Co-Design of Anytime Computation and Robust Control

Yash Vardhan Pant, Kartik Mohta, Houssam Abbas, Truong X. Nghiem, Joseph Devietti, Rahul Mangharam

APPENDIX

In this appendix we give the detailed mathematical derivation of the results of Section III. The controller is designed using a Robust Model Predictive Control (RMPC) approach via constraint restriction [1], [2], and augments it by an adaptation to the error-delay curve of the estimator. In order to ensure robust safety and feasibility, the key idea of the RMPC approach is to tighten the constraint sets iteratively to account for possible effect of the disturbances. As time progresses, this “robustness margin” is used in the MPC optimization with the nominal dynamics, i.e., the original dynamics where the disturbances are either removed or replaced by nominal disturbances. Because only the nominal dynamics are used, the complexity of the optimization is the same as for the nominal problem.

Since the controller only has access to the estimated state \( \hat{x} \), we need to rewrite the plant’s dynamics with respect to \( x \). The error between \( x_k \) and \( \hat{x}_k \) is \( e_k = x_k - \hat{x}_k \). At time step \( k + 1 \) we have

\[
\hat{x}_{k+1} = x_{k+1} - e_{k+1} = A x_k + B_1(\delta_k) u_{k-1} + B_2(\delta_k) w_k + e_{k+1},
\]

then, by writing \( x_k = \hat{x}_k + e_k \), we obtain the dynamics

\[
\hat{x}_{k+1} = A \hat{x}_k + B_1(\delta_k) u_{k-1} + B_2(\delta_k) u_k + \hat{w}_k \tag{1}
\]

where \( \hat{w}_k = w_k + A e_k - e_{k+1} \). The set of possible values \( \hat{w}_k \) depends on the estimation accuracy at steps \( k \) and \( k + 1 \) and is denoted by \( \hat{W}(e[k], e[k+1]) \), i.e.,

\[
\hat{W}(e, e') : = \{ w + A e - e' | w \in W, e \in E(e), e' \in E(e') \}.
\]

Note that \( \hat{W}(e[k], e[k+1]) \) is independent of the time step \( k \). It can be computed as \( \hat{W}(e, e') = W \oplus A E(e) \oplus (-e(e')) \) where the symbol \( \oplus \) denotes the Minkowski sum of two sets.

The dynamics in (1) has a nonstandard form where it depends on both the current and the previous control inputs. However we can expand the state variable to store the previous control input as

\[
\hat{z}_k = \begin{bmatrix} \hat{x}_k \\ u_{k-1} \end{bmatrix} \in \mathbb{R}^{n+m}
\]

and rewrite the dynamics as, for all \( k \geq 0 \),

\[
\hat{z}_{k+1} = \hat{A}(\delta_k) \hat{z}_k + \hat{B}(\delta_k) u_k + \hat{F} \hat{w}_k. \tag{2}
\]

Here, the system matrices are

\[
\hat{A}(\delta_k) = \begin{bmatrix} A & B_1(\delta_k) \\ 0_{m \times n} & 0_{m \times m} \end{bmatrix}, \quad \hat{B}(\delta_k) = \begin{bmatrix} B_2(\delta_k) \\ 0_{m \times m} \end{bmatrix}, \quad \hat{F} = \begin{bmatrix} I_n \\ 0_{m \times n} \end{bmatrix}.
\]

Let the actual expanded state be \( z_k = [x_k^T, u_{k-1}^T]^T \). Because the expanded state consists of both the plant’s state and the previous control input, the state constraint \( x_k \in X \) and the control constraint \( u_k \in U \) are equivalent to the joint constraint \( z_k \in X \times U \). We can now describe the RAMPC algorithm for the dynamics in (2).

A. Tractable RAMPC Algorithm

Let \( N \geq 1 \) be the horizon length of the RMPC optimization. Because the system matrices in the state equation (2) depend nonlinearly on the variables \( \delta_k \), the RMPC optimization is generally a mixed-integer nonlinear program, which is very hard to solve. To simplify the RMPC optimization to make it tractable, we fix the estimation mode for the entire RMPC horizon.

Let \( P_{(\delta, \epsilon)}(\hat{x}_k, \delta_k, \epsilon_k, u_{k-1}) \) denote the RMPC optimization problem at step \( k \geq 0 \) where the current state estimate is \( \hat{x}_k \), the current estimation mode is \( (\delta_k, \epsilon_k) \in \Delta \), the previous control input is \( u_{k-1} \), and the estimation mode for the entire horizon (after step \( k \)) is fixed at \( (\delta, \epsilon) \in \Delta \). Since the system matrices become constant now, if the stage cost \( \ell(\cdot) \) is linear or positive semidefinite quadratic, each optimization problem \( P_{(\delta, \epsilon)}(\cdot) \) is tractable and can be solved efficiently as we will show later. The RAMPC algorithm with Anytime Estimation is stated in Alg. 1.

B. RMPC Formulation

We formulate the RMPC optimization \( P_{(\delta, \epsilon)}(\hat{x}_k, \delta_k, \epsilon_k, u_{k-1}) \) with respect to the nominal dynamics, which is the original dynamics in Eq. (2) but the disturbances are either removed or replaced by nominal disturbances. To ensure robust feasibility and safety, the state constraint set is tightened after each step using a candidate stabilizing state feedback control, and a terminal constraint is derived. In this RMPC formulation, we extend the approach in [1], [2]. At time step \( k \), given \( (\hat{x}_k, \delta_k, \epsilon_k, u_{k-1}) \) and for a fixed \( (\delta, \epsilon) \), we solve the following optimization

\[
J_{\delta, \epsilon}^* (\hat{x}_k, \delta_k, \epsilon_k, u_{k-1}) = \min_{u, x} \sum_{j=0}^{N} \ell(\pi_{k+j|k}, u_{k+j|k}) \tag{4a}
\]

subject to, \( \forall j \in \{0, \ldots, N\} \)

*This work was supported by STARnet a Semiconductor Research Corporation program sponsored by MARCO and DARPA. The Departments of Electrical and Systems Engineering and Computer and Information Sciences, University of Pennsylvania, Philadelphia, U.S.A. {yashpant, kmohta, habbas, nghiem, rahulm}@seas.upenn.edu, devietti@cis.upenn.edu
\[ \begin{align*}
    \overline{z}_{k+j+1|k} &= \hat{A}(\delta_k|j,k)\overline{z}_{k+j|k} + \hat{B}(\delta_k|j,k)u_{k+j|k} \\
    (\delta_{k+j+1|k}, \epsilon_{k+j+1|k}) &= (\delta_k, \epsilon) \\
    \overline{z}_{k+j|k} &= \begin{bmatrix} \overline{I}_n & 0_{n \times m} \end{bmatrix} \overline{z}_{k+j|k} \\
    \overline{z}_{k|k} &= \begin{bmatrix} \hat{x}_k, u_{k-1} \end{bmatrix}^T \\
    \overline{z}_{k+j|k} &\in Z_j(\epsilon_k, \epsilon) \\
    \overline{z}_{k+N+1|k} &\in Z_f(\epsilon_k, \epsilon)
\end{align*} \]

where \( C(\delta, \epsilon) \) is a robust control invariant admissible set for \( \delta \) [3], i.e., there exists a feedback control law \( u = \kappa(z) \) such that \( \forall z \in C(\delta, \epsilon) \) and \( \forall w \in \bar{W}(\epsilon, \epsilon) \)

\[ \begin{align*}
    \hat{A}(\delta)z + \hat{B}(\delta)\kappa(z) + L_N\hat{F}w &\in C(\delta, \epsilon) \\
    z &\in Z(\epsilon, \epsilon)
\end{align*} \]

We remark that \( C(\delta, \epsilon) \) does not depend on \((\delta_k, \epsilon_k)\), therefore it can be computed offline for each mode \((\delta, \epsilon)\).

### C. Proofs of Feasibility

The RMPC formulation of the previous section, with a fixed estimation mode \((\delta, \epsilon) \in \Delta\), is designed to ensure that the control problem is robustly feasible, as stated in the following theorem.

**Theorem 0.1 (Robust Feasibility of RAMPC):** For any estimation mode \((\delta, \epsilon)\), if \( P_{\delta, \epsilon}(\hat{x}_k, \delta_k, \epsilon_k, u_{k-1}) \) is feasible then the system (2) controlled by the RAMPC and subjected to disturbances constrained by \( w_k \in \bar{W} \) robustly satisfies the state constraint \( x_k \in X \) and the control input constraint \( u_k \in U \), and all subsequent optimizations \( P_{\delta, \epsilon}(\hat{x}_{k+1}, \delta[k+1], \epsilon[k+1], u_{k+1}) \), for \( k > k_0 \), are feasible.

**Proof:** We will prove the theorem by recursion. We will show that if at any time step \( k \) the RMPC problem \( P_{\delta, \epsilon}(\hat{x}_k, \delta[k], \epsilon[k], u_{k-1}) \) is feasible and feasible control input \( u_k = u_{k|k} \) is applied with estimation mode \((\delta[k+1], \epsilon[k+1]) = (\delta, \epsilon)\) then \( u_k \) is admissible and at the next time step \( k + 1 \), the actual plant's state \( x_{k+1} \) is inside \( X \) and the optimization \( P_{\delta, \epsilon}(\hat{x}_{k+1}, \delta[k+1], \epsilon[k+1], u_{k}) \) is feasible for all disturbances. Then we can conclude the theorem because, by recursion, feasibility at time step \( k_0 \) implies robust constraint satisfaction and feasibility at time step \( k_0 + 1 \), and so on at all subsequent time steps.

Suppose \( P_{\delta, \epsilon}(\hat{x}_k, \delta[k], \epsilon[k], u_{k-1}) \) is feasible. Then it has a feasible solution \( \{(\hat{x}_{j+k}, \delta[j+k], \epsilon[j+k], u_{j+k-1})\}_{j=0}^{N+1} \) that satisfies all the constraints in (4). Now we will construct a feasible candidate solution for \( P_{\delta, \epsilon}(\hat{x}_{k+1}, \delta[k+1], \epsilon[k+1], u_{k}) \) at the next time step by shifting the above solution by one step. Consider the following candidate solution for \( P_{\delta, \epsilon}(\hat{x}_{k+1}, \delta[k+1], \epsilon[k+1], u_{k}) \):

\[ \begin{align*}
    \overline{z}_{k+j+1|k+1} &= \overline{z}_{k+j+1|k} + L_j\hat{F}w_k \\
    \overline{z}_{k+N+1|k+1} &= \hat{A}(\delta)\overline{z}_{k+N+1|k} + \hat{B}(\delta)u_{k+N+1|k} + L_j\hat{F}w_k \\
    u_{k+i+1|k+1} &= u_{k+i+1|k} + K_i(\delta)\hat{F}w_k \\
    u_{k+N+1|k+1} &= \kappa(\overline{z}_{k+N+1|k+1})
\end{align*} \]

in which \( j \in \{0, \ldots, N\} \), \( i \in \{0, \ldots, N - 1\} \), and \( \kappa(\cdot) \) is the feedback control law for the invariant set \( C(\delta, \epsilon) \) that is used in the terminal set. We first show that the input and state constraints are satisfied for \( u_k \) and \( x_{k+1} \), then we will prove the feasibility of the above candidate solution for \( P_{\delta, \epsilon}(\hat{x}_{k+1}, \delta[k+1], \epsilon[k+1], u_{k}) \).
Validity of the applied input and the next state: The next plant’s state is
\[ x_{k+1} = Ax_k + B_1(\delta[k])u_{k-1} + B_2(\delta[k])u_k + w_k \]
\[ = A(\hat{x}_k + e_k) + B_1(\delta[k])u_{k-1} + B_2(\delta[k])u_k^* + u_k + w_k \]
\[ \text{in which } e_{k+1} \in \mathcal{E}(\epsilon) \text{ and } (u_k + Ae_k - e_{k+1}) \in \hat{\mathcal{W}}(\epsilon[k], \epsilon). \]

From the construction of the candidate solution we have dynamics constraint in Eq. (4b). For \( \delta[k] \in \mathcal{Z}_f \), \( \delta[k] \in \mathcal{C} \), \( \epsilon[k] \in \mathcal{W} \), and \( w_k \in \mathcal{W} \), we have \( \mathcal{Z}_{k+1} \in \mathcal{Z} \). The definition of \( C(\delta, \epsilon) \) in (10) implies \( C(\delta, \epsilon) \subseteq \mathcal{Z} \). Therefore \( \mathcal{Z}_{k+1} \in \mathcal{Z} \), and note that \( u_{k+1} = \kappa(\mathcal{Z}_{k+1}) \), we have
\[ \mathcal{Z}_{k+1} + 1 + L_N \hat{w} = \hat{A}(\delta) \mathcal{Z}_{k+1} + 1 + L_N \hat{w} \]

It follows from \( \mathcal{Z}_{k+1} \in \mathcal{Z} \) and from the definition of the invariant control invariant admissible set \( C(\delta, \epsilon) \) (Eq.(10)) that \( \mathcal{Z}_{k+1} + 1 + L_N \hat{w} \in \mathcal{Z}(\delta, \epsilon) \) for all \( w \in \mathcal{W} \). Then by definition of the Pontryagin difference, we conclude that \( \mathcal{Z}_{k+1} \in \mathcal{Z}(\delta, \epsilon) \), to both sides of (12b) and note that \( u_{k+1} = \kappa(\mathcal{Z}_{k+1}) \),

We need to show that \( \mathcal{Z}_{k+1} \in \mathcal{Z} \), and from the definition of the Pontryagin difference, we conclude that \( \mathcal{Z}_{k+1} \in \mathcal{Z} \).

The control algorithm in Alg. 1, in each time step \( k \), solves \( \mathcal{P}(\hat{x}_k, \delta_k, \epsilon_k, u_{k-1}) \) for each estimation mode \( (\delta, \epsilon) \in \Delta \) and selects the control input \( u_k \) and the next estimation mode \( (\delta_{k+1}, \epsilon_{k+1}) \) corresponding to the best total cost \( J(\delta, \epsilon, u_{k-1}) \).

Therefore, during the course of control, the algorithm may switch between the estimation modes in \( \Delta \) depending on the system’s state. Thm. 0.2 states that if the control algorithm Alg. 1 is feasible in its first time step then it will be robustly feasible and the state and control input constraints are also robustly satisfied.

**Proof:** The Theorem can be proved by recursively applying Thm. 0.1. Indeed, suppose at time step \( k \) the algorithm is feasible and results in control input \( u_k \) and next estimation mode \( (\delta_{k+1}, \epsilon_{k+1}) \), then \( \mathcal{P}(\hat{x}_{k+1}, \delta_{k+1}, \epsilon_{k+1}, u_{k+1}) \) is feasible. By Thm. 0.1, \( u_k \in U \) and at the next time step \( k+1 \), \( \hat{x}_{k+1} = x_k \in X \) and \( \mathcal{P}(\hat{x}_{k+1}, \delta_{k+1}, \epsilon_{k+1}, u_{k+1}) \) is also feasible, hence the algorithm is feasible. Therefore, the Theorem holds by induction.

### REFERENCES

