Robotik Pages

Karla Schramm
Hochschule für Angewante Wissenschaften Hof
Hof, Germany
kschramm2@hof-university.de

I. Robot Execution Layer

A. Design Constraints and Integration Rationale

The robot execution layer was developed as part of a larger Sense–Plan–Act pipeline, making subsystem decoupling a primary objective. In particular, introducing additional vision dependencies into the execution layer would have increased integration complexity and risk, especially during system integration. Therefore, the execution component was designed to operate without a dedicated camera-based localization module beyond the perception subgroup.

The physical setup imposed further constraints. The employed game set exhibited noticeable manufacturing tolerances, including non-uniform chip heights, which complicate stable grasping. Although a magnetic end-effector can increase robustness under such variability, mechanical grasping was retained to ensure compatibility with standard, non-magnetic playing pieces. Consequently, the execution pipeline emphasizes (i) lightweight geometric calibration from a small set of reference measurements, (ii) a conservative approach-from-above motion primitive to reduce collision risk and stack disturbance, and (iii) empirical tuning of grasp parameters to improve robustness under piece-height variation.

B. Geometric Board Calibration and Target Pose Calculation

The vendor-provided PyNiryo API [1] was used as the primary control interface. The robot is commanded in task space using Cartesian end-effector poses (x, y, z, roll, pitch, yaw), represented in code via the PoseObject data structure [2]. Translational coordinates are specified in meters and orientations in radians. End-effector motion is executed in the robot's base coordinate frame using move_pose(PoseObject).

The get_pose() function returns the current end-effector pose directly in this base frame. As a result, the pose parameters are immediately suitable for calibration, motion target generation, and execution, and no additional coordinate-frame conversion is required.

The first calibration point O serves as the origin. The second and third points, B and C, define the board's width and height directions, respectively. Two vectors are computed from the origin, $w = B - O$ and $h = C - O$. Since the relative layout of Nine Men's Morris is known and fixed, each playable position can be expressed by predetermined scalar coefficients (u_i, v_i) indicating how far to move along (i.e., how strongly to scale) the width and height vectors [3]. The target position for board index i is then computed as

$$P_i = O + u_i w + v_i h, \tag{1}$$

where u_i and v_i are taken from a lookup table.

In our implementation, the mapping is applied in the x –y plane only, and the height is kept constant by setting $z := O_z$. The value O_z is determined during calibration, either as an average height derived from the recorded reference points (e.g., using their mean) or as a predefined constant specified in niryo_config.toml for even more reliability if the board height is known.

C. Robust Motion Execution via a Hover–Descend–Retreat Pattern

```
procedure ViaAbove(p, action)                    pseudocode
    MovePose(AtZ(p, SAFE_Z))        } (1) hover
    MovePose(p)                     } (2) descend
    action()                        } (3) grasp/release
    MovePose(AtZ(p, SAFE_Z))        } (4) retreat
end procedure
ReleaseWithTool()
ViaAbove(src, GraspWithTool)
ViaAbove(dst, ReleaseWithTool)
```

Listing 1. Approach-from-above pick-and-place routine (pseudocode).

The executed motion strategy is summarized in Listing 1 as pseudocode. Each manipulation step follows a fixed four-phase primitive (hover → descend → actuate → retreat) using a predefined safe height (equal to the configured idle pose height). This design reduces collision risk and improves repeatability, since horizontal movements are performed only at a clearance height. In particular, the approach-from-above strategy avoids sweeping motions near the board surface and reduces the likelihood of disturbing adjacent pieces or destabilizing stacks during pick-and-place operations.

D. Grasp Parameterization and Empirical Validation

Reliable manipulation was challenged by non-uniform chip heights originating from manufacturing tolerances of the physical game set. To mitigate sensitivity to such variation in chip-height without introducing additional perception dependencies, grasping was treated as a parameterized routine with configuration values stored in niryo_config.toml. The most relevant parameters include the nominal chip height (chip_height), an optional fixed board height (fixed_z), and small vertical clearance offsets for pickup and placement (pick_z_offset, place_z_offset). These offsets implement controlled over-travel during pickup and a gentler approach during placement, improving tolerance to piece-height variability.

Parameter settings were validated using a dedicated stress-test routine (stresstest.py) designed to exercise the pick-and-place pipeline under representative operating conditions. The test executes a fixed sequence of 20 ned2.move operations, covering (i) repeated stack-to-board placements, (ii) repeated board-to-stack removals up to the maximum expected stack height, and (iii) additional board-to-board transfers for coverage. Throughout the procedure, the end-effector follows the approach-from-above motion primitive (Listing 1), thereby restricting lateral motion to a predefined clearance height. The routine was used iteratively to tune the grasp offsets and the stack-height compensation (parameterized by chip_height) until stable pickup and placement were achieved without dropping pieces or destabilizing stacks.

References

[1] "PyNiryo 1.2.3 Documentation." [Online]. Available: https://niryorobotics.github.io/pyniryo/v1.2.3/index.html

[2] "PyNiryo 1.2.3 Examples: Movement." [Online]. Available: https://niryorobotics.github.io/pyniryo/v1.2.3/examples/examples_movement.html

[3] "Scalar (Mathematics)." [Online]. Available: https://en.wikipedia.org/w/index.php?title=Scalar_(mathematics)