Robotic Mushroom Harvesting with Real2Sim2Real and Model Predictive Path Integral (MPPI) based Planning

Konstantinos Vasios[1,2], Antonis Porichis[1,2], Vishwanathan Mohan[1] and Panagiotis Chatzakos[1]

***Abstract*—We present a strategy for the problem of robotic button mushroom harvesting (Agaricus Bisporus) that involves a Real2Sim2Real pipeline with dynamic scene reconstruction and a Model Predictive Path Integral (MPPI) control & planning architecture for generating optimal uprooting motion primitives based on a physics engine simulation framework. Given the complex, nonlinear, anisotropic material properties of the mushrooms in combination with the multiple failure-mode modalities involved, we design a simulation framework around the PyBullet rigid-body physics engine by utilizing first-order approximations of the equivalent continuum mechanics models. By exploiting the computational efficiency of the aforementioned simulation framework, we directly apply the MMPI control framework to generate offline optimal mushroom uprooting motion primitives, defining a set of cost objectives for an optimal and within-constraint harvesting plan. We show that with this planning strategy, the "root-bending" action emerges autonomously for the single mushroom case as an optimal uprooting maneuver, which corresponds well to empirical knowledge obtained by expert pickers. A video demonstration of the proposed architecture can be found in https://youtu.be/k38ePBsBego.**

I. Introduction

Agricultural robotics is a rapidly advancing field in academia-research and commerce due to multiple synergistic effects on recent scientific and technological breakthroughs, most notably in perception and planning, in combination with hardware cost and availability [1], [2]. In particular, the task of mushroom harvesting is an exemplary case for demonstrating advanced perception, long-horizon planning, and dexterous robotic manipulation capabilities, as it simultaneously involves scene decluttering based on long-horizon Task and Motion Planning (TAMP) with forceful manipulation and grasp synthesis for purposeful fracturing, given a set of cost objectives and operational constraints necessary for an optimal crop yield. As is usually the case to avoid bruises in the case of grasping and manipulation of delicate produce, soft grippers are necessary, which further increases the complexities around perception, modeling, and control.

Our strategy for tackling this complex and multifaceted problem revolves around the development of a simulation framework with the PyBullet physics engine to capture the dominant dynamics involved by first-order approximations of the mushroom and gripper overall system. By doing so, we can take advantage of a rich set of tools & capabilities that the physics engine provides, such as Newton-Euler governing motion dynamics, collision detection between scene objects, and friction models coupled with RGB-D rendering with scene segmentation. As we are dealing with complex, nonlinear dynamics with a high degree of uncertainty, like object deformations, failure modes, and potentially soft-gripper actuation with the additional requirement for fast computation for allowing data-driven and MPC-like techniques, we are utilizing first-order approximations of the equivalent continuum mechanics models. For the derivation of the modeling parameters, we conducted real-world experiments for the characterization of mushroom stiffness with force sensors and the phasespace tracking system.

To extract the necessary scene semantics for an on-the-fly physics simulation scene instantiation (Real2Sim) we capture a set of depth images from a set of different poses which we then integrate with the Iterative Closest Point (ICP)

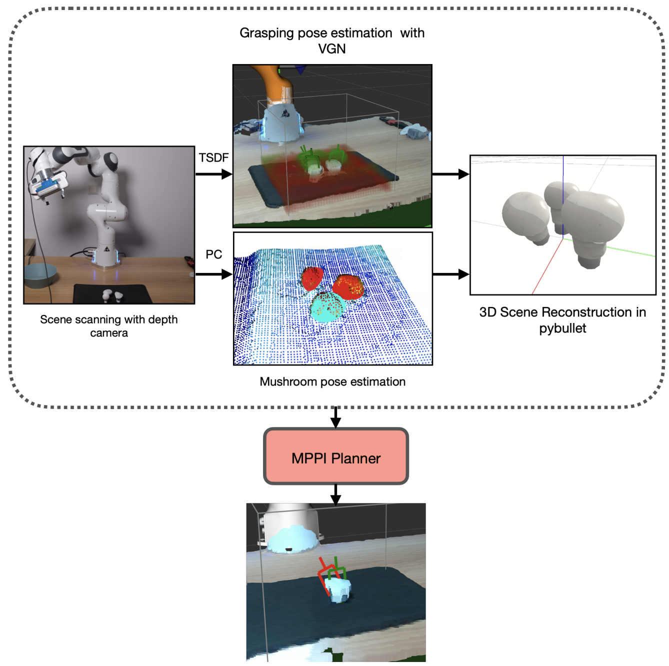


Fig. 1: Scheme of the proposed architecture for mushroom harvesting.

[1]AI Innovation Centre, University of Essex, Granta Park, Great Abington, Cambridge CB21 6AL {k.vasios, a.porichis, p.chatzakos}@essex.ai, vishwanathan.mohan@essex.ac.uk

[2]National Structural Integrity Research Centre, Granta Park, Great Abington, Cambridge CB21 6AL

This research has received funding from the EU's Horizon 2020 research and innovation programme, under grant agreement No 101017054.

for generating a Point Cloud (PC) and a Truncated Signed Function (TSDF) equivalent representation. We utilize these representations twofold, first to estimate the mushroom poses, using the pipeline developed by Mavridis *et al.* [3] and second for selecting a good mushroom candidate for grasping together with its close surrounding clutter for an on-the-fly generation of a simulation instance which acts as scene reconstruction and a model for our system where we perform the planning reasoning. For the mushroom & the consequent region selection, we make use of the pre-trained Volumetric Grasping Network (VGN) [4] which can generate a set of grasping pose candidates for scene decluttering.

Based on the scene reconstruction with the simulation framework, we can now reason in an offline manner to generate an optimal mushroom uprooting strategy. We utilize the MPPI control framework as a simple, yet effective [5]–[7] framework to solve the task of optimal mushroom picking in simulation and to carry out subsequent rollouts of the set of critical waypoints of the end-effector in the real-world setup. This framework allows for a definition of cost objectives for task shaping together with collision avoidance and grasping force minimization. It is also gradient-free, which fits well in the context of the inherent hybrid dynamic modalities involved in grasping & manipulation dynamics.

Our paper makes the following contributions:

- A dynamic simulation framework for mushroom crop harvesting with a soft or rigid robotic gripper with first-order continuum mechanics approximations, with parameter identification based on real-world mushroom characterization (SysId).
- A Real2Sim 3D scene reconstruction with embedded physics based on mushroom pose estimation and the aforementioned simulation framework.
- An offline planner based on the Volumetric Grasping Network (VGN) for mushroom selection and MPPI for generating optimal mushroom picking primitives, which despite being a local planner subject to potentially stuck in local minima, is able to generate composite picking actions that correspond well to empirical evidence.

II. Related Work

Previous attempts have been made on white button mushroom harvesting with robotic actuators [8]–[12] where the system involves a camera-based system for mushroom localization and a set of predefined picking motion primitives for actuation of the end effector. The bending motion has been reported to be a critical factor in improving the overall success rate of harvesting.

The definition of the mechanical properties of the mushroom root system has been the subject of previous studies. In the works of A.McGarry [13] and Bohdziewicz et al. [14] the anisotropic elastic properties of the mushroom stiffness are studied and experimental values for the Young Elastic Modulus and strain are derived. Similar methodologies for crop characterization can be found in other types of produce [15].

Similar challenges arise in other types of produce in agriculture. Wei et al. [16] where an adaptive impedance controller has been developed for apple harvesting, following an analytical modeling and characterization process. Bu et al. [17] study optimal picking patterns for apple harvest based on an analytical Finite Element Model (FEM). Junge et al. [18] develop a strategy based on human demonstrations on a physical twin for the harvest of raspberries. Mehta et al. [19] present a visual servo controller for autonomous citrus harvesting.

The task of mushroom harvesting can be examined as an attempt at purposeful fracturing of an object with a robotic actuator, which seems to be a surprisingly underexplored topic in robotics research on manipulation. The work of M. Abdeetedal [20] seems to be the single instance that studies this specific problem and is based on material failure theories and optimization techniques on grasp quality measures.

III. Method

An overview of the overall system architecture is presented in Figures 1, 6, and Alg. 1.

Algorithm 1 Autonomous Mushroom Harvesting

while *mushroomsPosesDetected* **do**
  $\{\ \mathrm{D_{imgs}}\} \leftarrow \mathit{sceneScan}()$
  $\mathrm{PC},\ \mathrm{TSDF} \leftarrow \mathit{get3DPointCloudWithICP}(\{\ \mathrm{D_{imgs}}\})$
  $\mathrm{MP},\ \mathrm{MS} \leftarrow \mathit{mushroomPoseEstimation}(\ \mathrm{PC})$
  $\mathrm{GP},\ \mathrm{GPR} \leftarrow \mathit{graspingPosesEstimation}(\ \mathrm{TSDF})$
  $\mathrm{TMP},\ \mathrm{SMP} \leftarrow \mathit{selectTargetMushroom}(\ \mathrm{MP},\ \mathrm{GP})$
  $\mathrm{Sim} \leftarrow \mathit{instantiateMasterSimulation}(\ \mathrm{TMP},\ \mathrm{SMP})$
  $\mathrm{GWP} \leftarrow \mathit{graspingManueverWithMPPI}(\ \mathrm{Sim})$
  $\mathrm{TRJ} \leftarrow \mathit{trajectoryGenWithRRTandIK}(\ \mathrm{GWP})$
  $\mathit{rolloutTrajectoryToTheRealWorld}(\ \mathrm{GTR})$
end while

During the first step of the pipeline, a subregion of the mushroom crop is scanned to obtain a set of depth images $\{\ \mathrm{D_{imgs}}\}$ from various poses. The 3D point cloud and equivalent TSDF representations are then obtained with the ICP method and the open3D library. We can then obtain a set of mushroom pose estimations with [3] and a set of grasping pose candidates with respective VGN score rankings [4]. We select the mushroom with the highest grasping score to be picked on the basis of the Euclidean distance between the grasping candidate from VGN and the mushroom pose estimates. By instantiating the "master simulation" scene equivalent and the MPPI planner we can extract a set of waypoints for the gripper which are descriptive of the optimal picking maneuver. We finally deploy to the real Franka robot by feeding the generated waypoints to the MoveIt! library which generates the final joint trajectories with RRT and inverse kinematics.

In the following two subsections *A* and *B*, we present the implementations of our main original contributions, namely the mushroom modeling & characterization and the MPPI planner design.

A. Mushroom Modeling & Characterization

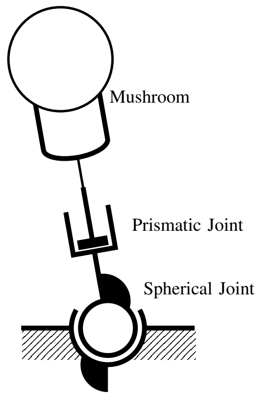


Fig. 2: Mushroom root kinematic model.

An analytical modeling of the mushroom root system introduces a set of challenging requirements, given the anisotropic elastic material in combination with the failure modes that must be predicted during execution. In addition, we add the requirement of fast execution time for fast evaluations for potential utilization of Reinforcement Learning (RL) and Model Predictive Control (MPC) techniques. Ideally, for this type of system, we would choose an analytical finite element method (FEM), but the execution time would be prohibitively slow. Given the empirical knowledge that ultimately the mushroom-picking maneuvers from human experts seem to distill around a basic set of primitives, namely “bending”, “twisting”, and “pulling” around the root, we suggest that capturing a set of first-order approximations of the continuum mechanics models of the mushroom root system should suffice. Based on the work found in [20]–[22], we result in a mushroom root model as a combination of a spherical and a prismatic joint, controlled with a fixed position PD law to emulate the stiffness and damping characteristics (see Fig. 2) of elastic deformation. To calculate the stress threshold necessary for material failure and the consequent detachment of the mushroom from the ground, the VonMises criterion is utilized. Lastly, instead of incorporating the complete robot model, we only include the gripper by defining a fixed constraint in the flange frame in the 3D space, as suggested in [4]. For the implementation of the proposed model, we rely on the PyBullet rigid multibody dynamics physics engine, as it easily provides the ability to add and remove fixed constraints between objects during run-time.

The axial, rotational, and bending stiffness components of an isotropic cylindrical rod representing a mushroom-root model are defined as follows.

$$K_{\text{pull}} = \frac{EA}{l}, \qquad K_{\text{rot}} = \frac{GJ}{l}, \qquad K_{\text{bend}} = \frac{GA}{l} L^2 \tag{1}$$

Where E, G are the Young and Shear Modulus terms and A, I, L are the axial surface area, second moment of area, and length, respectively. For bending, the definition is for transversal action given the relatively short length of the mushroom stem. The VonMises stress factor for a cylindrical rod under shear and axial stress is defined as follows.

$$\sigma_{VM} = \left(\sigma_x^2 + 3\tau_{zx}^2\right)^{1/2} \tag{2}$$

Failure and therefore mushroom yielding will occur when the VonMises stress factor exceeds the yielding stress factor value, namely:

$$\sigma_{VM} > S_y \tag{3}$$

Regarding the damping terms of the PD controllers, we manually tune them in order to approximate critical damping characteristics.

To tune the stiffness parameters and determine the yielding stress threshold, we proceed with a series of experiments. We attach a Touchence Shokac chip force sensor to the index, middle fingers, and thumb of an expert mushroom picker and a Phasespace-led tracker to the mushroom cap to determine the relative displacement. The goal here is to identify the stress-strain diagrams for distinctive “pulling”, “bending”, and “twisting” motions (see Fig. 4). A similar approach has been adopted by Huang et al. [12] where an IMU unit was used for position tracking. In practice, we found that by setting values for the Young Modulus that resemble the real world, namely, on the order of magnitude of 0.5 MPa, the system becomes too stiff, which in turn requires a sample rate dt for the physics engine at 0.1 ms for ensuring stability, rendering the simulation $2x$ slower than the real world clock time. As the design purpose of this simulation as a dynamic model is to derive a kinodynamic plan with the motion primitives for mushroom picking, we can scale the stiffness to the order of magnitude of 10kPa, resulting in a kinodynamically equivalent model that is $10x$ faster than the real-world clock time.

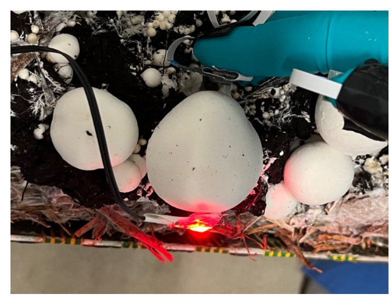
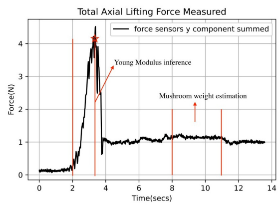
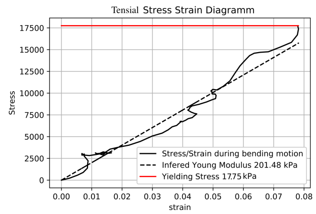


Fig. 3: Experimental configuration for root stiffness characterization during bending action with a Touchence Shokac Chip force sensor attached on the finger that will perform the action, and the Phasespace tracker led for measuring the total displacement.

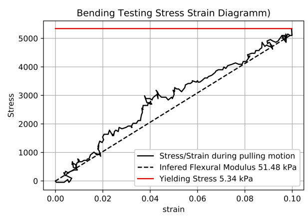
Lastly, we found that the bending action resulting from the lineal isotropic model defined above is too stiff compared to the experimental results, as also confirmed in [13]. We simply readjust to emulate the anisotropic properties as $K_{\text{bend adj}} = 1/5 K_{\text{bend}}$.



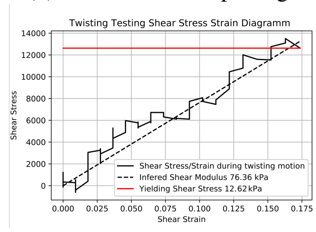
(a) Total pulling force over time.



(b) Stress-Strain - "pulling"



(c) Stress-Strain - "bending"



(d) Stress-Strain - "twisting"

Fig. 4: Stress - Strain experiments results for mushroom root stress characterization (SysId). We derive critical modeling parameters for the Young modulus, Poisson ratio, and yielding stress with simple linear regression over the close to linear stress-strain regions.

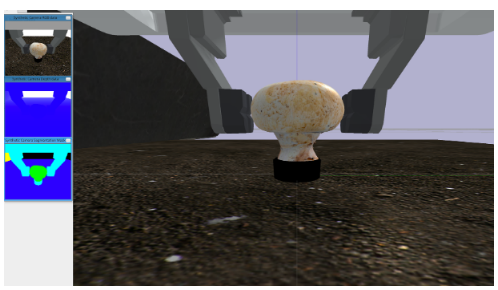


Fig. 5: The PyBullet debug GUI view of the simulation framework for mushroom harvesting.

B. Optimal uprooting planning with MPPI

To generate an optimal mushroom picking planning, we base our implementation on the MPPI controller in [6]. Our simulation framework is implemented in PyBullet instead of NVIDIA's Isaac gym, as we make use of the active constraint removal (for mushroom detachment) during the simulation run-time, which is not available in the latter. We modify therefore the implementation for sampling from our simulation environment in parallel in CPU as PyBullet is not GPU parallelizable. This modification does not allow for real-time control; therefore, we do use a "master" simulator where we can extract a set of waypoints, rendering the system, in essence, an offline planner. This subsampling methodology has the added benefit of denoising the trajectory, which is a known issue with MPPI controllers. For an overview of the proposed implementation, see Fig. 6.

To describe the mushroom picking task, we define a set of cost functions 5,6,7.

TABLE I: Mushroom Model Parameters Definitions

Mushroom Model Parmeters	Values
pose	$[p_x, p_y, p_z, \omega_x, \omega_y, \omega_z]$ m, rad
mass	m Kg
cap/stem diameter & height	$[c_d, c_h, s_d, s_h]$
young modulus	Y_m Pa
poisson ratio	P_r
yielding stress	Y_s Pa
cap contact stiffness	c_k N/m
cap contact damping	c_d Ns/m
cap restitution	c_r
cap lateral friction	c_c
cap spinning friction	c_{sc}

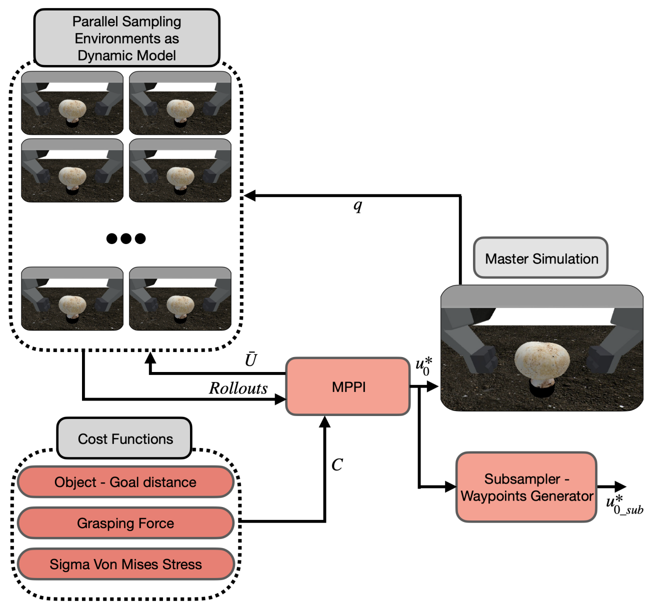


Fig. 6: MPPI planner for mushroom picking based on the MPPI implementation in [6].

$$C_{\text{pick}} = C_{\text{dist}} + C_{\text{force}} + C_{\text{stress}} \tag{4}$$

$$C_{dist} = \omega_d \|p_{EE} - p_M\| + \omega_{M_p} \|p_G - p_M\| \tag{5}$$

The C_{dist} incentivizes the interaction between the gripper and the mushroom cap and describes its desired goal position, which in our case is just an offset by the z-axis.

$$C_{\text{force}} = \begin{cases} \omega_f \sum f_{fingers}, & \text{if } \sum f_{fingers} < f_{\text{safe}} \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

C_{force} should penalize actions that are outside the safe limits in an attempt to avoid bruising and, consequently, destruction of the produce. In practice $f_{\text{safe}} = 3$ N.

$$C_{\text{stress}} = -\omega_\sigma \sigma_{VM} \tag{7}$$

The C_{stress} cost term incentivizes actions that will cause stress on the root and ultimately failure for the mushroom to be picked. We found this cost term to be necessary to allow the MPPI controller to discover the bending action, as it would otherwise be stuck to local minima.

For the current demonstrator, we have avoided adding a collision-cost term, as this would be necessary in a cluttered environment. Given the aforementioned compromise of a CPU parallelization instead of a GPU one, we found in practice that including collision detection between multiple mushroom instances was prohibitively expensive and impractical in terms of computation. This is within the scope of future work, which should include a GPU parallelizable physics engine that also allows for the removal of fixed constraints during runtime for material failure-mode emulation.

TABLE II: Cost function weights

Cost function weights	**Values**
ω_d	8
ω_{M_p}	12
ω_f	0.005
ω_σ	0.0001

TABLE III: MPPI parameters tuning

MPPI parameters	**Values**
Sampling Environments Number	60
Planning Horizon	20 steps
Sampling Method	Halton Spline
Sampling Noise σ	$[diag(0.1), 0.05]$
Rollout var discount λ	0.95

TABLE IV: PyBullet simulation parameters tuning

Simulation Parameters	**Values**
Controller sampling time	0.01 sec
Physics engine *dt*	0.001 sec
Solver iterations	10
Friction ERP	0.02

IV. Experiments

We start by testing the efficacy of the simulation framework in combination with the MPPI planner with an AB test purely in simulation. In case A, we set the friction c_c of the mushroom cap at an abnormally high value $c_c = 1.0$ where the development of high shear grasping forces should overcome the axial root resistance, allowing for a direct pulling motion within the desired constraints. We repeat the experiment in case B, under the same configurations, only this time setting the friction of the mushroom cap to a more realistic value $c_c = 0.3$, where a pure pulling motion would only be possible by exerting high lateral grasping forces, which would exceed the safe thresholds or would even be impossible to develop. As we see in Fig. 7 our hypothesis is satisfied and the MPPI planner discovers autonomously the bending action to successfully detach the mushroom within the desired constraints. The total planning time with an Intel i5 13600K CPU and the configurations as described in Tables IV and III was in both cases 70 real world secs which corresponds to 1.4 'master' simulation *secs*.

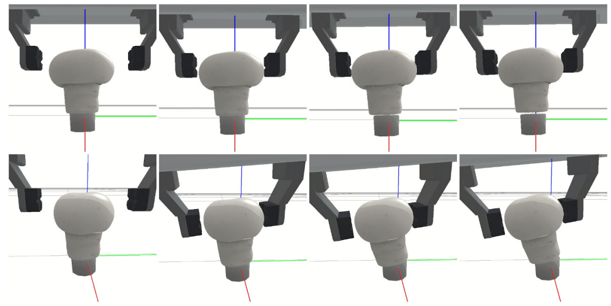


Fig. 7: In the first row we observe the picking attempt of the MPPI planner when we set an abnormally high friction value at $c_c = 1.0$ at the mushroom cap. In that case, a direct pull motion is emerging because the necessary high shear forces can be easily generated. In the second row, we observe the result where we set a lower friction value $c_c = 0.3$ where the MPPI planner autonomously discovers the bending motion as the optimal picking strategy.

We finally deploy the generated trajectory of the B test by first subsampling, in order to generate a set of key-waypoints for the gripper pose. We manually choose the gripper pose when contact of both fingers is initiated with the mushroom cap, and the pose at the precise moment when the mushroom is detached. We then feed the waypoints directly to the MoveIt! group commander for generating the final trajectory with RRT and handling the inverse kinematics. Automated extraction of an optimal set of waypoints to accurately describe the picking maneuver is within the scope of future work.

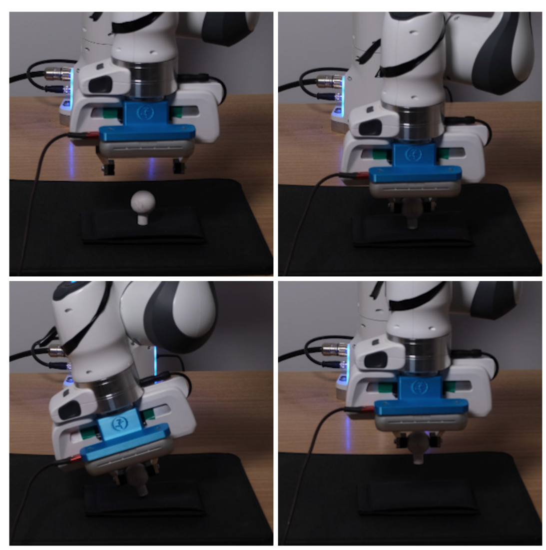


Fig. 8: The bending action as transferred to the real-world setup with the Franka Robot and a 3D printed mushroom-like shape.

V. Conclusion

We have demonstrated a pipeline for mushroom harvesting with a robotic gripper, based on on-the-fly 3D scene reconstruction with a physical simulation and an offline MPPI planner for generating optimal uprooting manipulation maneuvers. We show that despite the lumped system dynamic model and the CPU parallelization limitations of the MPPI planner, the system manages to discover optimal actions for mushroom picking. This is a sense-plan-act open-loop architecture; namely, it is not reactive to unpredictable events due to environmental changes and failures due to modeling discrepancies. Therefore, the next step is to close the loop, which means that the pass-through computation should be achieved in real time. We can do so through GPU parallelization [6] and an on-the-fly perception module for mushroom state estimation that includes pose and root stiffness. This would also allow for a significant increase in the total number of sampling environments and the planning horizon, resulting in more complex and robust grasping strategies.

Acknowledgment

This publication was made possible by the sponsorship and support of Lloyd's Register Foundation (LRF). The work was enabled through, and undertaken at the National Structural Integrity Research Centre (NSIRC), a postgraduate engineering facility for industry-led research into structural integrity established and managed by TWI through a network of both national and international Universities.

We would like to thank Andrea Uccello and Helen Grogan from Teagasc (Oak Park, Carlow, R93 XE12, IE) for their support and expert knowledge of mushroom harvesting.

References

[1] L. F. Oliveira, M. Silva, and A. Moreira, *Agricultural Robotics: A State of the Art Survey*, Aug. 2020, pages: 286.

[2] L. Droukas, Z. Doulgeri, N. L. Tsakiridis, D. Triantafyllou, I. Kleitsiotis, I. Mariolis, D. Giakoumis, D. Tzovaras, D. Kateris, and D. Bochtis, "A Survey of Robotic Harvesting Systems and Enabling Technologies," *Journal of Intelligent & Robotic Systems*, vol. 107, no. 2, p. 21, Jan. 2023. [Online]. Available: https://doi.org/10.1007/s10846-022-01793-z

[3] P. Mavridis, N. Mavrikis, A. Mastrogeorgiou, and P. Chatzakos, "Low-cost, accurate robotic harvesting system for existing mushroom farms," in *2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, Jun. 2023, pp. 144–149, iSSN: 2159-6255. [Online]. Available: https://ieeexplore.ieee.org/document/10196219

[4] M. Breyer, J. J. Chung, L. Ott, R. Siegwart, and J. Nieto, "Volumetric Grasping Network: Real-time 6 DOF Grasp Detection in Clutter," Jan. 2021, arXiv:2101.01132 [cs]. [Online]. Available: http://arxiv.org/abs/2101.01132

[5] T. Howell, N. Gileadi, S. Tunyasuvunakool, K. Zakka, T. Erez, and Y. Tassa, "Predictive Sampling: Real-time Behaviour Synthesis with MuJoCo," Dec. 2022, arXiv:2212.00541 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2212.00541

[6] C. Pezzato, C. Salmi, E. Trevisan, J. Alonso-Mora, and C. H. Corbato, "Sampling-Based MPC Using a GPU-parallelizable Physics Simulator as Dynamic Model: an Open Source Implementation with IsaacGym."

[7] M. Bhardwaj, B. Sundaralingam, A. Mousavian, N. Ratliff, D. Fox, F. Ramos, and B. Boots, "STORM: An Integrated Framework for Fast Joint-Space Model-Predictive Control for Reactive Manipulation," Sep. 2021, arXiv:2104.13542 [cs]. [Online]. Available: http://arxiv.org/abs/2104.13542

[8] J. N. Reed and R. D. Tillett, "Initial experiments in robotic mushroom harvesting," *Mechatronics*, vol. 4, no. 3, pp. 265–279, Apr. 1994. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0957415894900043

[9] R. Noble, J. N. Reed, S. Miles, A. F. Jackson, and J. Butler, "Influence of Mushroom Strains and Population Density on the Performance of a Robotic Harvester," *Journal of Agricultural Engineering Research*, vol. 68, no. 3, pp. 215–222, Nov. 1997. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021863497901955

[10] J. N. Reed, S. J. Miles, J. Butler, M. Baldwin, and R. Noble, "AE—Automation and Emerging Technologies: Automatic Mushroom Harvester Development," *Journal of Agricultural Engineering Research*, vol. 78, no. 1, pp. 15–23, Jan. 2001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021863400906292

[11] M. Huang, X. Jiang, L. He, D. Choi, J. Pecchia, and Y. Li, "Development of a Robotic Harvesting Mechanism for Button Mushrooms," *Transactions of the ASABE*, vol. 64, pp. 565–575, Jan. 2021.

[12] M. Huang, L. He, D. Choi, J. Pecchia, and Y. Li, "Picking dynamic analysis for robotic harvesting of Agaricus bisporus mushrooms," *Computers and Electronics in Agriculture*, vol. 185, p. 106145, Jun. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168169921001630

[13] A. McGarry and K. Burton, "Mechanical properties of the mushroom, Agaricus bisporus," *Mycological Research*, vol. 98, no. 2, pp. 241–245, Feb. 1994. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0953756209801925

[14] J. Bohdziewicz, G. Czachor, and P. Grzemski, "Anisotropy of mechanical properties of mushrooms (Agaricus bisporus (J.E. Lange) Imbach)," *Inżynieria Rolnicza*, vol. R. 17, nr 4, t. 2, 2013. [Online]. Available: http://yadda.icm.edu.pl/baztech/element/bwmeta1.element.baztech-8bb4c8c2-307f-4680-aa54-65da0d0d33e2

[15] A. Roshanianfard and N. Noguchi, "Characterization of pumpkin for a harvesting robot," *IFAC-PapersOnLine*, vol. 51, no. 17, pp. 23–30, Jan. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896318311728

[16] J. Wei, D. Yi, X. Bo, C. Guangyu, and Z. Dean, "Adaptive Variable Parameter Impedance Control for Apple Harvesting Robot Compliant Picking," *Complexity*, vol. 2020, p. e4812657, Apr. 2020, publisher: Hindawi. [Online]. Available: https://www.hindawi.com/journals/complexity/2020/4812657/

[17] L. Bu, G. Hu, C. Chen, A. Sugirbay, and J. Chen, "Experimental and simulation analysis of optimum picking patterns for robotic apple harvesting," *Scientia Horticulturae*, vol. 261, p. 108937, Feb. 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0304423819308234

[18] K. Junge, C. Pires, and J. Hughes, "Lab2Field transfer of a robotic raspberry harvester enabled by a soft sensorized physical twin," *Communications Engineering*, vol. 2, no. 1, pp. 1–11, Jun. 2023, number: 1 Publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/s44172-023-00089-w

[19] B. Mehta, M. Diaz, F. Golemo, C. J. Pal, and L. Paull, "Active Domain Randomization," Jul. 2019, arXiv:1904.04762 [cs]. [Online]. Available: http://arxiv.org/abs/1904.04762

[20] M. Abdeetedal and M. R. Kermani, "Grasp synthesis for purposeful fracturing of object," *Robotics and Autonomous Systems*, vol. 105, pp. 47–58, Jul. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S092188901730266X

[21] C. Della Santina, C. Duriez, and D. Rus, "Model Based Control of Soft Robots: A Survey of the State of the Art and Open Challenges," *IEEE Control Systems*, vol. 43, no. 3, pp. 30–65, Jun. 2023, arXiv:2110.01358 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2110.01358

[22] M. A. Graule, T. P. McCarthy, C. B. Teeple, J. Werfel, and R. J. Wood, "SoMoGym: A Toolkit for Developing and Evaluating Controllers and Reinforcement Learning Algorithms for Soft Robots," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4071–4078, Apr. 2022, conference Name: IEEE Robotics and Automation Letters.