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 [17], [18], 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 $\hat{x}$. The error between x_k and $\hat{x}_k$ is $e_k = x_k - \hat{x}_k$. At time step $k + 1$ we have

$$\hat{x}_{k+1} = x_{k+1} - e_{k+1}$$
$$= Ax_k + B_1(\delta_k)u_{k-1} + B_2(\delta[k])u_k + w_k - e_{k+1},$$

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

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

where $\hat{w}_k = w_k + Ae_k - e_{k+1}$. The set of possible values of $\hat{w}_k$ depends on the estimation accuracy at steps k and $k + 1$ and is denoted by $\widehat{\mathcal{W}}(\epsilon[k], \epsilon[k + 1])$, i.e., $\widehat{\mathcal{W}}(\epsilon, \epsilon') := \{w + Ae - e' \mid w \in \mathcal{W}, e \in \mathcal{E}(\epsilon), e' \in \mathcal{E}(\epsilon')\}$. Note that $\widehat{\mathcal{W}}(\epsilon[k], \epsilon[k + 1])$ is independent of the time step k. It can be computed as $\widehat{\mathcal{W}}(\epsilon, \epsilon') = \mathcal{W} \oplus A\mathcal{E}(\epsilon) \oplus (-\mathcal{E}(\epsilon'))$ 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 $k \geq 0$,

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

Here, the system matrices are

$$\hat{A}(\delta_k) = \begin{bmatrix} A & B_1(\delta_k) \\ \mathbf{0}_{m \times n} & \mathbf{0}_{m \times m} \end{bmatrix},$$
$$\hat{B}(\delta_k) = \begin{bmatrix} B_2(\delta_k) \\ \mathbb{I}_m \end{bmatrix}, \quad \hat{F} = \begin{bmatrix} \mathbb{I}_n \\ \mathbf{0}_{m \times n} \end{bmatrix}. \quad (3)$$

Let the actual expanded state be $z_k = \begin{bmatrix} x_k^T, u_{k-1}^T \end{bmatrix}^T$. 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 δ_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 $\mathbb{P}_{(\delta,\epsilon)}(\hat{x}_k, \delta_k, \epsilon_k, u_{k-1})$ 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 $\mathbb{P}_{(\delta,\epsilon)}(\hat{x}., \delta., \epsilon., u._{-1})$ 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 $\mathbb{P}_{(\delta,\epsilon)}(\hat{x}_k, \delta_k, \epsilon_k, u_{k-1})$ 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 [17], [18]. At time step k, given $(\hat{x}_k, \delta_k, \epsilon_k, u_{k-1})$ and for a fixed (δ, ϵ), we solve the following optimization

$$J_{\delta,\epsilon}^*(\hat{x}_k, \delta_k, \epsilon_k, u_{k-1}) = \min_{\boldsymbol{u}, \boldsymbol{x}} \sum_{j=0}^{N} \ell(\overline{x}_{k+j|k}, u_{k+j|k}) \quad (4a)$$

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

$$\overline{z}_{k+j+1|k} = \hat{A}(\delta_{k+j|k})\overline{z}_{k+j|k} + \hat{B}(\delta_{k+j|k})u_{k+j|k} \quad (4b)$$

$$(\delta_{k+j+1|k}, \epsilon_{k+j+1|k}) = (\delta, \epsilon)$$

$$(\delta_{k|k}, \epsilon_{k|k}) = (\delta_k, \epsilon_k) \quad (4c)$$

$$\overline{x}_{k+j|k} = \begin{bmatrix} \mathbb{I}_n & \mathbf{0}_{n \times m} \end{bmatrix} \overline{z}_{k+j|k} \quad (4d)$$

$$\overline{z}_{k|k} = \begin{bmatrix} \hat{x}_k^T, u_{k-1}^T \end{bmatrix}^T \quad (4e)$$

$$\overline{z}_{k+j|k} \in \mathcal{Z}_j(\epsilon_k, \epsilon) \quad (4f)$$

$$\overline{z}_{k+N+1|k} \in \mathcal{Z}_f(\epsilon_k, \epsilon) \quad (4g)$$

in which $\overline{z}$ and $\overline{x}$ are the variables of the nominal dynamics. The constraints of the optimization are explained below.

- (4b) is the nominal dynamics.
- (4c) states that the estimation mode is fixed at (δ, ϵ) except for the first time step when it is (δ_k, ϵ_k).
- (4d) extracts the nominal state $\overline{x}$ of the plant from the nominal expanded state $\overline{z}$.
- (4e) initializes the nominal expanded state at time step k by stacking the current state estimate and the previous control input.
- (4f) tightens the admissible set of the nominal expanded states by a sequence of shrinking sets.
- (4g) constrains the terminal expanded state to the terminal constraint set $\mathcal{Z}_f$.

The state constraint $\mathcal{Z}_j$: The tightened state constraint sets $\mathcal{Z}_j(\epsilon_k, \epsilon)$ are parameterized with two parameters ϵ_k and ϵ. They are defined as follows, for all $j \in \{0, \ldots, N\}$

$$\mathcal{Z}_0(\epsilon_k, \epsilon) = \mathcal{Z} \ominus \hat{F}\mathcal{E}(\epsilon_k) \quad (5)$$
$$\mathcal{Z}_{j+1}(\epsilon_k, \epsilon) = \mathcal{Z}_j(\epsilon, \epsilon) \ominus L_j \hat{F}\widehat{\mathcal{W}}(\epsilon_k, \epsilon) \quad (6)$$

in which the symbol $\ominus$ denotes the Pontryagin difference between two sets. The set $\mathcal{Z}$ combines the constraints for both the plant's state and the control input: $\mathcal{Z} = X \times U$. The matrix L_j is the state transition matrix for the nominal dynamics in (4b) under a candidate state feedback gain $K_j(\delta)$, for $j \in \{0, \ldots, N\}$

$$L_0 = \mathbb{I} \quad (7)$$
$$L_{j+1} = (\hat{A}(\delta) + \hat{B}(\delta)K_j(\delta))L_j \quad (8)$$

Note that the possibly time-varying sequence $K_j(\delta)$ is designed for each choice of δ (i.e., the system matrices $\hat{A}(\delta)$ and $\hat{B}(\delta)$), hence L_j depends on δ; however we write L_j for brevity. The candidate control $K_j(\delta)$ is designed to stabilize the nominal system (4b), desirably as fast as possible so that the sets $\mathcal{Z}_j$ are shrunk as little as possible. In particular, if $K_j(\delta)$ renders the nominal system nilpotent after $M < N$ steps then $L_j = \mathbf{0}$ for all $j \geq M$, therefore $\mathcal{Z}_j(\epsilon_k, \epsilon) = \mathcal{Z}_M(\epsilon_k, \epsilon)$ for all $j > M$.

The terminal constraint $\mathcal{Z}_f$: $\mathcal{Z}_f$ is given by

$$\mathcal{Z}_f(\epsilon_k, \epsilon) = \mathcal{C}(\delta, \epsilon) \ominus L_N \hat{F}\widehat{\mathcal{W}}(\epsilon_k, \epsilon) \quad (9)$$

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

$$\hat{A}(\delta)z + \hat{B}(\delta)\kappa(z) + L_N \hat{F}w \in \mathcal{C}(\delta, \epsilon) \quad (10)$$
$$z \in \mathcal{Z}_N(\epsilon, \epsilon) \quad (11)$$

We remark that $\mathcal{C}(\delta, \epsilon)$ does not depend on (δ_k, ϵ_k), therefore it can be computed offline for each mode (δ, ϵ).

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 (δ, ϵ), if $\mathbb{P}_{(\delta, \epsilon)}(\hat{x}_k, \delta_k, \epsilon_k, u_{k-1})$ is feasible then the system (2)controlled by the RAMPC and subjected to disturbances constrained by $w_k \in \mathcal{W}$ robustly satisfies the state constraint $x_k \in X$ and the control input constraint $u_k \in U$, and all subsequent optimizations $\mathbb{P}_{\delta, \epsilon}(\hat{x}_k, \delta[k], \epsilon[k], u_{k-1}), \forall k > k_0$, are feasible.

Proof: We will prove the theorem by recursion. We will show that if at any time step k the RMPC problem $\mathbb{P}_{\delta, \epsilon}(\hat{x}_k, \delta[k], \epsilon[k], u_{k-1})$ is feasible and feasible control input $u_k = u_{k|k}^\star$ 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 $\mathbb{P}_{\delta, \epsilon}(\hat{x}_{k+1}, \delta[k+1], \epsilon[k+1], u_k)$ 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 $\mathbb{P}_{\delta, \epsilon}(\hat{x}_k, \delta[k], \epsilon[k], u_{k-1})$ is feasible. Then it has a feasible solution $\left(\{\overline{z}_{k+j|k}^\star\}_{j=0}^{N+1}, \{u_{k+j|k}^\star\}_{j=0}^N \right)$ that satisfies all the constraints in (4). Now we will construct a feasible candidate solution for $\mathbb{P}_{\delta, \epsilon}(\hat{x}_{k+1}, \delta[k+1], \epsilon[k+1], u_k)$ at the next time step by shifting the above solution by one step. Consider the following candidate solution for $\mathbb{P}_{\delta, \epsilon}(\hat{x}_{k+1}, \delta[k+1], \epsilon[k+1], u_k)$:

$$\overline{z}_{k+j+1|k+1} = \overline{z}_{k+j+1|k}^\star + L_j \hat{F}\hat{w}_k \quad (12a)$$
$$\overline{z}_{k+N+2|k+1} = \hat{A}(\delta)\overline{z}_{k+N+1|k+1} + \hat{B}(\delta)u_{k+N+1|k+1} \quad (12b)$$
$$u_{k+i+1|k+1} = u_{k+i+1|k}^\star + K_i(\delta)L_i\hat{F}\hat{w}_k \quad (12c)$$
$$u_{k+N+1|k+1} = \kappa(\overline{z}_{k+N+1|k+1}) \quad (12d)$$

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 $\mathcal{C}(\delta, \epsilon)$ 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 $\mathbb{P}_{\delta, \epsilon}(\hat{x}_{k+1}, \delta[k+1], \epsilon[k+1], u_k)$.

Validity of the applied input and the next state: The next plant's state is

$$x_{k+1} = Ax_k + B_1\left(\delta[k]\right)u_{k-1} + B_2\left(\delta[k]\right)u_k + w_k$$
$$= A\left(\hat{x}_k + e_k\right) + B_1\left(\delta[k]\right)u_{k-1} + B_2\left(\delta[k]\right)u_{k|k}^\star + w_k$$
$$= \begin{bmatrix} A & B_1\left(\delta[k]\right) \end{bmatrix} \begin{bmatrix} \hat{x}_k \\ u_{k-1} \end{bmatrix} + B_2\left(\delta[k]\right)u_{k|k}^\star$$
$$+ e_{k+1} + \left(w_k + Ae_k - e_{k+1}\right)$$

in which $e_{k+1} \in \mathcal{E}\left(\epsilon\right)$ and $\left(w_k + Ae_k - e_{k+1}\right) \in \widehat{\mathcal{W}}\left(\epsilon[k], \epsilon\right)$. Note that $\overline{z}_{k|k}^\star = \left[\hat{x}_k^T, u_{k-1}^T\right]^T$. Hence we have

$$\begin{bmatrix} x_{k+1} \\ u_k \end{bmatrix} = \hat{A}(\delta[k])\overline{z}_{k|k}^\star + \hat{B}(\delta[k])u_{k|k}^\star$$
$$+ \hat{F}e_{k+1} + \hat{F}\left(w_k + Ae_k - e_{k+1}\right)$$
$$= \overline{z}_{k+1|k}^\star + \hat{F}e_{k+1} + \hat{F}\left(w_k + Ae_k - e_{k+1}\right)$$

where we use the dynamics in (4b). From (4f) and (6), $\overline{z}_{k+1|k}^\star$ satisfies $\overline{z}_{k+1|k}^\star \in \mathcal{Z}_1\left(\epsilon[k], \epsilon\right) = \mathcal{Z} \ominus \hat{F}\mathcal{E}\left(\epsilon\right) \ominus \hat{F}\widehat{\mathcal{W}}\left(\epsilon[k], \epsilon\right)$. It follows that $\left[x_{k+1}^T, u_k^T\right]^T \in \mathcal{Z} = X \times U$, therefore $x_{k+1} \in X$ and $u_k \in U$.

Initial condition: We have from (2) that $\hat{z}_{k+1} = \hat{A}(\delta[k])\hat{z}_k + \hat{B}(\delta[k])u_k + \hat{F}\hat{w}_k$. On the other hand, by (12a),

$$\overline{z}_{k+1|k+1} = \overline{z}_{k+1|k}^\star + L_0\hat{F}\hat{w}_k$$
$$= \hat{A}(\delta[k])\overline{z}_{k|k}^\star + \hat{B}(\delta[k])u_{k|k}^\star + L_0\hat{F}\hat{w}_k.$$

Note that $\overline{z}_{k|k}^\star = \hat{z}_k$, $u_k = u_{k|k}^\star$, and $L_0 = \mathbb{I}$. Therefore $\overline{z}_{k+1|k+1} = \hat{z}_{k+1}$, hence the initial condition is satisfied.

Dynamics: We show that the candidate solution satisfies the dynamics constraint in Eq. (4b). For $0 \le j < N$ we have

$$\overline{z}_{k+j+2|k+1}$$
$$= \overline{z}_{k+j+2|k}^\star + L_{j+1}\hat{F}\hat{w}_k$$
$$= \hat{A}\left(\delta\right)\overline{z}_{k+j+1|k}^\star + \hat{B}(\delta)u_{k+j+1|k}^\star + L_{j+1}\hat{F}\hat{w}_k$$
$$= \hat{A}\left(\delta\right)\left(\overline{z}_{k+j+1|k+1} - L_j\hat{F}\hat{w}_k\right)$$
$$+ \hat{B}(\delta)\left(u_{k+j+1|k+1} - K_j\left(\delta\right)L_j\hat{F}\hat{w}_k\right) + L_{j+1}\hat{F}\hat{w}_k$$
$$= \hat{A}\left(\delta\right)\overline{z}_{k+j+1|k+1} + \hat{B}(\delta)u_{k+j+1|k+1}$$
$$- \left(\hat{A}\left(\delta\right) + \hat{B}(\delta)K_j\left(\delta\right)\right)L_j\hat{F}\hat{w}_k + L_{j+1}\hat{F}\hat{w}_k$$
$$= \hat{A}\left(\delta\right)\overline{z}_{k+j+1|k+1} + \hat{B}(\delta)u_{k+j+1|k+1}$$

where the equality in (8) is used to derive the last equality. Therefore the dynamics constraint is satisfied for all $0 \le j < N$. For $j = N$, the constraint is satisfied by construction by (12b).

State constraints: We need to show that $\overline{z}_{(k+1)+j|k+1} \in \mathcal{Z}_j(\epsilon, \epsilon)$ for all $j \in \{0, \ldots, N\}$. Consider any $0 \le j < N$. (6) states that $\mathcal{Z}_{j+1}\left(\epsilon[k], \epsilon\right) = \mathcal{Z}_j\left(\epsilon, \epsilon\right) \ominus L_j\hat{F}\widehat{\mathcal{W}}\left(\epsilon[k], \epsilon\right)$. From the construction of the candidate solution we have $\overline{z}_{k+j+1|k+1} = \overline{z}_{k+j+1|k}^\star + L_j\hat{F}\hat{w}_k$, where $\hat{w}_k \in \widehat{\mathcal{W}}\left(\epsilon[k], \epsilon\right)$ and $\overline{z}_{k+j+1|k}^\star \in \mathcal{Z}_{j+1}\left(\epsilon[k], \epsilon\right)$. By definition of the Pontryagin difference, we conclude that $\overline{z}_{k+j+1|k+1} \in \mathcal{Z}_j\left(\epsilon, \epsilon\right)$ for all $j \in \{0, \ldots, N-1\}$.

At $j = N$ the candidate solution in (12a) gives us $\overline{z}_{(k+1)+N|k+1} = \overline{z}_{k+N+1|k}^\star + L_N\hat{F}\hat{w}_k$. Because $\overline{z}_{k+N+1|k}^\star \in \mathcal{Z}_f\left(\epsilon[k], \epsilon\right) = \mathcal{C}\left(\delta, \epsilon\right) \ominus L_N\hat{F}\widehat{\mathcal{W}}\left(\epsilon[k], \epsilon\right)$ and $\hat{w}_k \in \widehat{\mathcal{W}}\left(\epsilon[k], \epsilon\right)$, we have $\overline{z}_{(k+1)+N|k+1} \in \mathcal{C}\left(\delta, \epsilon\right)$. The definition of $\mathcal{C}\left(\delta, \epsilon\right)$ in (10) implies $\mathcal{C}\left(\delta, \epsilon\right) \subseteq \mathcal{Z}_N\left(\epsilon, \epsilon\right)$. Therefore $\overline{z}_{(k+1)+N|k+1} \in \mathcal{Z}_N\left(\epsilon, \epsilon\right)$.

Terminal constraint: We need to show that $\overline{z}_{k+N+2|k+1} \in \mathcal{Z}_f\left(\epsilon, \epsilon\right) = \mathcal{C}\left(\delta, \epsilon\right) \ominus L_N\hat{F}\widehat{\mathcal{W}}\left(\epsilon, \epsilon\right)$. Add $L_N\hat{F}\hat{w}$, for any $w \in \widehat{\mathcal{W}}\left(\epsilon, \epsilon\right)$, to both sides of (12b) and note that $u_{k+N+1|k+1} = \kappa\left(\overline{z}_{k+N+1|k+1}\right)$, we have

$$\overline{z}_{k+N+2|k+1} + L_N\hat{F}\hat{w} = \hat{A}\left(\delta\right)\overline{z}_{k+N+1|k+1}$$
$$+ \hat{B}\left(\delta\right)\kappa\left(\overline{z}_{k+N+1|k+1}\right) + L_N\hat{F}\hat{w}.$$

It follows from $\overline{z}_{k+N+1|k+1} \in \mathcal{C}\left(\delta, \epsilon\right)$ and from the definition of the invariant control invariant admissible set $\mathcal{C}\left(\delta, \epsilon\right)$ (Eq.(10)) that $\overline{z}_{k+N+2|k+1} + L_N\hat{F}\hat{w} \in \mathcal{C}\left(\delta, \epsilon\right)$ for all $w \in \widehat{\mathcal{W}}\left(\epsilon, \epsilon\right)$. Then by definition of the Pontryagin difference, we conclude that $\overline{z}_{k+N+2|k+1} \in \mathcal{C}\left(\delta, \epsilon\right) \ominus L_N\hat{F}\widehat{\mathcal{W}}\left(\epsilon, \epsilon\right) = \mathcal{Z}_f\left(\epsilon, \epsilon\right)$. ∎

The control algorithm in Alg. 1 , in each time step k, solves $\mathbb{P}_{(\delta, \epsilon)}(\hat{x}_k, \delta_k, \epsilon_k, u_{k-1})$ 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 $J_{(\delta, \epsilon)}$. Therefore, during the course of control, the algorithm may switch between the estimation modes in Δ 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.

Theorem 0.2: If at the initial time step there exists $(\delta, \epsilon) \in \Delta$ such that $\mathbb{P}_{(\delta, \epsilon)}(\hat{x}_0, \delta_0, \epsilon_0, u_{0-1})$ is feasible then the system Eq. 1 controlled by Alg. 1 and subjected to disturbances constrained s.t. $w_k \in \mathcal{W}, \forall k \ge 0$ robustly satisfies the state constraint $x_k \in X, \forall k \ge 0$ and the control input constraint $u_k \in U, \forall k \ge 0$, and all subsequent iterations of the algorithm are feasible.

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 $\mathbb{P}_{(\delta_{k+1}, \epsilon_{k+1})}(\hat{x}_k, \delta_k, \epsilon_k, u_{k-1})$ is feasible. By Thm. 0.1, $u_k \in U$ and at the next time step $k+1$, $x_{k+1} \in X$ and $\mathbb{P}_{(\delta_{k+1}, \epsilon_{k+1})}(\hat{x}_{k+1}, \delta_{k+1}, \epsilon_{k+1}, u_{k+1-1})$ is also feasible, hence the algorithm is feasible. Therefore, the Theorem holds by induction. ∎

REFERENCES

[1] M. Boddy and T. Dean, "Solving Time-dependent Planning Problems," *Joint Conf. on AI*, pp. 979–984, 1989.

[2] S. Zilberstein, "Using anytime algorithms in intelligent systems," *AI Magazine*, vol. 17, no. 3, 1996.

[3] M. Likhachev, D. Ferguson, G. Gordon, A. Stentz, and S. Thrun, "Anytime Search in Dynamic Graphs," *Artif. Intell.*, vol. 172, no. 14, pp. 1613–1643, 2008.

[4] M. Wellman and C. L. Liu, "State-Space Abstraction for Anytime Evaluation of Probabilistic Networks," *Conf. on Uncertainty in AI*, 1994.

[5] R. Mangharam and A. Saba, "Anytime Algorithms for GPU Architectures," in *Proc. of the IEEE Real-Time Systems Symposium*, 2011.

[6] D. Quevedo and V. Gupta, "Sequence-based anytime control," *IEEE Trans. Autom. Control*, vol. 58, no. 2, pp. 377–390, Feb 2013.

[7] R. Bhattacharya and G. J. Balas, "Anytime control algorithm: Model reduction approach," *Journal of Guidance and Control*, vol. 27, no. 5, pp. 767–776, 2004.

[8] D. Fontanelli, L. Greco, and A. Bicchi, "Anytime control algorithms for embedded real-time systems," in *Hybrid Systems: Computation and Control*. Springer, 2008, pp. 158–171.

[9] G. Frehse, A. Hamann, S. Quinton, and M. Woehrle, "Formal analysis of timing effects on closed-loop properties of control software," in *Real-Time Systems Symposium (RTSS), 2014 IEEE*, Dec 2014, pp. 53–62.

[10] D. de Niz, L. Wrage, N. Storer, A. Rowe, and R. Rajukar, "On Resource Overbooking in an Unmanned Aerial Vehicle," *IEEE/ACM Third International Conference on Cyber-Physical Systems*, 2012.

[11] R. Rajkumar, C. Lee, J. Lehoczky, and D. Siewiorek, "A Resource Allocation Model for QoS Mgmt." *IEEE RTSS*, 1997.

[12] S. Sidiroglou-Douskos, S. Misailovic, H. Hoffmann, and M. Rinard, "Managing performance vs. accuracy trade-offs with loop perforation," in *Proceedings of the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering*, ser. ESEC/FSE '11. New York, NY, USA: ACM, 2011, pp. 124–134. [Online]. Available: http://doi.acm.org/10.1145/2025113.2025133

[13] M. Carbin, S. Misailovic, and M. C. Rinard, "Verifying quantitative reliability for programs that execute on unreliable hardware," in *Proceedings of the 2013 ACM SIGPLAN International Conference on Object Oriented Programming Systems Languages & Applications*, ser. OOPSLA '13. New York, NY, USA: ACM, 2013, pp. 33–52. [Online]. Available: http://doi.acm.org/10.1145/2509136.2509546

[14] R. St. Amant, A. Yazdanbakhsh, J. Park, B. Thwaites, H. Esmaeilzadeh, A. Hassibi, L. Ceze, and D. Burger, "General-purpose code acceleration with limited-precision analog computation," in *Proceeding of the 41st Annual International Symposium on Computer Architecuture*, ser. ISCA '14. Piscataway, NJ, USA: IEEE Press, 2014, pp. 505–516. [Online]. Available: http://dl.acm.org/citation.cfm?id=2665671.2665746

[15] Y. V. Pant, K. Mohta, H. Abbas, T. X. Nghiem, J. Devietti, and R. Mangharam, "Co-design of anytime computation and robust control (supplemental)," Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA, Tech. Rep. UPenn-ESE-15-324, May 2015, http://repository.upenn.edu/mlab_papers.

[16] E. Camacho and C. Bordons, *Model predictive control*. Springer Verlag, 2004.

[17] A. Richards and J. How, "Robust model predictive control with imperfect information," in *American Control Conference*, 2005, pp. 268–273.

[18] L. Chisci, J. A. Rossiter, and G. Zappa, "Systems with persistent disturbances: predictive control with restricted constraints," *Automatica*, vol. 37, no. 7, pp. 1019–1028, 2001.

[19] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast Semi-Direct Monocular Visual Odometry," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 15–22.

[20] "ODROID-U3," http://odroid.com/, accessed: 2015-05-13.

[21] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.

[22] J. Mattingley and S. Boyd, "Cvxgen: a code generator for embedded convex optimization," *Optimization and Engineering*, 2012.

[23] "ODROID Smart Power," http://odroid.com/, accessed: 2015-05-13.