RockBarrierAnalytica: First Draft

Ivo Gasparini*
Supervisor: Dr. Katharina Schwarz-Platzer*

Technical Report
Burgdorf, 25.08.2023

ivo.gasparini@bfh.ch

*Bern University of Applied Sciences BFH, School of Architecture, Wood and Civil Engineering AHB

Abstract

Rockfall barriers play a pivotal role in ensuring safety in areas susceptible to rockslides. Despite the numerous existing models, certain complexities in rockfall barrier mechanics remain unclear. This report introduces RockBarrierAnalytica (RBA), an open-source explicit Discrete Element Method (DEM) simulator built in Taichi Lang. RBA blends the Euler method with Taichi's parallelization capabilities to offer a unique, adaptable, and intuitive simulation framework. This tool discretizes barrier components, integrates comprehensive interaction logic, and provides an interactive visualization platform. RBA is still under development and requires further refinement. It invites community involvement and, with continued enhancements, could emerge as a new useful simulation tool.

1 Introduction

Flexible rockfall barriers are essential in ensuring safety in regions prone to rockslides and geological disturbances [1]. Over the past two decades, various numerical models have been developed to understand and optimize these structures [2], employing methods like the Finite Element Method (FEM) [3, 4, 5, 6, 7], the Discrete Element Method (DEM) [8, 9, 10] or a hybrid of both [11].

However, capturing the dynamic impact response of these barriers remains challenging. The diverse geometrical configurations and potential impact scenarios introduce complexities that can make many models fall short. As a result, significant gaps persist in our understanding and predictive capabilities of rockfall barriers.

Historically, most solutions in this domain have been closed-source, built on commercial software platforms. While these models, rooted in general-purpose physics engines, are effective, they might not be optimized for unique rockfall scenarios or innovative barrier designs. Many of these models prioritize simulations of test conditions based on the guidelines set by the European Organisation for Technical Approval [12], which may not always reflect real-world conditions. This underscores the potential advantage of a more adaptable, open-source framework to spur innovation and deepen understanding.

This report introduces RockBarrierAnalytica (RBA): a novel, open-source simulator built in Taichi Lang, a high-performance parallel programming language embedded in Python. Developed as a student project, the model is characterized by its simplicity, adaptability, and the seamless integration of intuitive algorithms with the Euler method. The synergy between Python and Taichi ensures enhanced code readability. While the model's streamlined design affords it flexibility, it's not without its limitations. As it stands, the simulator lays a foundation for future, more intricate and comprehensive simulations of rockfall barriers.

2 Materials and methods

2.1 Time Integration Using Euler's Method

Euler's method is a simple explicit first-order numerical procedure employed to solve ordinary differential equations. Given an initial value problem:

$$\frac{du}{dt} = f(u, t) \tag{1}$$

with an initial condition $u(t_0) = u_0$, the Euler method approximates the solution at time t_{n+1} as:

$$u_{n+1} = u_n + \Delta t f(u_n, t_n) \tag{2}$$

where Δt is the time step.

For dynamic problems, the acceleration $\mathbf{a}$ at each node of the simulation is related to the force $\mathbf{F}$ and mass $\mathbf{M}$ as:

$$\mathbf{F} = \mathbf{M}\mathbf{a} \tag{3}$$

Using Euler's method, the velocity $\mathbf{v}$ and displacement $\mathbf{u}$ at the next time step are updated as:

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \Delta t \mathbf{a}_n \tag{4}$$

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \Delta t \mathbf{v}_n \tag{5}$$

The Euler method is straightforward but has limitations, especially for dynamic systems with rapid changes. Its linear approach can miss quick shifts within the chosen time step, Δt, leading to inaccuracies. The solution's stability can also be compromised if Δt is too large relative to the system's smallest element size and wave speed. For RBA, which simulates impacts occurring in less than a second, using a small Δt is both practical and necessary. By choosing a smaller time step, we capture the rapid events accurately, mitigating the Euler method's shortcomings for such brief scenarios.

2.2 Taichi Lang

Taichi is a recent high-performance programming language that provides a flexible platform for creating, among other things, physically based simulations using a blend of imperative and data-oriented styles. It has many relevant advantages:

- Embeddability: Taichi is embedded in Python, making it easy to leverage the Python ecosystem for data processing, visualization, and more.
- Performance: Taichi can handle large-scale simulations efficiently. Its backend supports both CPUs and GPUs, allowing for parallel execution and optimized performance.
- Flexibility: The language supports a range of parallel primitives, such as parallel for-loops and atomic operations. This makes it ideal for simulating complex physical systems that involve interactions between many elements.
- Ease of Use: Taichi prioritizes user-friendliness. Its Pythonic syntax ensures that the code remains readable and maintainable, and the switch between CPUs and GPUs is easy and seamless.

In the context of RBA, Taichi provides the computational backbone, allowing for the efficient simulation of rockfall barriers. Its synergy with Python ensures that the simulator remains accessible and easy to extend.

3 RockBarrierAnalytica: structure and logic

3.1 Initialization of the Rockfall Barrier Components

To effectively simulate the dynamics of the rockfall barrier, its various components are discretized within the computational framework. This includes the rock, net, shackles (both horizontal and vertical, which interconnect the nets and fix them to the ropes), ropes, and posts. Each of these components is represented by a set of discrete points, classifying RBA as a DEM simulator. These discrete points, or nodes, form the foundation for computations using the Euler method. The resolution of ropes and nets, determined by the number of points, is adjustable. However, it's important to note that increasing the resolution directly impacts the computational demands of the simulation.

- **Position Matrices:** For each component, a matrix x is defined to store the positions, with 3D vectors, of these discrete points. For instance, x_ball represents the position of the ball, x_net denotes the positions of the nodes in the net, and so on. These matrices are initialized based on the specific geometrical configurations of the barrier components.
- **Velocity Matrices:** Corresponding to each position matrix, a velocity matrix v is defined. This matrix captures, with 3D vectors, the velocity of each discrete point in the system. For example, v_ball stores the velocity of the ball, v_net contains the velocities of the net nodes, and so forth. Initially, these matrices might be set to zero or any predefined values based on the initial conditions of the simulation.

Barrier geometries are freely adjustable and follow the designations given in [13], according to the schematic in fig. 1:

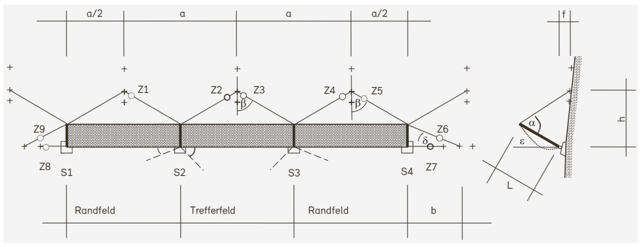

Figure 1. Barrier geometries according to the swiss quality assessment standards for rockfall barriers [13].

Because the geometries (fig. 1) are variable, the contact points between some barrier elements (e.g., between shackles and ropes) are found iteratively before the simulation begins and saved as indices in additional matrices to speed up and reduce the computational overhead of the contact search algorithms during the simulation.

A brief overview of the discretization of each barrier component as points in the 3D space is given below:

Component	Description
Ropes	Represented as series of points. Categorized into lower bearing ropes, upper bearing ropes, upslope ropes, and lateral support ropes. Each type is distinctly initialized based on its geometry and position.
Nets	Discretized into a grid of nodes. The net is further segmented into quads, determined by the total width and height of the net divided by the number of nodes in each direction.
Shackles	Represented in two distinct types: horizontal and vertical. Function as connectors, bridging the nets to the ropes and amongst themselves.
Posts	Initialized as 2 points: one for the post base, and one for the post top, based on the net's width and its height.
Rock	Represented as a singular point with a radius (so a sphere).

Table 1. Discretization of rockfall barrier components.

In fig. 2, a rendered scene is depicted. From the viewer's perspective, the orientation of the 3D axes is as follows:

- The x axis is horizontal, extending positively from left to right.
- The y axis is vertical, increasing in value as it moves upwards.
- The z axis extends positively towards the viewer.

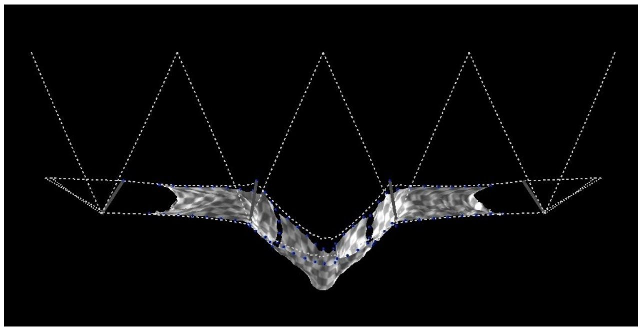

Figure 2. A snapshot from a trial simulation using RBA.

3.2 Node Interactions

After initialization, a Taichi Kernel (ensuring parallelization) is invoked at each timestep Δt, encapsulating the physical logic governing the simulation components and their interactions. For instance, gravitational acceleration (scaled by the timestep Δt) is imparted to the velocity vectors of each node of the simulation.

3.2.1 Spring/Damper Logic

The primary mechanism that dictates the interactions among the barrier components, as well as between individual nodes within those components, is a spring/damper model. This model calculates the forces by considering the relative movements and speeds of the nodes. The general formula for a spring/damper system is given by:

$$\mathbf{F} = -k(\Delta x - l_0) - c\Delta v$$

where $\mathbf{F}$ is the force, k is the spring constant, Δx is the displacement, l_0 is the equilibrium length of the spring, c is the damping constant, and Δv is the relative velocity.

Various spring/damper configurations, tailored for specific barrier elements or unique needs, have been integrated, providing adaptability in modeling a range of interactions in the system. Additionally, these configurations can be swapped out as needed.

1. **Standard Spring/Damper**: This is the basic model that computes the force based on the relative displacement and velocity of two points. Damping is applied in the direction of displacement, following the dashpot principle.
2. **1D Spring/Damper**: Similar to the standard model but the damping is applied regardless of direction, making it effectively one-dimensional.
3. **Yielding Spring/Damper**: This model allows for yielding behavior in the spring. When the force exceeds a certain threshold (yield force), the spring exhibits plastic behavior.
4. **Bending Spring/Damper**: Designed to resist lateral movements or bending. It computes the bending angle between two directions and applies an additional force if this angle exceeds a threshold.
5. **Pin Joint Spring/Damper**: Modified to work as a pin joint connection. The damping force is applied based only on the velocity of the second point.
6. **No Compression Spring/Damper**: This model does not resist compression. Forces are only computed when there's an effective displacement, i.e., when the spring is stretched.

3.2.2 Posts

The two points of each post are vertically interconnected by a pin joint spring/damper system, with its equilibrium length equivalent to the net's height. The bottom node is fixed in space at the position of the post foundation. To emulate the pin joint's restricted degrees of freedom in the y/z plane, the top node's lateral movement in the x direction is constrained by an auxiliary spring/damper mechanism.

3.2.3 Nets

The nets are modeled as a rectangular grid of nodes, following the simple modeling assumption shown in fig. 3. By default, a node is connected via springs to 4 neighbouring nodes (red), but additional springs can be added to the green nodes (8 connections) and up to the yellow nodes (12 connections), changing the behaviour of the net. At each timestep, the total forces at each node get computed using a yelding spring/damper.

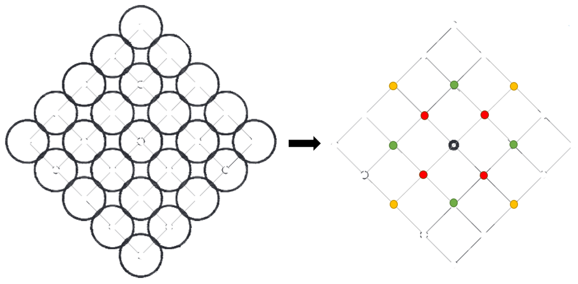

Figure 3. Model of the nets (adapted from [4]).

3.2.4 Ropes

Rope nodes are also interconnected using the spring/damper mechanism, as illustrated in fig. 4. This system can be enhanced with resistance to significant bending by employing either the bending or the 1D spring/damper. The ropes can be pre-tensioned by adjusting the equilibrium length to a specific percentage of the default rope segment length. The behavior and connections vary based on the rope type: Upslope ropes connect both to the posts at their endpoints and to the assumed vertical wall. Lateral support ropes have one end anchored to the wall, while the other links to a post. Bearing ropes are anchored at their extremities but can slide through the posts, with the sliding mechanism detailed later. During each timestep, forces on a node are calculated based on the displacements and velocities of its immediate preceding and succeeding nodes.

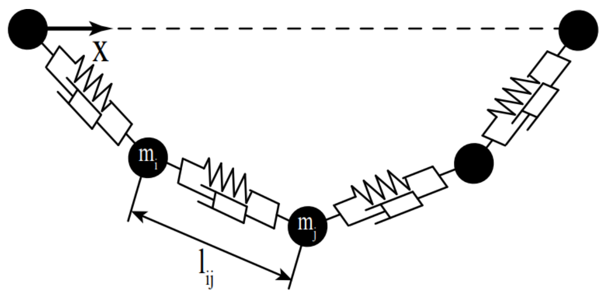

Figure 4. Model of the ropes (from [14]).

3.2.5 Rock Collision

In the rock collision logic, for each node in the net, the displacement between the node and the rock's center is computed. When the distance between a node and the rock's center is less than or equal to the rock's radius, a collision is detected. Upon this detection, a collision normal, derived from the direction of the displacement, is determined. The depth of penetration into the ball is then calculated as the difference between the ball's radius and the computed distance.

A collision response force is subsequently computed using again a spring/damper model. This time, the spring is proportional to the square of the penetration depth and directed along the collision normal. To account for energy loss during the collision, the damping force is applied in the direction of the collision normal based on the relative velocities of the rock and net nodes. To ensure the ball and net no longer overlap, the net node's position is adjusted to resolve the penetration.

3.2.6 Nets/Shackles Interaction

The nets are connected vertically to one another through shackles, that bind a node from one net with the corresponding node of the next net. A force is computed between these two nodes using the spring-damper without compression, with an equilibrium length set to $0.2cm$. This force effectively pulls the two nodes towards each other. The vertical shackle's position is set to the midpoint between these two net nodes, ensuring it remains centered between them. For the interaction between the net and the horizontal shackles, each horizontal shackle interacts via generic spring/damper with a specific node on the net. The nodes corresponding to each horizontal shackle are determined with the search algorithm during the components initialization process.

3.2.7 Shackles/Ropes Sliding Interaction

Modeling the sliding interactions is more complex, involving a multi-step process:

Identifying the Current Rope Segment: The segment of the rope on which the shackle currently resides is taken from the last time step through the appropriate index matrix. This segment is defined by two nodes of the rope, and its direction is given by:

$$\mathbf{d}_{\text{segment}} = \frac{\mathbf{p}_{\text{next}} - \mathbf{p}_{\text{current}}}{\|\mathbf{p}_{\text{next}} - \mathbf{p}_{\text{current}}\|}$$

where $\mathbf{p}_{\text{current}}$ and $\mathbf{p}_{\text{next}}$ are the positions of the current and next nodes of the rope, respectively.

Relative Velocity Calculation: The relative velocity of the shackle with respect to the rope is computed, subtracting the average velocity of the rope segment to the velocity of the shackle. This difference is projected onto the direction of the current rope segment to obtain the relative velocity:

$$v_{\text{relative}} = \mathbf{v}_{\text{difference}} \cdot \mathbf{d}_{\text{segment}}$$

where $\mathbf{v}_{\text{difference}}$ is the difference of the velocities of the rope segment and of the shackle.

Projected Movement: The potential distance the shackle would move in the current timestep, without any interruptions, is determined by multiplying the relative velocity by the timestep duration.

Transitioning to the Next Segment: If the shackle's projected movement exceeds the current segment's length, it transitions to the next segment, and the remaining distance to move is adjusted accordingly. The process is then reiterated from the beginning. Otherwise, the shackle remains on this segment, and its rope indices are stored in the index matrix for reference in the next timestep.

Friction Implementation: Friction between the shackle and the rope is modeled as a force opposing the shackle's movement. This force is proportional to the shackle's velocity and is given by:

$$\mathbf{F}_{\text{friction}} = -\mu \times \mathbf{v}_{\text{shackle}}$$

where μ is the friction coefficient. This frictional force reduces the shackle's velocity, simulating the resistance encountered during sliding.

Collision Handling: If the shackle approaches another shackle, a collision is detected. The overlap or penetration depth between the two shackles is calculated, and a correction is applied to ensure they remain separated. This correction is based on the direction of the current rope segment and the magnitude of the overlap.

Spring-Damper Interaction: To ensure the shackle remains attached to the rope, a spring-damper force is applied between the shackle and orthogonal projected position of the shackle on the rope segment. This force acts to pull the shackle towards the rope, counteracting any deviations due to external forces or the shackle's own momentum. The force is then distributed to the rope's nodes defining the current segment. Let t be the relative position of the shackle along the segment, calculated as:

$$t = \frac{(x_{shackle} - x_{start}) \cdot (x_{end} - x_{start})}{\|x_{end} - x_{start}\|^2}$$

where $x_{shackle}$, x_{start}, and x_{end} are the positions of the shackle, starting node, and ending node of the current rope segment, respectively. The forces applied to the starting and ending nodes of the segment are then:

$$F_{start} = (1 - t) \times F_{sd}$$

$$F_{end} = t \times F_{sd}$$

Through these calculations, the shackle is ensured to slide along the rope, responding to forces, friction, and potential collisions with other shackles.

3.2.8 Ropes/Posts Sliding Interaction

The interaction of ropes sliding through posts follows, for the moment, a simpler logic that than of the shackles. It still encompasses several steps:

Identyfing the current rope node: The rope segment interacting with the post top is taken from the indeces matrix. The algorithm then searches locally the previous and next 10 rope nodes to check if a new rope node is closer to the post top, and saves its index in the matrix for the next timestep.

Virtual Post Line and Sprind-Damper Interaction: Given that the spring damper would simply pull the rope node towards the top post node, countering the sliding, a line between the post node and its base on the x/y plane is computed. The rope node gets projected on this line and a spring/damper force is computed between the node and this projection. In this way, the computed force doesn't have a lateral x component and the rope will be able to slide past the post.

3.3 Graphical User Interface

The Graphical User Interface (GUI) leverages the internal capabilities of Taichi. The position matrices of all simulation elements get "flattened" and inserted in the 3D scene either as particles (shackles, rock), lines (ropes and posts) and meshes (nets), as visible in fig. 2. The GUI's inherent tools allow users to view the scene from multiple perspectives in real-time during the simulation and also offer the capability to capture and save the visual output in various format.

4 Discussion

The simulator currently achieves stability with timesteps on the order of magnitude of 1×10^{-5} seconds. While the system's behavior appears realistic, it's essential to note that it hasn't been calibrated using real-world experimental data and established material properties. The foundational assumptions in the logic and modeling are basic, serving as a preliminary starting point for the simulation.

- *Integration Method:* Transitioning to more advanced methods like the velocity Verlet or Runge-Kutta could enhance the simulation's stability and symplectic properties.

- *Net Model:* Given the DEM approach, there's potential to model the entire net discretely. This would involve representing each ring as a circle of interconnected points, mirroring a real net. With the right contact algorithms, this could lead to a highly realistic net simulation. To the author's understanding, no such model has been presented in existing literature.

- *Sliding Algorithms:* The current sliding algorithms don't seem to accurately capture the real-world "draping effect" of the net. Whether this is intertwined with the net modeling remains to be seen, but there's undoubtedly potential for refining the sliding logic.

- *Breaking Elements:* The simulator currently lacks energy dissipators. Incorporating them would be straightforward, requiring the isolation of end nodes of the bearing ropes and assigning them the desired stretch properties relative to the rope's anchor point.

- *Nonlinearity and Plastic Deformation:* While a nonlinear spring/damper model simulating plastic deformation exists, there's room for improvement. Presently, node displacements don't factor in plastic deformations.

- *FEM Posts:* To accurately simulate potential deformations, the posts could be represented using FEM mesh elements.

- *Arbitrary Rock Shapes:* With the DEM framework in place, rocks of any shape, represented as discretized points or spheres, can be simulated. This would necessitate modifications to the contact algorithm with the net in order to calculate the new interaction distances.

- *Force Measurement:* The current version lacks a system to measure forces at strategic barrier points. To do this, the force data calculated at every timestep at the desired nodes has to be stored. While early RBA prototypes had this feature, it's yet to be integrated into the current version.

By focusing on these areas, the simulator can be further refined, offering a more accurate and comprehensive representation of real-world scenarios.

5 Conclusions

Rockfall barriers are essential in regions prone to geological events. Despite many existing numerical models, certain facets of rockfall barrier mechanics remain ambiguous. This underscores a potential need for flexible, specifically tailored open-source simulation platforms.

This report introduced RockBarrierAnalytica (RBA), an open-source explicit Discrete Element Method (DEM) simulator developed in Taichi Lang. RBA stands out due to its simplicity, adaptability, and the integration of the Euler method with Taichi, enhancing model readability and accessibility.

Key features from RBA include:

- Efficient discretization of rockfall barrier components.

- Use of Euler's method for time integration.

- Leveraging Taichi's powerful parallelization capabilities for simulation.

- Comprehensive interaction logic for barrier components.

- An interactive graphical user interface for real-time simulation visualization.

Nevertheless, RBA is still in development and comes with its set of limitations. It has yet to be calibrated with real-world data. Moreover, it currently omits the typical braking elements of rockfal barriers and needs to better integrate aspects like plastic deformations. Additionally, areas like the sliding algorithms and net representations demand further refinement.

In summary, RBA offers a new foundation for rockfall barrier simulations. Its open-source nature invites community enhancements. With further refinements and real-world feedback, RBA could evolve into a new tool for rockfall barrier research and optimization.

6 Code Availability

The source code and documentation for RockBarrierAnalytica are openly accessible and can be found on GitHub at the following link:
`https://github.com/igasparini/RockBarrierAnalytica`
Contributions are welcomed.

References

[1] A. Volkwein, K. Schellenberg, V. Labiouse, F. Agliardi, F. Berger, F. Bourrier, L. K. A. Dorren, W. Gerber, and M. Jaboyedoff. Rockfall characterisation and structural protection – a review. *Natural Hazards and Earth System Sciences*, 11(9):2617–2651, September 2011.

[2] C. Gentilini, G. Gottardi, L. Govoni, A. Mentani, and F. Ubertini. Design of falling rock protection barriers using numerical models. *Engineering Structures*, 50:96–106, May 2013.

[3] A. Volkwein. *Numerische Simulation von flexiblen Steinschlagschutzsystemen*. PhD thesis, ETH Zurich, 2004.

[4] C. Gentilini, L. Govoni, S. De Miranda, G. Gottardi, and F. Ubertini. Three-dimensional numerical modelling of falling rock protection barriers. *Computers and Geotechnics*, 44:58–72, June 2012.

[5] J.P. Escallón, V. Boetticher, C. Wendeler, E. Chatzi, and P. Bartelt. Mechanics of chain-link wire nets with loose connections. *Engineering Structures*, 101:68–87, October 2015.

[6] L. Castanon-Jano, E. Blanco-Fernandez, D. Castro-Fresno, and D. Ferreño. Use of explicit FEM models for the structural and parametrical analysis of rockfall protection barriers. *Engineering Structures*, 166:212–226, July 2018.

[7] L. Zhao, Z. Yu, Y. Liu, J. W. He, S. L. Chan, and S. C. Zhao. Numerical simulation of responses of flexible rockfall barriers under impact loading at different positions. *Journal of Constructional Steel Research*, 167:105953, April 2020.

[8] F. Nicot, B. Cambou, and G. Mazzoleni. From a constitutive modelling of metallic rings to the design of rockfall restraining nets. *International Journal for Numerical and Analytical Methods in Geomechanics*, 25(1):49–70, January 2001.

[9] D. Bertrand, A. Trad, A. Limam, and C. Silvani. Full-Scale Dynamic Analysis of an Innovative Rockfall Fence Under Impact Using the Discrete Element Method: from the Local Scale to the Structure Scale. *Rock Mechanics and Rock Engineering*, February 2012.

[10] K. Thoeni, A. Giacomini, C. Lambert, S. W. Sloan, and J. P. Carter. A 3D discrete element modelling approach for rockfall analysis with drapery systems. *International Journal of Rock Mechanics and Mining Sciences*, 68:107–119, June 2014.

[11] K. B. Sautter, H. Hofmann, C. Wendeler, P. Wilson, P. Bucher, K. U. Bletzinger, and R. Wüchner. Advanced Modeling and Simulation of Rockfall Attenuator Barriers Via Partitioned DEM-FEM Coupling. *Frontiers in Built Environment*, 7:659382, June 2021.

[12] EOTA. EAD 340059-00-0106. European Assessment Document: Falling Rock Protection Kits, 2018.

[13] R. Baumann and W. Gerber. *Grundlagen zur Qualitätsbeurteilung von Steinschlagschutznetzen und deren Fundation. Anleitung für die Praxis*. Number 1805 in Umwelt-Wissen. Bundesamt für Umwelt, Bern, 2018.

[14] C. Schenk, C. Masone, P. Miermeister, and H. H. Bulthoff. Modeling and analysis of cable vibrations for a cable-driven parallel robot. In *2016 IEEE International Conference on Information and Automation (ICIA)*, pages 454–461, Ningbo, China, August 2016. IEEE.