Vision-Based MPC for Robotic Manipulation - Project Plan

Alexander Wegener | ME/SE 740 | Spring 2026

Project Timeline Overview

Total Duration: 12 weeks (February 10 - April 30, 2026)

- **Phase 1**: Foundation & Setup (Weeks 1-3)
- **Phase 2**: Baseline Implementation (Weeks 4-5)
- **Phase 3**: Vision Integration (Weeks 6-8)
- **Phase 4**: Uncertainty-Aware MPC (Weeks 9-10)
- **Phase 5**: Experiments & Analysis (Weeks 11-12)
- **Phase 6**: Final Report & Presentation (Week 12-13)

Phase 1: Foundation & Setup (Weeks 1-3)

Timeline: Feb 10 - March 2 | **Milestone**: Interim Progress Presentation (March 17)

Week 1: Literature Review & Environment Setup (8-10 hours)

Tasks:

1. **Literature Deep Dive** (4 hours)
   - Review visual servoing fundamentals [Chaumette & Hutchinson, 2006]
   - Study stochastic MPC formulations [Mesbah, 2016]
   - Examine planar pushing dynamics [Hogan & Rodriguez, 2018]
   - Review MPC for contact-rich manipulation (recent papers 2020-2025)
   - Document key equations and algorithmic approaches
2. **Simulation Environment Setup** (4-6 hours)
   - Install/verify MuJoCo and Python bindings (dm_control or mujoco-py)
   - Set up Franka Panda URDF/XML model in MuJoCo
   - Configure basic scene: robot, table, pushable object
   - Test basic robot control (joint position/velocity commands)
   - Verify camera rendering and image retrieval

Deliverables:

- Literature review summary document (2-3 pages)
- Working MuJoCo environment with robot visualization
- Test script demonstrating basic robot motion

Week 2: Dynamics Modeling & Contact Simulation (10-12 hours)

Tasks:

1. **Planar Pushing Dynamics** (5-6 hours)
   - Implement/verify quasi-static pushing model
   - Define state: object pose (x, y, θ), end-effector pose
   - Identify friction parameters in MuJoCo
   - Test contact behavior: sliding, sticking, rotation
   - Validate physics against simple test cases
2. **MPC Problem Formulation** (5-6 hours)
   - Define state space: $x = [x_{obj}, y_{obj}, \theta_{obj}, x_{ee}, y_{ee}]$
   - Define control inputs: $u = [v_x, v_y]$ (end-effector velocities)
   - Formulate cost function:
     - State error: $\|x - x_{goal}\|^2_Q$
     - Control effort: $\|u\|^2_R$
     - Terminal cost: $\|x_N - x_{goal}\|^2_{Q_N}$
   - Define constraints:
     - Workspace limits
     - Velocity limits
     - Contact force limits (if applicable)
   - Document mathematical formulation

Deliverables:

- Validated contact dynamics model
- Complete MPC problem formulation document
- Test scenarios for pushing primitives (translate, rotate)

Week 3: MPC Framework Implementation (10-12 hours)

Tasks:

1. **MPC Solver Setup** (4-5 hours)
    - Choose optimization backend (CasADi, CVXPY, or acados)
    - Implement discrete-time system dynamics
    - Set up quadratic program (QP) structure
    - Configure solver parameters (horizon N=10-20, dt=0.05-0.1s)
2. **Baseline Controller (Perfect State)** (4-5 hours)
    - Implement MPC with full state feedback
    - Create simple test task: push object to target pose
    - Tune cost function weights (Q, R, Q_N)
    - Verify constraint satisfaction
    - Test trajectory smoothness and convergence
3. **Prepare for Interim Presentation** (2-3 hours)
    - Document progress on project definition
    - Prepare slides on background, scope, and progress
    - Create visualizations of simulation environment
    - Outline remaining work and timeline

Deliverables:

- Working MPC controller with perfect state
- Successful completion of simple pushing task
- Interim presentation slides (due March 17)

Phase 2: Baseline Implementation (Weeks 4-5)

Timeline: March 3 - March 16

Week 4: Baseline MPC Refinement (8-10 hours)

Tasks:

1. **Task Complexity Expansion** (3-4 hours)

- Test multiple target poses (translation + rotation)
- Evaluate different initial configurations
- Test corner cases (large rotations, edge-of-workspace)

2. **Performance Metrics Implementation** (3-4 hours)
   - Define metrics:
     - Tracking error: $\|x(t) - x_{goal}\|$
     - Task success rate (within tolerance ε)
     - Completion time
     - Constraint violations (count, magnitude)
     - Control effort: $\int\|u(t)\|^2 dt$
   - Implement automated logging system
   - Create baseline performance dataset (20-30 trials)
3. **Visualization Tools** (2 hours)
   - Implement trajectory plotting
   - Create video recording of trials
   - Set up real-time state/cost visualization

Deliverables:

- Comprehensive baseline performance dataset
- Visualization and logging infrastructure
- Reference trajectories for comparison

Week 5: Camera Integration & Pose Estimation (10-12 hours)

Tasks:

1. **Camera Configuration** (3-4 hours)
   - Set up eye-in-hand camera in MuJoCo
   - Configure intrinsic parameters (focal length, resolution)
   - Verify camera-robot calibration
   - Test image rendering at control frequency
2. **Fiducial-Based Pose Estimation** (4-5 hours)

- Add AprilTag/ArUco marker to object in simulation
- Implement marker detection (OpenCV)
- Compute object pose from marker (PnP algorithm)
- Validate against ground truth
- Characterize measurement noise

3. **Alternative: Learning-Based Estimation** (optional, 3-4 hours)
   - If time permits, explore simple CNN-based pose estimation
   - Compare accuracy/latency with fiducial approach
   - Document trade-offs

Deliverables:

- Working vision pipeline with pose estimation
- Noise characterization report
- Vision-based state estimation at 10-30 Hz

Phase 3: Vision Integration (Weeks 6-8)

Timeline: March 17 - April 6 | **Checkpoint**: Interim Presentation

Week 6: Certainty-Equivalent MPC (10-12 hours)

Tasks:

1. **Vision-in-the-Loop Control** (5-6 hours)
   - Replace perfect state with vision-based estimates
   - Implement certainty-equivalent MPC (use mean estimate)
   - Handle measurement latency (buffer recent states)
   - Test with nominal noise levels
2. **Noise Model Tuning** (3-4 hours)
   - Add Gaussian noise to pose estimates: $\hat{x} = x + \varepsilon, \varepsilon \sim N(0, \Sigma)$
   - Tune Σ to match realistic vision systems
   - Implement measurement dropout (frame loss)
   - Add latency (1-3 frames delay)
3. **Performance Comparison** (2-3 hours)

- Run same tasks as baseline
- Compare tracking error and success rate
- Identify failure modes
- Document performance degradation

Deliverables:

- Certainty-equivalent MPC with vision
- Noise model parameter set
- Performance comparison report (baseline vs. CE-MPC)

Week 7: Occlusion & Perception Failure Handling (10-12 hours)

Tasks:

1. **Occlusion Scenarios** (4-5 hours)
    - Simulate partial/full object occlusion
    - Implement state prediction during occlusion (open-loop)
    - Test recovery when vision returns
    - Characterize maximum tolerable occlusion duration
2. **State Estimation Enhancement** (4-5 hours)
    - Implement simple Kalman filter for state smoothing
    - Use motion model to predict object state
    - Fuse visual measurements with predictions
    - Test filtering effectiveness
3. **Robustness Testing** (2-3 hours)
    - Vary noise levels (low, medium, high)
    - Test with intermittent measurements
    - Document robustness boundaries

Deliverables:

- Occlusion-aware control strategy
- Kalman filter implementation
- Robustness characterization report

Week 8: Buffer Week & Documentation (6-8 hours)

Tasks:

1. **Catch-Up & Debugging** (3-4 hours)
   - Address any issues from Weeks 6-7
   - Refactor code for clarity
   - Add unit tests for critical components
2. **Mid-Project Documentation** (3-4 hours)
   - Update technical documentation
   - Document all parameters and design decisions
   - Prepare interim results summary
   - Revise project timeline if needed

Deliverables:

- Clean, documented codebase
- Mid-project technical report

Phase 4: Uncertainty-Aware MPC (Weeks 9-10)

Timeline: April 7 - April 20

Week 9: Stochastic MPC Formulation (10-12 hours)

Tasks:

1. **Uncertainty Quantification** (3-4 hours)
   - Propagate state uncertainty through dynamics
   - Compute covariance evolution: $\Sigma(t+1) = A\Sigma(t)A^T + Q$
   - Characterize uncertainty growth over horizon
2. **Chance-Constrained MPC** (5-6 hours)

- Formulate probabilistic constraints: $P(g(x) \leq 0) \geq 1-\delta$
- Implement using scenario-based or ellipsoidal approximations
- Alternatively: tube-based MPC with RPI sets
- Tune risk parameter δ (e.g., $\delta = 0.05, 0.1$)

3. **Solver Integration** (2-3 hours)
   - Adapt MPC solver for stochastic formulation
   - Verify constraint satisfaction probability
   - Test computational performance

Deliverables:

- Uncertainty-aware MPC implementation
- Stochastic constraint verification
- Computational benchmarks

Week 10: Validation & Tuning (10-12 hours)

Tasks:

1. **Parameter Tuning** (4-5 hours)
   - Tune Q, R for uncertainty-aware MPC
   - Adjust risk parameter δ
   - Balance performance vs. conservativeness
   - Document tuning methodology
2. **Comparative Testing** (4-5 hours)
   - Run all three controllers on same scenarios:
     - Baseline MPC (perfect state)
     - CE-MPC (vision, no uncertainty handling)
     - UA-MPC (uncertainty-aware)
   - Test across noise levels and scenarios
   - Record all performance metrics
3. **Preliminary Analysis** (2-3 hours)

- Analyze when UA-MPC outperforms CE-MPC
- Identify scenarios where uncertainty handling matters most
- Document qualitative observations

Deliverables:

- Tuned uncertainty-aware MPC
- Complete performance dataset (all controllers)
- Preliminary results summary

Phase 5: Experiments & Analysis (Weeks 11-12)

Timeline: April 21 - April 27

Week 11: Comprehensive Experiments (12-15 hours)

Tasks:

1. **Experimental Design** (2-3 hours)
    - Define test scenarios:
        - Nominal case (low noise)
        - High noise case
        - Occlusion case
        - Latency case
        - Combined disturbances
    - Define number of trials per scenario (N=20-50)
    - Set up automated experiment runner
2. **Data Collection** (6-8 hours)
    - Run full experimental suite
    - Ensure sufficient statistical power
    - Monitor for anomalies/bugs
    - Back up all data
3. **Initial Data Analysis** (4-5 hours)

- Compute statistical summaries (mean, std, median)
- Generate performance plots:
  - Tracking error vs. time
  - Success rate vs. noise level
  - Completion time distributions
  - Constraint violation rates
- Perform statistical tests (t-tests, ANOVA)

Deliverables:

- Complete experimental dataset
- Statistical analysis report
- Performance comparison plots

Week 12: Results Interpretation & Refinement (10-12 hours)

Tasks:

1. **Deep Analysis** (5-6 hours)
   - Investigate failure cases
   - Analyze uncertainty propagation
   - Examine trade-offs (performance vs. robustness)
   - Identify practical insights for vision-based MPC
2. **Visualization & Figures** (3-4 hours)
   - Create publication-quality figures
   - Generate trajectory comparison videos
   - Make uncertainty ellipse visualizations
   - Create summary table of results
3. **Key Findings Synthesis** (2-3 hours)
   - Identify main contributions
   - Document when uncertainty-aware MPC is beneficial

- Document when uncertainty-aware MPC is beneficial
- Outline limitations and future work

Deliverables:

- Complete results analysis
- All figures and videos for report/presentation
- Key findings summary

Phase 6: Final Deliverables (Week 13)

Timeline: April 28 - April 30

Final Report Writing (15-20 hours total over Weeks 11-13)

Structure:

1. **Abstract** (1 hour) - 200 words, main findings
2. **Introduction** (2-3 hours) - Motivation, problem statement, contributions
3. **Background** (2-3 hours) - Visual servoing, MPC, uncertainty quantification
4. **Approach** (3-4 hours) - System model, controller designs, implementation
5. **Experiments** (2-3 hours) - Setup, scenarios, metrics
6. **Results** (3-4 hours) - Data presentation, analysis, discussion
7. **Conclusion** (1-2 hours) - Summary, limitations, future work
8. **References** - Throughout
9. **Appendix** (1 hour) - Additional plots, parameters

Target Length: 8-12 pages

Final Presentation (6-8 hours)

Tasks:

1. **Slide Preparation** (4-5 hours)
    - 12-15 slides, 15-20 minute talk
    - Structure: Motivation → Approach → Results → Conclusions
    - Include key equations, diagrams, and result plots
    - Add demo video clips

- Add demo video clips

2. **Presentation Practice** (2-3 hours)

    - Rehearse timing
    - Prepare for questions
    - Test technical setup

Deliverables:

- Final written report (due April 30)
- Final presentation (late April)
- Code repository with documentation

Risk Management & Contingency Plans

High-Risk Items:

1. **MuJoCo Contact Dynamics Instability**
    - Contingency: Simplify contact model, use quasi-static approximation
    - Extra time: 5-8 hours
2. **Uncertainty-Aware MPC Too Complex**
    - Contingency: Focus on CE-MPC vs. baseline comparison only
    - Extra time: Saves 10-15 hours
3. **Vision Pipeline Issues**
    - Contingency: Use ground truth with added noise instead of full vision
    - Extra time: Saves 5-8 hours
4. **Computational Performance**
    - Contingency: Reduce horizon N, decrease control frequency
    - Extra time: 3-5 hours tuning

Time Buffer:

- Week 8 serves as a dedicated buffer week
- Weekends provide additional capacity if needed
- Some tasks can be parallelized (writing + experiments)

Resource Requirements

Software:

- MuJoCo (free academic license)
- Python 3.8+: numpy, scipy, matplotlib
- CasADi or CVXPY (optimization)
- OpenCV (vision)
- Optional: acados (fast MPC solver)

Hardware:

- Standard laptop (GPU helpful but not required)
- ~20GB disk space for data/videos

References:

- All key papers already identified in proposal
- Additional recent MPC literature as needed

Success Criteria

Minimum Viable Project:

✓ Baseline MPC working with perfect state
✓ CE-MPC working with vision
✓ Performance comparison showing impact of perception uncertainty
✓ Final report documenting findings

Target Goals:

✓ All of above, plus:
✓ Uncertainty-aware MPC implementation
✓ Comprehensive experimental evaluation
✓ Statistical analysis of performance differences
✓ Clear practical insights

Stretch Goals:

✓ Learning-based pose estimation

✓ Learning-based pose estimation
✓ Multiple object manipulation
✓ Online MPC parameter adaptation

✓ Real-world validation (if hardware available)

Weekly Time Commitment Summary

Phase	Week	Hours	Cumulative
1	1	8-10	8-10
1	2	10-12	18-22
1	3	10-12	28-34
2	4	8-10	36-44
2	5	10-12	46-56
3	6	10-12	56-68
3	7	10-12	66-80
3	8	6-8	72-88
4	9	10-12	82-100
4	10	10-12	92-112
5	11	12-15	104-127
5	12	10-12	114-139
6	13	15-20	129-159

Total Estimated Time: 130-160 hours over 13 weeks (~10-12 hours/week)

Next Immediate Steps (This Week)

1. **Today/Tomorrow**: Set up MuJoCo environment, test Franka model
2. **This Week**: Complete literature review, formulate MPC problem
3. **By March 2**: Have working baseline MPC with perfect state

4. **By March 17**: Deliver interim presentation

This plan is a living document. Update as project evolves.