Hardware Acceleration for Model Predictive Path Integral Control for UAV Applications

Alexander-Odysseus Farmakis

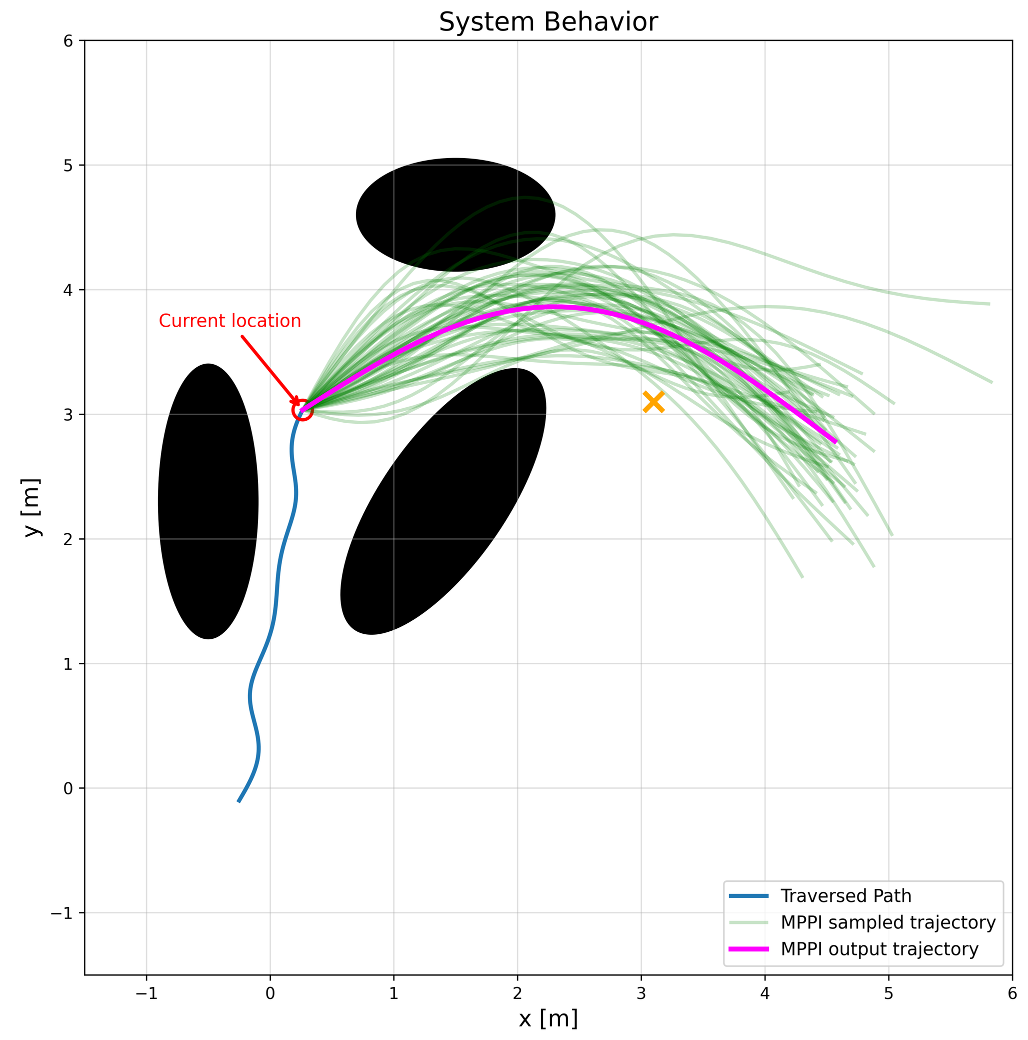
INRIA, IRISA, Université de Rennes | alex.farmakis@inria.fr

1) INTRODUCTION

- The Model Predictive Path Integral (MPPI) control algorithm **[1]** is a state-of-the-art control scheme for finding an optimal path for autonomous vehicles.
- Instead of solving difficult non-convex optimization problems that traditional MPC does, the idea is to sample all possible paths the autonomous vehicle could follow in order to find a solution.
- These paths are all independent of each other, so their computation is inherently parallelizable. The community thus far has focused on GPU-based implementations.
- In smaller vehicles where every bit of power or latency counts, bespoke computational units are a must.

2) METHODOLOGY

- Studying the algorithm, the computation of the rollouts and the trajectory costs are mapped to a systolic array.
- The white noise tensor for the exploration of the is e generated locally, as having these values be communicated would unnecessarily stall the accelerator.
- The calculation of the optimal control over the prediction horizon H maps to a SIMD-style computation
- Due to MPPI being a stochastic algorithm, much of this computation can be done approximately **[2]**.
- FIFOs synchronize data from systolic to SIMD datapath to form coarse-grain 3-stage pipeline.



Algorithm 1 General MPPI Algorithm

Inputs: current state $\boldsymbol{x}_c$, reference trajectory $\boldsymbol{R}$, current control sequence $\boldsymbol{v}$, number of samples N, horizon H, actuators A, noise variance σ
Outputs: optimal control v^*

```
Sample white noise tensor w ~ N(0, σ) where w ∈ R^{N×H×A}
u_{i,t,a} ← v_{t,a} + w_{i,t,a}                    ▷ Centering the Gaussian
for i in 0 to N − 1 do in parallel
    x_{i,0} ← x_c
    c_i ← 0
    for t in 0 to H − 1 do                         ▷ Rollout calculations
        x_{i,t+1} ← F(x_{i,t}, u_{i,t,a})
        c_i ← c_i + ℓ(x_{i,t}, u_{i,t,a}, R_t)
    end for
    J_i ← c_{H−1} + ϕ(x_{i,H}, R_H)                ▷ Final cost calculations
end for
J_min ← min{J_i}
J_max ← max{J_i}
u′ ← 0
d ← 0
for i in 0 to N − 1 do                             ▷ Optimal control
    ω_i ← exp(−h (J_max − J_i)/(J_max − J_min))
    u′_{t,a} ← u′_{t,a} + ω_i · u_{i,t,a}
    d ← d + ω_i
end for
v*_{t,a} ← u′_{t,a} / d
return v*
```

Sample generation
Rollout computation
Final cost calculation
Min-max cost search
Weighting function
Normalization and update

3) CURRENT STATUS

- Design stage of the accelerator has been completed, final functional verification steps remain.
- For verification, other than standard flight paths for the UAV (e.g. hovering, strafing etc.), the QoR is to be compared to the software-based golden model.
- After the architectural verification, we plan on porting this accelerator to an FPGA to test with real sensor data.

4) DISCUSSION

- Experimentation on aggressive quantization (e.g. FP8, INT16 etc.) or even mixed-precision formats could potentially boost the efficiency of such a design.
- Efficient methods for random number generation targeting Gaussian distributions should be explored.
- Addition of safety-critical features, such as Barrier Rate functions **[3]** could be very interesting.

5) REFERENCES

- **[1]** G. Williams, A. Aldrich, and E. A. Theodorou, "Model Predictive Path Integral Control: From Theory to Parallel Computation," vol. 40, no. 2, pp. 344–357, Feb. 2017, doi: 10.2514/1.g001921.
- **[2]** V. Leon et al., "Approximate Computing Survey, Part I: Terminology and Software & Hardware Approximation Techniques," ACM Computing Surveys, vol. 57, no. 7, pp. 1–36, Mar. 2025, doi: 10.1145/3716845.
- **[3]** H. Parwana et al., "BR-MPPI: Barrier Rate Guided MPPI for Enforcing Multiple Inequality Constraints with Learned Signed Distance Field," June 2025, doi: https://doi.org/10.48550/arXiv.2506.07325.