations," in *AIAA Atmospheric Flight Mechanics Conference*, Minneapolis, Minnesota, 2012.

[14] Y. Wang, M. Soltani and A. H. Dil Muhammad, "Dynamic Modeling and Simulation of Marine Satellite Tracking Antenna Using Lagrange Method," in *18th International Conference on Computer Modelling and Simulation*, Cambridge, UK, 2016.