Supplementary Appendices

Zachary Brei[1], Jonathan Michaux[2], Bohao Zhang[2], Patrick Holmes[2], Ram Vasudevan[1]

Appendix A
Definitions of Operations in Tab. I

This appendix provides definitions for the interval and polynomial zonotope operations given in Tab. I. Before proceeding further, note that for vectors $a, b \in \mathbb{R}^3$, we write the cross product $a \times b$ as $a^{\times} b$, where

$$a^{\times} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}. \tag{A1}$$

A. Interval Operations

The Minkowski sum and difference of $[x]$ and $[y]$ are

$$[x] \oplus [y] = [\underline{x} + \underline{y}, \overline{x} + \overline{y}], \tag{A2}$$

$$[x] \ominus [y] = [\underline{x} - \overline{y}, \overline{x} - \underline{y}]. \tag{A3}$$

The product of $[x]$ and $[y]$ is

$$[x][y] = \left[\min\left(\underline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{x}\overline{y}\right), \max\left(\underline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{x}\overline{y}\right)\right]. \tag{A4}$$

Given a scalar interval $[a]$ or interval matrix $[Y]$ multplied by an interval matrix $[X]$, the element in the i^{th} row and j^{th} column of the product is

$$([a][X])_{ij} = [a][X]_{ij}, \tag{A5}$$

$$([X][Y])_{ij} = \bigoplus_{k=1}^{n} ([X]_{ik}[Y]_{kj}),. \tag{A6}$$

where n is the number of columns of $[X]$ and number of rows of $[Y]$. Lastly, given two interval vectors $[x], [y] \subset \mathbb{R}^3$, their cross product is

$$[x] \otimes [y] = [x]^{\times}[y], \tag{A7}$$

where $[x]^{\times}$ is the skew-symmetric matrix representation of $[x]$ as in (A1) (i.e., a matrix with interval entries).

B. Polynomial Zonotope Operations

Similar to zonotopes, intervals can also be written as polynomial zonotopes. Consider the interval $[z] = [\underline{z}, \overline{z}] \subset \mathbb{R}^n$. We can convert $[z]$ to a polynomial zonotope $\mathbf{z}$ using

$$\mathbf{z} = \frac{\overline{z} + \underline{z}}{2} + \sum_{i=1}^{n} \frac{\overline{z}_i - \underline{z}_i}{2} x_i, \tag{A8}$$

where $x \in [-1, 1]^n$ is the indeterminate vector.

[1]Mechanical Engineering, University of Michigan, Ann Arbor, MI ⟨breizach, ramv⟩@umich.edu.

[2]Robotics, University of Michigan, Ann Arbor, MI ⟨jmichaux, jimzhang, pdholmes, ramv ⟩@umich.edu.

First, the Minkowski Sum of two polynomial zonotopes $\mathbf{P}_1 \subset \mathbb{R}^n = \mathcal{PZ}(g_i, \alpha_i, x)$ and $\mathbf{P}_2 \subset \mathbb{R}^n = \mathcal{PZ}(h_j, \beta_j, y)$ follows from polynomial addition:

$$\mathbf{P}_1 \oplus \mathbf{P}_2 = \{z \in \mathbb{R}^n \mid z = p_1 + p_2, p_1 \in \mathbf{P}_1, p_2 \in \mathbf{P}_2\} \tag{A9}$$

$$= \left\{ z \in \mathbb{R}^n \mid z = \sum_{i=0}^{n_g} g_i x^{\alpha_i} + \sum_{j=0}^{n_h} h_j y^{\beta_j} \right\}. \tag{A10}$$

Similarly, we may write the matrix product of two polynomial zonotopes $\mathbf{P}_1$ and $\mathbf{P}_2$ when the sizes are compatible (i.e., elements in $\mathbf{P}_1$ have the same number of columns as elements of $\mathbf{P}_2$ have rows). Letting $\mathbf{P}_1 \subset \mathbb{R}^{n \times m}$ and $\mathbf{P}_2 \subset \mathbb{R}^{m \times k}$, we obtain $\mathbf{P}_1\mathbf{P}_2 \subset \mathbb{R}^{n \times k}$:

$$\mathbf{P}_1\mathbf{P}_2 = \{z \in \mathbb{R}^{n \times k} \mid z = p_1 p_2, p_1 \in \mathbf{P}_1, p_2 \in \mathbf{P}_2\} \tag{A11}$$

$$= \left\{ z \in \mathbb{R}^{n \times k} \mid z = \sum_{i=0}^{n_g} g_i (\sum_{j=0}^{q} h_j y^{\beta_j}) x^{\alpha_i} \right\}. \tag{A12}$$

When $\mathbf{P}_1 \subset \mathbb{R}^{n \times n}$ is square, exponentiation $\mathbf{P}_1^m$ may be performed by multiplying $\mathbf{P}_1$ by itself m times.

Furthermore, if $\mathbf{P}_1 \subset \mathbb{R}^3$ and $\mathbf{P}_2 \subset \mathbb{R}^3$, we implement a set-based cross product as matrix multiplication. We create $\mathbf{P}_1^{\times} \subset \mathbb{R}^{3 \times 3}$ as

$$\mathbf{P}_1^{\times} = \left\{ A \in \mathbb{R}^{3 \times 3} \mid A = \sum_{i=0}^{n_g} \begin{bmatrix} 0 & -g_{i,3} & g_{i,2} \\ g_{i,3} & 0 & -g_{i,1} \\ -g_{i,2} & g_{i,1} & 0 \end{bmatrix} x^{\alpha_i} \right\} \tag{A13}$$

where $g_{i,j}$ refers to the j^{th} element of g_i. Then, the set-based cross product $\mathbf{P}_1 \otimes \mathbf{P}_2 = \mathbf{P}_1^{\times}\mathbf{P}_2$ is well-defined. We briefly note that the addition, multiplication and cross product of a polynomial zonotope with a constant vector or matrix is well-defined if the constant is appropriately sized. In this case, one constructs a polynomial zonotope with that constant vector or matrix as the center g_0 and no other generators, and applies the definitions above.

Both Minkowski summation and multiplication of polynomial zonotopes can be complicated by the fact that $\mathbf{P}_1$ and $\mathbf{P}_2$ may share indeterminates. For instance, in the examples above, the i-th element of x and the j-th element of y may represent the same indeterminate. In practice, polynomial zonotopes can be brought to a *common representation* by only considering unique indeterminates before applying the operations above, as discussed in [16, Sec. II.a.1].

Given the j^{th} indeterminate x_j and a value $\sigma \in [-1, 1]$, the slicing operation which yields a subset of $\mathbf{P}$ by plugging σ into the specified element x_j and is defined as

$$\texttt{slice}(\mathbf{P}, x_j, \sigma) \subset \mathbf{P} = \left\{ z \in \mathbf{P} \mid z = \sum_{i=0}^{n_g} g_i x^{\alpha_i}, x_j = \sigma \right\}. \tag{A14}$$

In particular, we define the $\texttt{sup}$ and $\texttt{inf}$ operations which return these upper and lower bounds, respectively by taking the absolute values of generators. For $\mathbf{P} \subseteq \mathbb{R}^n$, these return

$$\texttt{sup}(\mathbf{P}) = g_0 + \sum_{i=1}^{n_g} |g_i|, \tag{A15}$$

$$\texttt{inf}(\mathbf{P}) = g_0 - \sum_{i=1}^{n_g} |g_i|. \tag{A16}$$

Note that for any $z \in \mathbf{P}$, $\texttt{sup}(\mathbf{P}) \geq z$ and $\texttt{inf}(\mathbf{P}) \leq z$, where the inequalities are taken element-wise. Recall these bounds may not be tight because possible dependencies between indeterminates are not accounted for, but they are quick to compute.

Though we have defined several basic operations like addition and multiplication above, it may be desirable to use polynomial zonotopes as inputs to more complicated functions. One can overapproximate any analytic function evaluated on a polynomial zonotope using a Taylor expansion, which itself can be represented as a polynomial zonotope [17, Sec 4.1][16, Prop. 13]. Consider an analytic function $f : \mathbb{R} \to \mathbb{R}$ and $\mathbf{P}_1 = \mathcal{PZ}(g_i, \alpha_i, x)$, with each $g_i \in \mathbb{R}$. Then,

$$f(\mathbf{P}_1) = \{y \in \mathbb{R} \mid y = f(z), z \in \mathbf{P}_1\}. \tag{A17}$$

We generate $\mathbf{P}_2$ such that $f(\mathbf{P}_1) \subseteq \mathbf{P}_2$ using a Taylor expansion of degree $d \in \mathbb{N}$, where the error incurred from the finite approximation is overapproximated using a Lagrange remainder. The method follows the Taylor expansion found in the reachability algorithm in [16], which builds on previous work on conservative polynomialization found in [17]. Recall that the Taylor expansion about a point $c \in \mathbb{R}$ is

$$f(z) = \sum_{n=0}^{\infty} \frac{f^{(n)}(c)}{n!}(z-c)^n, \tag{A18}$$

where $f^{(n)}$ is the n^{th} derivative of f. Note that the error incurred by a finite Taylor expansion can be bounded using the Lagrange remainder r [18, p. 7.7]:

$$|f(z) - \sum_{n=0}^{d} \frac{f^{(n)}(c)}{n!}(z-c)^n| \leq r, \tag{A19}$$

where r is given by

$$r = \frac{M|z-c|^{d+1}}{(d+1)!}, \tag{A20}$$

$$M = \max_{\delta \in [c,z]} (|f^{d+1}(\delta)|). \tag{A21}$$

For a polynomial zonotope, the infinite dimensional Taylor expansion is given by

$$f(\mathbf{P}_1) = \sum_{n=0}^{\infty} \frac{f^{(n)}(c)}{n!}(\mathbf{P}_1 - c)^n \tag{A22}$$

In practice, only a finite Taylor expansion of degree $d \in \mathbb{N}$ can be computed. Letting $c = g_0$ (i.e., the center of $\mathbf{P}_1$), and noting that $(z-c) = \sum_{i=1}^{n_g} g_i x^{\alpha_i}$ for $z \in \mathbf{P}_1$, we write

$$\mathbf{P}_2 := \left\{ z \in \mathbb{R} \,|\, z \in \sum_{n=0}^{d} \left(\frac{f^{(n)}(g_0)}{n!} (\sum_{i=1}^{n_g} g_i x^{\alpha_i})^n \right) \oplus [r] \right\} \tag{A23}$$

and the Lagrange remainder $[r]$ can be computed using interval arithmetic as

$$[r] = \frac{[M][(\mathbf{P}_1 - c)^{d+1}]}{(d+1)!}, \tag{A24}$$

$$[M] = f^{(d+1)}([\mathbf{P}_1]) \tag{A25}$$

where $[(\mathbf{P}_1 - c)^{d+1}] = [\texttt{inf}((\mathbf{P}_1 - c)^{d+1}), \texttt{sup}((\mathbf{P}_1 - c)^{d+1})]$ is an overapproximation of $(\mathbf{P}_1 - c)^{d+1}$. Note that $\mathbf{P}_2$ can be expressed as a polynomial zonotope because all terms in the summation are polynomials of x, and the interval $[r]$ can be expressed as a polynomial zonotope as in (A8). Just as we denote polynomial zonotopes using bold symbols, we denote the polynomial zonotope overapproximation of a function evaluated on a zonotope using bold symbols (*i.e.*, $\mathbf{f}(\mathbf{P}_1)$ is the polynomial zonotope over approximation of f applied to $\mathbf{P}$). Note the usual order of operations for addition, multiplication and exponentiation apply for polynomial zonotope operations as well. Table I summarizes these operations. Note as described in the table these operations can either be computed exactly or in an overapproximative fashion using polynomial zonotopes.

The operations defined above (multiplication in particular) increases the number of generators required to represent a polynomial zonotope, therefore increasing the memory required to store a polynomial zonotope. In practice, successively applying these operations can become computationally intractable. To combat this computational burden, we define a *reduce* operation for a polynomial zonotope. The reduce operation generates overapproximations of polynomial zonotopes through using fewer generators. If a polynomial zonotope $\mathbf{P} \subset \mathbb{R}^n$ has n_g terms, but a maximum of q terms are desired, excess terms can be overapproximated by an interval:

$$\texttt{reduce}(\mathbf{P}, n_h) = \left\{ z \in \mathbb{R}^n \mid z \in \sum_{i=0}^{n_h - n} g_i x^{\alpha_i} \oplus [-b, b] \right\} \tag{A26}$$

where b_j is equal to $\sum_{i=n_h-n+1}^{n_g} |g_{i,j}|$ where $g_{i,j}$ is the j^{th} element of g_i. This means that the last $n_g - n_h - n + 1$ terms are overapproximated by an n-dimensional hyperbox represented by an interval. This interval can be expressed as a polynomial zonotope as in (A8), and so the output of $\texttt{reduce}(\mathbf{P}, q)$ is itself a polynomial zonotope. Notice that $\texttt{reduce}(\mathbf{P}, q)$ always overapproximates $\mathbf{P}$, i.e. $\mathbf{P} \subseteq \texttt{reduce}(\mathbf{P}, q)$ [16, Prop. 16]. One can reorder the terms of the polynomial zonotope such that only certain desirable terms are replaced by intervals, e.g. to produce a tighter overapproximation.

Appendix B
Newton-Euler Algorithm

This appendix summarizes the formulas used to recursively compute the angular velocity and acceleration of link j using the angular velocity and acceleration of link $j-1$ and formulas to similarly compute the linear acceleration of each link frame and center of mass of each link. Note that the angular velocity associated with link j expressed in frame $j-1$ is denoted ω_j^j. Then, the Newton-Euler equations can be used to iteratively calculate the forces and moments both at the CoM of each link and at each joint. In particular, we exploit this formulation with the fixed joint between the tray and object in order to calculate the contact wrench. We use this convention for all quantities of interest not just angular velocities. Note for convenience, throughout this appendix, we drop the dependence of the velocity, acceleration, and rotation matrices on the configuration of the robot.

Lemma 10 (Iterative Newton Euler Formulation)**.** *[12, Ch 6] Given the angular velocity of link $j-1$ and the velocity of the robot, one can compute the angular velocity of link j*

$$\omega_j^j = R_{j-1}^j \omega_{j-1}^{j-1} + \dot{q}_j z_j. \tag{B1}$$

where z_j is the rotation axis vector of the j^{th} joint. Similarly, given the angular acceleration of link $j-1$, the angular acceleration of link j is:

$$\dot{\omega}_j^j = R_{j-1}^j \dot{\omega}_{j-1}^{j-1} + (R_{j-1}^j \omega_j^j) \times (\dot{q}_j z_j) + \ddot{q}_j z_j. \tag{B2}$$

The linear acceleration of each link frame is then

$$\dot{v}_j^j = (R_{j-1}^j \dot{v}_{j-1}^{j-1}) + (\dot{\omega}_j^j \times p_j^{j-1}) + \left(\omega_j^j \times (\omega_j^j \times p_j^{j-1})\right), \tag{B3}$$

and the linear acceleration of the CoM of link j is

$$\dot{v}_{CoM,j}^j = \dot{v}_j^j + (\dot{\omega}_j^j \times p_{CoM,j}^j) + \left(\omega_j^j \times (\omega_{a,j}^j \times p_{CoM,j}^j)\right). \tag{B4}$$

Then the inertial force and torque acting at the center of mass of each link is:

$$F_j^j = m_j \dot{v}_{j,CoM}^j \tag{B5}$$

$$N_j^j = I_j \dot{\omega}_j^j + \omega_j^j \times \left(I \omega_j^j\right) \tag{B6}$$

where I_j is the spatial inertia matrix of the j^{th} link about it's CoM. In addition, the forces and moments acting on the j^{th} link can be defined as:

$$f_j^j = R_{j+1}^j f_{j+1}^{j+1} + F_j^j, \tag{B7}$$

$$n_j^j = R_{j+1}^j n_{j+1}^{j+1} + c_j^j \times F_j^j + N_j^j + \left(p_{j+1}^j \times \left(R_{j+1}^j f_{j+1}^{j+1}\right)\right) \tag{B8}$$

Then the wrench exerted by the $(j-1)^{th}$ link onto the j^{th} link, through the j^{th} joint, is $\mathrm{w}_j = \begin{bmatrix} f_j^j \\ n_j^j \end{bmatrix}$.

When implementing these equations, a Recursive Newton Euler Algorithm (RNEA) is used. Eqs. (B1)-(B4) are performed on the forward pass while Eqs. (B7)-(B8) are performed on the backwards pass, as seen in Alg. 1. Note that the base case for the backwards pass requires initialization of $f_{n_q+2}^{n_q+2}$, $n_{n_q+2}^{n_q+2}$ and $R_{n_q+2}^{n_q}$. Since we assume no external wrenches are applied to the object, these are initialized to zero vectors for $f_{n_q+2}^{n_q+2}$ and $n_{n_q+2}^{n_q+2}$ and an identity matrix for $R_{n_q+2}^{n_q}$. Further, the effect of gravity on each link is accounted for by initializing the base joint's acceleration to be $a_0^0 = (0,0,9.81)^\top$ m s^{-2} [12, Sec. 6.5].

Appendix C
Proof of Lemma 4 and Polynomial Zonotope Formulation of the Constraints

This appendix walks through the formulation of the constraints to prevent relative motion and how they are overapproximated using polynomial zonotopes. The constraints have to ensure that the object being manipulated does not move relative to the supporting tray. Therefore, all six degrees of freedom of the object's motion have to be addressed. In order to form the polynomial zonotope overapproximations, the overapproximation of the contact wrench, $\mathbf{w_o}(\mathbf{q_A}(\mathbf{T_i};\mathbf{K}),[\Delta])$, is used.

A. Vertical Separation Constraint

The first degree of freedom to constrain is translation in the $\hat{z}_o$ direction, shown in Fig. 2. Translation *into* the tray is prevented by a reaction normal force from the tray, which is assumed to be a rigid body. The first constraint on the contract wrench prevents translation *away* from the tray surface. A violation of this constraint means that the two bodies separate from one another, resulting in no normal force between the two bodies. Thus this separation constraint can be written as

$$-\mathrm{f}_{o,z}(q_A(t;k),\Delta) \leq 0, \tag{C1}$$

where $\mathrm{f}_{o,z}(q_A(t;k),\Delta)$ is the vertical force component of the contact wrench expressed in the contact frame, thus it corresponds to the normal force. Satisfaction of this constraint means that a normal force exists, and therefore the object is not translating vertically relative to the tray.

The normal component of the contact force is overapproximated by $\mathbf{f_{o,z}}(\mathbf{q_A}(\mathbf{T_i};k),[\Delta])$, i.e., for each $k \in \mathbf{K}$

$$f_{o,z}(q_A(t;k),\Delta) \in \mathbf{f_{o,z}}(\mathbf{q_A}(\mathbf{T_i};k),[\Delta]), \quad \forall t \in \mathbf{T_i} \tag{C2}$$

Plugging the overapproximation of the normal force into (3):

$$\mathbf{h_{sep}}(\mathbf{w_{n_q+1}^{n_q+1}}(\mathbf{q_A}(\mathbf{T_i};k),[\Delta])) := -\mathbf{f_{o,z}}(\mathbf{q_A}(\mathbf{T_i};k),[\Delta])) \tag{C3}$$

In order to ensure satisfaction of (7), the polynomial zonotope version of the constraint entails choosing k such that

$$\texttt{sup}(\mathbf{h_{sep}}(\mathbf{w_{n_q+1}^{n_q+1}}(\mathbf{q_A}(\mathbf{T_i};k),[\Delta]))) \leq 0, \quad \forall i \in \{1,\ldots,n_t\} \tag{C4}$$

Since this constraint is conservatively overapproximative of the actual normal force applied, it is guaranteed that there is a normal force between the loose object and tray when this constraint is satisfied.

B. Linear Slipping Constraint

The next two degrees of freedom, translation in the $\hat{x}_o$ and $\hat{y}_o$ directions, can be constrained using a standard Coulomb static friction law. The static friction law is formed using the both the normal and tangential components of the contact force. Normally, the tangential component would be calculated by taking the norm of the planar contact force components. However, a square root operation for polynomial zonotopes does not currently exist. Therefore, we reformulate the static friction law so that a polynomial zonotope version can be written. The reformulation is as follows:

$$|\mathrm{f}_{o,T}(q_A(t;k),\Delta)| \leq \mu_s |\mathrm{f}_{o,z}(q_A(t;k),\Delta)| \tag{C5}$$

is equivalent to

$$\mathrm{f}_{o,T}((q_A(t;k),\Delta))^2 \leq \mu_s^2 \mathrm{f}_{o,z}((q_A(t;k),\Delta))^2 \tag{C6}$$

which can be expanded to

$$\left(\sqrt{\mathrm{f}_{o,x}((q_A(t;k),\Delta))^2 + \mathrm{f}_{o,y}((q_A(t;k),\Delta))^2}\right)^2 \tag{C7}$$

$$\leq \mu_s^2 \mathrm{f}_{o,z}((q_A(t;k),\Delta))^2 \tag{C8}$$

with the final reformulated slipping constraint written as:

$$\mathrm{f}_{o,x}((q_A(t;k),\Delta))^2 + \mathrm{f}_{o,y}((q_A(t;k),\Delta))^2 \tag{C9}$$

$$-\mu_s^2 \mathrm{f}_{o,z}((q_A(t;k),\Delta))^2 \leq 0 \tag{C10}$$

This constraint requires the tangential components of the contact force to lie within the static friction cone. Satisfaction of this constraint means that there is no relative linear slip between the object and supporting surface. Note that we do not consider rotational friction for simplicity, but extending the formulation to include such a constraint is a straightforward extension.

Like the normal component of the contact force, the tangential components are overapproximated by elements of $\mathbf{w_o}(\mathbf{T_i};\mathbf{K})$. In particular, for each $k \in K$

$$\begin{aligned} f_{o,x}(q_A(t;k),\Delta) &\in \mathbf{f_{o,x}}(\mathbf{q_A}(\mathbf{T_i};k),[\Delta]) \quad \forall t \in \mathbf{T_i} \\ f_{o,y}(q_A(t;k),\Delta) &\in \mathbf{f_{o,y}}(\mathbf{q_A}(\mathbf{T_i};k),[\Delta]) \quad \forall t \in \mathbf{T_i} \end{aligned} \tag{C11}$$

The polynomial zonotope terms can be substituted into (4).

$$\mathbf{h_{slip}}(\mathbf{w^{n_q+1}_{n_q+1}}(\mathbf{q_A}(\mathbf{T_i};k),[\Delta])) := \tag{C12}$$

$$\mathbf{f_{o,x}}(\mathbf{q_A}(\mathbf{T_i};\mathbf{K}),[\Delta])\mathbf{f_{o,x}}(\mathbf{q_A}(\mathbf{T_i};\mathbf{K}),[\Delta]) \tag{C13}$$

$$\oplus \mathbf{f_{o,y}}(\mathbf{q_A}(\mathbf{T_i};\mathbf{K}),[\Delta])\mathbf{f_{o,y}}(\mathbf{q_A}(\mathbf{T_i};\mathbf{K}),[\Delta]) \tag{C14}$$

$$\ominus \mu_s^2 \mathbf{f_{o,z}}(\mathbf{q_A}(\mathbf{T_i};\mathbf{K}),[\Delta])\mathbf{f_{o,z}}(\mathbf{q_A}(\mathbf{T_i};\mathbf{K}),[\Delta]) \tag{C15}$$

Then, the polynomial zonotope version of the constraint in (8) can be written as

$$\texttt{sup}(\mathbf{h_{slip}}(\mathbf{w^{n_q+1}_{n_q+1}}(\mathbf{q_A}(\mathbf{T_i};k),[\Delta]))) \leq 0 \tag{C16}$$

and by choosing k such that (C16) is satisfied $\forall i \in \{1,\dots,n_t\}$, it is guaranteed that the object does not translate tangentially to the support surface.

Note that the coefficient of static friction μ_s could be uncertain, meaning that the actual coefficient exists in an interval $\mu_s \in [\mu_{s,lower}, \mu_{s,upper}]$. However, it is only necessary to consider the smallest possible value to ensure no slip occurs, so in (C12), $\mu_s = \mu_{s,lower}$, since the static coefficient of friction is always a positive number.

C. Tipping Constraint

Finally, we must constrain the last two degrees of freedom, which are rotation about $\hat{x}_o$ and $\hat{y}_o$. Motion about these axes corresponds to the object tipping over. To prevent this, we use a Zero Moment Point (ZMP) constraint, which requires that the ZMP point exists inside the convex hull of the contact area of the object [19]. This ensures that the normal component of the contact force can apply a sufficient counteracting moment to balance the gravito-inertial wrench of the object. First, let the gravito-inertial wrench acting on the manipulated object be defined as

$$\tilde{\mathrm{w}}_o(q_A(t;k),\Delta) = \begin{bmatrix} \tilde{\mathrm{f}}(q_A(t;k),\Delta) \\ \tilde{\mathrm{n}}(q_A(t;k),\Delta) \end{bmatrix}, \tag{C17}$$

where this wrench, $\tilde{\mathrm{w}}_o(q_A(t;k),\Delta) \in \mathbb{R}^6$, is described in the frame associated with joint o. Next, the vector from p_{CoM} to the ZMP point is [19, Sec. 2]:

$$p_{\mathrm{ZMP}}(q_A(t;k),\Delta) = \frac{\hat{n} \times \tilde{\mathrm{n}}_{p_{\mathrm{CoM}}}(q_A(t;k),\Delta)}{\hat{n} \cdot \tilde{\mathrm{f}}_{p_{\mathrm{CoM}}}(q_A(t;k),\Delta)}, \tag{C18}$$

where $\tilde{\mathrm{n}}_{p_{\mathrm{CoM}}}(q_A(t;k),\Delta)$ is the gravito-inertial moment acting on the object about p_{CoM}, and $\tilde{\mathrm{f}}_{p_{\mathrm{CoM}}}(q_A(t;k),\Delta)$ is the gravito-inertial force acting on p_{CoM}.

Next, there are only two wrenches acting on the object, the wrench applied by the manipulator and the gravito-inertial wrench of the object. For there to be no relative motion, these two wrenches must balance each other. Therefore $\tilde{\mathrm{n}}_{p_{\mathrm{CoM}}}(q_A(t;k),\Delta) = -\mathrm{n}_o(q_A(t;k),\Delta)$ and $\tilde{\mathrm{f}}_{p_{\mathrm{CoM}}}(q_A(t;k),\Delta) = -\mathrm{f}_o(q_A(t;k),\Delta)$. Substituting these terms in (C18) yields

$$p_{\mathrm{ZMP}}(q_A(t;k),\Delta) = \frac{\hat{n} \times \mathrm{n}_o(q_A(t;k),\Delta)}{\hat{n} \cdot \mathrm{f}_o(q_A(t;k),\Delta)}. \tag{C19}$$

Note that the contact frame is located at p_{CoM}, so (C19) gives the position of the ZMP point with respect to the origin of the contact frame. Using the description of the contact patch as in Ass. 2, the ZMP constraint can be written as:

$$\left\| \frac{\hat{n} \times \mathrm{n}_o(q_A(t;k),\Delta)}{(\hat{n} \cdot \mathrm{f}_o(q_A(t;k),\Delta))} \right\|_2 \leq r \tag{C20}$$

Note that the denominator is a scalar quantity, and so (C20) can be rewritten as

$$\left| \frac{1}{(\hat{n} \cdot \mathrm{f}_o(q_A(t;k),\Delta))} \right| * \|\hat{n} \times \mathrm{n}_o(q_A(t;k),\Delta)\|_2 \leq r \tag{C21}$$

which is equivalent to

$$\|\hat{n} \times \mathrm{n}_o(q_A(t;k),\Delta)\|_2 - r|\hat{n} \cdot \mathrm{f}_o(q_A(t;k),\Delta)| \leq 0 \tag{C22}$$

If this constraint is satisfied, then the ZMP point stays inside the circular contact area and the object does not rotate about the $\hat{x}_o$ and $\hat{y}_o$ axes.

The tipping constraint must also be reformulated to work with polynomial zonotope objects. The calculation of the tipping constraint, as shown in (5), requires a square root operation in order to evaluate the l^2-norm, which does not currently exist for polynomial zonotopes. Therefore, we rewrite the constraint as follows:

$$\|\hat{n} \times \mathrm{n}_o(q_A(t;k),\Delta)\|_2 - r|\hat{n} \cdot \mathrm{f}_o(q_A(t;k),\Delta)| \leq 0 \tag{C23}$$

$$\|\hat{n} \times \mathrm{n}_o(q_A(t;k),\Delta)\|_2 \leq r|\hat{n} \cdot \mathrm{f}_o(q_A(t;k),\Delta)| \tag{C24}$$

$$\sqrt{(\hat{n} \times \mathrm{n}_o(q_A(t;k),\Delta))^2} \leq r|\hat{n} \cdot \mathrm{f}_o(q_A(t;k),\Delta)| \tag{C25}$$

$$(\hat{n} \times \mathrm{n}_o(q_A(t;k),\Delta))^2 \leq r^2(\hat{n} \cdot \mathrm{f}_o(q_A(t;k),\Delta))^2 \tag{C26}$$

$$(\hat{n} \times \mathrm{n}_o(q_A(t;k),\Delta))^2 - r^2(\hat{n} \cdot \mathrm{f}_o(q_A(t;k),\Delta))^2 \leq 0 \tag{C27}$$

Overapproximations of these components are calculated in order to form the polynomial zonotope overapproximation of the tipping constraint. First, we have that for each $k \in \mathbf{K}$

$$\mathrm{w}_o(q_A(t;k),\Delta) \in \mathbf{w_o}(\mathbf{q_A}(\mathbf{T_i};k),[\Delta]) \quad \forall t \in \mathbf{T_i} \tag{C28}$$

This means that the contact force and moment vector can be overapproximated by the corresponding components of the wrench overapproximation

$$\begin{aligned} \mathrm{n}_o(q_A(t;k),\Delta) &\in \mathbf{n_o}(\mathbf{q_A}(\mathbf{T_i};k),[\Delta]) \quad \forall t \in \mathbf{T_i} \\ \mathrm{f}_o(q_A(t;k),\Delta) &\in \mathbf{f_o}(\mathbf{q_A}(\mathbf{T_i};k),[\Delta]) \quad \forall t \in \mathbf{T_i} \end{aligned} \tag{C29}$$

The contact force and moment vector overapproximations can be substituted into (C27). The polynomial zonotope overapproximation of the cross product in (C27) is

$$\hat{n} \otimes \mathbf{n_o}(\mathbf{q_A}(\mathbf{T_i};\mathbf{K}),[\Delta]) = \begin{bmatrix} \mathbf{d_1}(\mathbf{q_A}(\mathbf{T_i};\mathbf{K}),[\Delta]) \\ \mathbf{d_2}(\mathbf{q_A}(\mathbf{T_i};\mathbf{K}),[\Delta]) \\ 0 \end{bmatrix} \tag{C30}$$

and the overapproximation of the dot product in (C27) is

$$\hat{n} \odot \mathbf{f_o}(\mathbf{q_A}(\mathbf{T_i};\mathbf{K}),[\Delta]) = \mathbf{d_3}(\mathbf{q_A}(\mathbf{T_i};\mathbf{K}),[\Delta]) \tag{C31}$$

Thus an overapproximation of (5) can be written as

$$\mathbf{h_{tip}}(\mathbf{w_{n_q+1}^{n_q+1}}(\mathbf{q_A}(\mathbf{T_i};k),[\Delta])) = \tag{C32}$$
$$\mathbf{d_1}(\mathbf{q_A}(\mathbf{T_i};k),[\Delta])\mathbf{d_1}(\mathbf{q_A}(\mathbf{T_i};k),[\Delta]) \tag{C33}$$
$$\oplus\, \mathbf{d_2}(\mathbf{q_A}(\mathbf{T_i};k),[\Delta])\mathbf{d_2}(\mathbf{q_A}(\mathbf{T_i};k),[\Delta]) \tag{C34}$$
$$\ominus\, \mathbf{d_3}(\mathbf{q_A}(\mathbf{T_i};k),[\Delta])\mathbf{d_3}(\mathbf{q_A}(\mathbf{T_i};k),[\Delta])r^2 \tag{C35}$$

Then, the polynomial zonotope version of the constraint can be written as

$$\mathtt{sup}(\mathbf{h_{tip}}(\mathbf{w_{n_q+1}^{n_q+1}}(\mathbf{q_A}(\mathbf{T_i};k),[\Delta]))) \leq 0 \tag{C36}$$

By choosing k such that (C36) is satisfied $\forall i \in \{1,...,n_t\}$, it is guaranteed that the object will not tip over. Similar to the coefficient of friction, the radius of the contact area could be uncertain but only the smallest possible value needs to be considered in order to guarantee that no tipping occurs. Therefore, in (C32), if $r \in [r_{lower}, r_{upper}]$, then $r = r_{lower}$, since the radius must be a positive quantity.

APPENDIX D
CONTROLLER IMPLEMENTATION DETAILS

We use the same controller as presented in [9, Sec. VII], with a different K_r, V_M and σ_m. In particular, we let $V_M = 2.0 \times 10^{-2}$ which together with the bound on the smallest eigenvalue σ_m, yields the uniform bound $||r|| \leq \sqrt{\frac{2V_M}{\sigma_m}} = \sqrt{\frac{2\times2.0\times10^{-2}}{8.0386}} \approx 0.0705$. Applying Lem. 5 yields $\varepsilon_{p,j} \approx 0.0176$ rad and $\varepsilon_v \approx 0.1411$ rad/s. We let $K_r = 4I_{7x7}$ where I_{7x7} is a $7x7$ identity matrix.