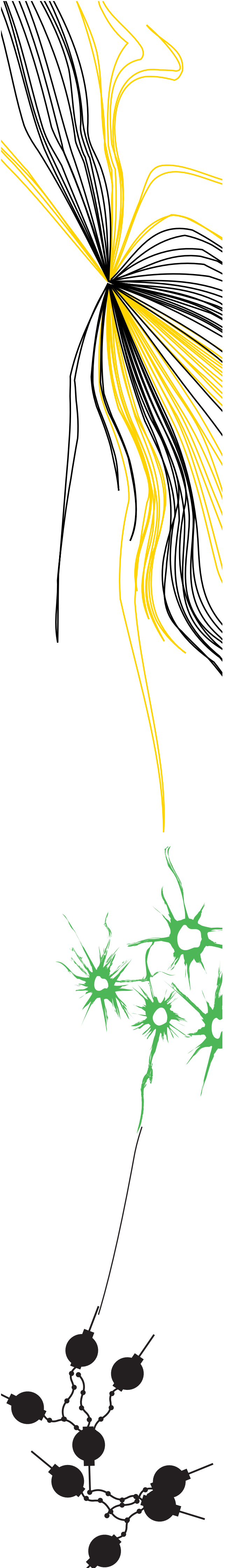

JERK-BASED CONTROL OF AERIAL MANIPULATORS IN PHYSICAL INTERACTION WITH THE ENVIRONMENT

T. (Thissa) Bastiaens

MSC ASSIGNMENT

Committee:
prof. dr. ir. A. Franchi
ir. A.N.M.G. Afifi
C. Gabellieri, Ph.D
dr. ir. W.B.J. Hakvoort

July, 2023

042RaM2023
Robotics and Mechatronics
EEMathCS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Contents

Abstract

For complex interaction tasks, in which there exist multiple constraints that must be satisfied for successful execution of task, it is common to formulate these control problems as quadratic programming (QP) problems. Recently in the humanoid robotics literature, these type of problems is formulated at the jerk level where feedback from force/torque (FT) sensors plays a pivotal roll. When jerk-based control is implemented on aerial manipulators, there is additionally feedback from an inertial measurement unit (IMU). This paper demonstrates that including feedback from this sensor improves performance on both motion and force tasks and that it improves robustness against noise and modelling errors.

1 Introduction

1.1 Jerk-based control

For complex interaction tasks, in which there exist multiple constraints that must be satisfied for successful execution of task, it is common to formulate these control problems as Quadratic Programming (QP) problems. Where the optimization variables are the system accelerations and forces. Recently in the humanoid robotics literature, these type of problems is formulated at the jerk level. At the jerk level, the optimazation variables are instead the system jerks and the derivative of the forces. Jerk-based controllers are promising strategies for humanoid robots balancing and basic locomotion[4]. The key feature of such control strategies is to include force feedback from force/torque (FT) sensors, a fundamental piece of information for robots performing physical interaction tasks. FT sensor feedback is integrated in the control design through a parametrization of the contact forces, ensuring also the maintenance of contact stability conditions. There are still open questions that need to be resolved. Firstly, it is required to investigate and possibly enforce the robustness of jerk controllers with regard to noisy measurements and modelling errors. Secondly, the parameterization of the contact stability conditions could be revised and extended to different contact conditions, for example, sliding along a surface while maintaining contact. Lastly, the controller design and stability analysis may be extended such that the parameterization also include limits relating to actuation, such as joint and torque limits.

1.2 Aerial manipulators

Jerk-based control design could find valuable and novel applications for the control of aerial manipulators: these robots are designed to perform physical interaction tasks, such as object manipulation and polishing[7]. The tasks involve a type of contact that may be different from the type used in the locomation task of humanoid robots. For example, a task may be exerting a specific force on the environment with an end-effector or to make the end-effector slide over a surface while maintaining contact. These Unmanned Aerial Vehicles (UAV's) often come equipped, by default, with a different suite of sensors that is absent from the previously mentioned humanoid robots. Next to FT sensors, they often contain Inertial Measurement Units (IMU's) and gyroscope's, which may prove valuable in jerk-based controllers. The final difference is that UAV's are often kinematically less complex than humanoid robots.

1.3 Research question

The goal of this paper is to further extend the theoretical framework of jerk-based control and experimentally validate the new extension on the aerial manipulator Fiber-THex present at the RAM laboratory. Specifically the aim is to investigate the implications of include IMU sensor information, which leads to the following research questions:

1. What are the advantage and disadvantages performance-wise, of the jerk-level controller, compared to the acceleration-level controller for UAV's with IMU's and FT sensors?
2. What are the advantages and disadvantages to robustness against noise and modelling errors, of the jerk-level controller, compared to the acceleration-level controller for UAV's with IMU's and FT sensors?
3. What are the advantages and disadvantages, practically and computationally, of the jerk-level controller when applied to UAV's, compared to humanoid robots?

1.4 Tasks

To answer these questions a set of tasks have to be defined on which advantages and disadvantaged can be defined and measured:

1. Unconstrained. In this task there is no contact with the environment; the UAV is in free flight and the task is motion tracking
2. Constrained: In this task the UAV end-effector is to make stable contact with the environment and to exert a specified force.
3. Push and slide: In this task, the UAV end-effector is to slide along the surface while maintaining stable contact. There is thus both motion and force tracking.

2 Background

The translational dynamics are expressed in the inertial frame and the rotational dynamics are expressed in the body-fixed frame. The equations of motion (EoM) of a 3D rigidbody, written using the Newton-Euler formalism are given by:[8]

$$\mathbf{M}\mathbf{a} = \mathbf{w}_g + \mathbf{w}_c + \mathbf{w}_a + \mathbf{w}_i + \mathbf{w}_{ext} \quad \text{with} \quad \mathbf{M} = \begin{bmatrix} \mathbf{m} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{1}$$

In these equations, $\mathbf{m} = \text{diag}\,(m, m, m)$ represent the mass matrix where $m \in \mathbb{R}$ is the mass of the UAV. Furthermore $\mathbf{I} = \text{diag}\,(I_{xx}, I_{yy}, I_{zz}) \in \mathbb{R}^{3\times 3}$ represent the inertia matrix, where I_{ii} represents the moment of inertia of the UAV about the i axis. Lastly, $\mathbf{w}_g$, $\mathbf{w}_c$, $\mathbf{w}_a$ and $\mathbf{w}_i$ represent the wrenches due to gravity, coriolis effect, actuation and interaction. Finally, $\mathbf{w}_{ext}$ represents the external wrench due to unmodelled effects such as aerodynamics phenomena (e.g., wind effect, blade flapping, cross interference between propellers, and ground/wall effect). For the applications proposed in this paper it is assumed that these effects do not significantly affect the robot dynamics, and that they can therefore can be neglected. Because of the choice in framing, $\dot{\mathbf{M}} = 0$ and thus the derivative of 1 with respect to time is simply given by:

$$\mathbf{M}\mathbf{j} = \dot{\mathbf{w}}_g + \dot{\mathbf{w}}_c + \dot{\mathbf{w}}_a + \dot{\mathbf{w}}_i + \dot{\mathbf{w}}_{ext} \tag{2}$$

The derivation of the wrenches and their derivatives can be found in section 8. In short, the wrench due to actuation is proportional to the propeller speeds squared:

$$\mathbf{w}_a = \mathbf{G}_v \omega_{prop}^2 \tag{3}$$

$$\dot{\mathbf{w}}_a = \mathbf{G}_v \dot{\omega}_{prop}^2 + \dot{\mathbf{G}}_v \omega_{prop}^2 \tag{4}$$

Furthermore, the wrench due to interaction, neglecting viscous effects, is written as a product of the Jacobian and the wrench on the end-effector:

$$\mathbf{w}_i = \mathbf{J}_{con} \mathbf{w}_{con} \tag{5}$$

$$\mathbf{w}_i = \mathbf{J}_{con} \dot{\mathbf{w}_{con}} + \dot{\mathbf{J}}_{con} \mathbf{w}_{con} \tag{6}$$

3 Quadratic Programming

3.1 Introduction

Quadratic programming aims to find a minimum for a cost:

$$\min_{u} \quad \text{cost}\,(u) = \frac{1}{2} u^T H u + c^T u \tag{7}$$

Subject to a set of 0 or more constraints:

$$b_{lower} \leq Au \leq b_{upper} \tag{8}$$

In this formulation, u is the search variable, H is the Hessian and c is the gradient. Multiple types of constraint can be given:

- if $b_{lower} \neq b_{upper}$ the constraint is an inequality constraint
- if $b_{lower} = b_{upper}$ the constraint is an inequality constraint
- if $A = \mathbf{I}$ the constraint directly bounds on the search variable

3.2 Acceleration level

3.2.1 Unconstrained formulation

The search variable is defined as $\mathbf{u} = \begin{bmatrix} \mathbf{a} & \omega_{prop}^2 \end{bmatrix}^T$ and the weighted cost function that full fills the tracking task is:

$$\text{cost}_{motion}^{acc}(\mathbf{u}) = (\mathbf{a} - \mathbf{a}^*)^T W_{motion} (\mathbf{a} - \mathbf{a}^*) \tag{9}$$

Where $\mathbf{W}_{motion}$ is a diagonal positive definite matrix of weights: $\mathbf{W}_{motion} = diag\left(W_{tr}^x, W_{tr}^y, W_{tr}^z, W_{or}^\phi, W_{or}^\theta, W_{or}^\psi\right)$. Where W_{tr}^i and W_{or}^i represent the weights corresponding to the translation along the i direction and orientation task along the i axis. This cost function corresponds to the following Hessian and gradient:

$$H_{motion}^{acc} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix}^T \mathbf{W}_{motion} \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} \qquad c_{motion}^{acc} = \left(-\mathbf{a}^{*T} \mathbf{W}_{motion} \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix}\right)^T \tag{10}$$

Equation 1 with $\mathbf{w}_i = \mathbf{0}$ is used to write the equality constraint:

$$\begin{bmatrix} \mathbf{M} & -\mathbf{G}_v \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ \omega_{prop}^2 \end{bmatrix} = \mathbf{w}_g + \mathbf{w}_c \tag{11}$$

There are no bounds on the acceleration and lower- and upperbounds on ω_{prop}^2 arise from lower- and upperbounds on ω_{prop}:

$$\begin{bmatrix}-\infty \\ \omega_{min}^2\end{bmatrix} \leq \begin{bmatrix}\mathbf{a} \\ \omega_{prop}^2\end{bmatrix} \leq \begin{bmatrix}\infty \\ \omega_{max}^2\end{bmatrix} \tag{12}$$

Where $\mathbf{0} < \omega_{min} < \omega_{max}$ are the maximum and minimum propeller speeds respectively.
After solving the QP formulation, the propeller speeds can simply be calculated via: $\omega_{prop} = \sqrt{\omega_{prop}^2}$.

3.2.2 Constrained formulation

In the constrained case the search variable is defined as $\mathbf{u} = \begin{bmatrix}\mathbf{a} & \omega_{prop}^2 & \mathbf{w}_{con}\end{bmatrix}^T$. The cost function for the motion task remains the same and it's Hessian and gradient are given by:

$$H_{motion}^{acc} = \begin{bmatrix}\mathbf{I} & \mathbf{0} & \mathbf{0}\end{bmatrix}^T \mathbf{W}_{motion} \begin{bmatrix}\mathbf{I} & \mathbf{0} & \mathbf{0}\end{bmatrix} \qquad c_{motion}^{acc} = \left(-\mathbf{a}^{*T} \mathbf{W}_{motion} \begin{bmatrix}\mathbf{I} & \mathbf{0} & \mathbf{0}\end{bmatrix}\right)^T \tag{13}$$

Equation 1 is again used to write the equality constraint:

$$\begin{bmatrix}\mathbf{M} & -\mathbf{G}_v & -\mathbf{J}_{con}^T\end{bmatrix} \begin{bmatrix}\mathbf{a} \\ \omega_{prop}^2 \\ \mathbf{w}_{con}\end{bmatrix} = \mathbf{w}_g + \mathbf{w}_c \tag{14}$$

Additional to the bounds described in 12, there are the contact stability conditions $\mathbf{w}_{con} \in \mathcal{K}$. Their derivation, including their linearization that is required for this formulation, can be found in section 8.4.3.

3.2.3 Required information

The desired acceleration $\mathbf{a}^*$ are computed from reference signals and measurements via a simple PD controller, with an additional feed-forward term:

$$\mathbf{a}^* = \begin{bmatrix} a_{ref} + K_v\left(v_{ref} - v_{meas}\right) + K_p\left(p_{ref} - p_{meas}\right) \\ \dot{\omega}_{ref} + K_\omega\left(\omega_{ref} - \omega_{meas}\right) + K_R \text{vmap}\left(R_{ref}, R_{meas}\right)\end{bmatrix} \tag{15}$$

The reference signals must have a finite second order derivative for $\mathbf{a}^*$ to be finite. The variables used for feedback when solving this QP formulation at the acceleration level are thus:

- Position p_{meas}: to compute $\mathbf{a}^*$
- Velocity v_{meas}: to compute $\mathbf{a}^*$
- Orientation R_{meas}: to compute $\mathbf{a}^*$ and $\mathbf{G}_v$
- Angular velocity: ω_{meas}: to compute $\mathbf{a}^*$ and $\mathbf{w}_c$

3.2.4 Limitations

There are several limitations of this formulation:

- First of all, the solution $\mathbf{u}$ may be discontinuous and thus discontinuous in ω_{prop}, for example when making or breaking contact. Discontinuity is infeasible in practice because propellers have inertia and thus cannot instantaneously change speeds.
- Second, the information from the IMU sensor, is not utilized
- Third, information from the FT sensor is not utilized. Two approaches for adderessing this are as follows:
  - The issue can be remedied by adding an equality constraint $\mathbf{w}_{con} = \mathbf{w}_{con}^*$. The desired contact wrench $\mathbf{w}_{con}^*$ can be computed from a measurement $\mathbf{w}_{con}^{meas}$ and reference wrench $\mathbf{w}_{con}^{ref}$ via for example a P controller:

    $$\mathbf{w}_{con}^* = K_{contact}\left(w_{con}^{meas} - w_{con}^{ref}\right) \tag{16}$$

    While this would include FT sensor information, this equality constraint may require the contact stability constraints $\mathbf{w}_{con} \in \mathcal{K}$ to be violated.
  - Another approach one may attempt, is to include a wrench task in the cost function:

$$\text{cost}_{wrench}^{acc}(\mathbf{u}) = \left(\mathbf{w}_{con} - \mathbf{w}_{con}^*\right)^T W_{wrench}\left(\mathbf{w}_{con} - \mathbf{w}_{con}^*\right) \tag{17}$$

$$H_{wrench}^{acc} = \begin{bmatrix}\mathbf{0} & \mathbf{0} & \mathbf{I}\end{bmatrix}^T \mathbf{W}_{wrench} \begin{bmatrix}\mathbf{0} & \mathbf{0I}\end{bmatrix} \tag{18}$$

$$c_{wrench}^{acc} = \left(-\mathbf{w}_{con}^{*T} \mathbf{W}_{wrench} \begin{bmatrix}\mathbf{0} & \mathbf{0} & \mathbf{I}\end{bmatrix}\right)^T \tag{19}$$

Where $\mathbf{W}_{wrench}$ is a diagonal positive definite matrix of weights: $\mathbf{W}_{wrench} = diag\left(W_f^x, W_f^y, W_f^z, W_\tau^\phi, W_\tau^\theta, W_\tau^\psi\right)$. Where W_f^i and W_τ^i represent the weight corresponding to the forces along the i direction and moments about the i axis respectively. The total cost for the problem is simply the sum of the motion task and the wrench task:

$$\text{cost}_{total} = \text{cost}_{motion} + \text{cost}_{wrench} \qquad H_{total} = H_{motion} + H_{wrench} \qquad c_{total} = c_{motion} + c_{wrench} \tag{20}$$

In this formulation, the priorities of the motion task and wrench task become intertwined. The challenge of this method is the tuning of the weights in $\mathbf{W}_{wrench}$ and the gains used to calculate $\mathbf{w}^*_{con}$, with respect to the weights in $\mathbf{W}_{motion}$ and the gain used to calculate $\mathbf{a}^*$.

- The fourth and final probem involves the holonomic constraint that models the interaction:

$$\mathbf{J}_{con}\nu = \mathbf{0} \tag{21}$$

$$\mathbf{J}_{con}\mathbf{a} = -\dot{\mathbf{J}}_{con}\nu \tag{22}$$

  This is because generally, the accelerations $\mathbf{a}$ that are part of the solution $\mathbf{u}$ violate the constraint. There are several options to address this, some of which may be applied simultaneously:

  - The first option is to add 22 as an equality constraint. This exacerbates the previously mentioed discontinuities.
  - The second options is to 22 as an inequality constraint instead: $b_{lower} = -\dot{\mathbf{J}}_{con}\nu - \delta$ and $b_{higher} = -\dot{\mathbf{J}}_{con}\nu + \delta$.
  - The third option is to add it to the cost function: as a heavy-weighted task:

$$\text{cost}^{acc}_{constraint}(\mathbf{u}) = \left(\mathbf{J}_{con}\mathbf{a} + \dot{\mathbf{J}}_{con}\nu\right)^T W_{constraint}\left(\mathbf{J}_{con}\mathbf{a} + \dot{\mathbf{J}}_{con}\nu\right) \tag{23}$$

$$H^{acc}_{constraint} = \begin{bmatrix}\mathbf{J}_{con} & \mathbf{0} & \mathbf{0}\end{bmatrix}^T \mathbf{W}_{constraint}\begin{bmatrix}\mathbf{J}_{con} & \mathbf{0} & \mathbf{0}\end{bmatrix} \tag{24}$$

$$c^{acc}_{constraint} = \left(\dot{\mathbf{J}}_{con}\nu^T \mathbf{W}_{constraint}\begin{bmatrix}\mathbf{J}_{con} & \mathbf{0} & \mathbf{0}\end{bmatrix}\right)^T \tag{25}$$

    The challenge with this approach is choosing the weights for the motion task, wrench task and constraint task.
  - The fourth option is to replace $\mathbf{a}^*$ by $\mathbf{a}^*_{con}$, where $\mathbf{a}^*_{con}$ does satisfy the constraint. For example, when only the position of the end-effector in the i-direction is constrained, one can write:

$$\mathbf{a}^*_x = K_v\left(-v^x_{meas}\right)K_p\left(p^x_{con} - p^x_{meas}\right) \tag{26}$$

    Where p^x_{con} is the position of the UAV at which there is contact between the end-effector and the environment.

3.3 Jerk level

3.3.1 Unconstrained formulation

At the jerk level the search variable becomes $\mathbf{u} = \begin{bmatrix}\mathbf{j} & \dot{\omega}^2_{prop}\end{bmatrix}$ and the cost function changes into:

$$\text{cost}^{jerk}_{free}(\mathbf{u}) = (\mathbf{j} - \mathbf{j}^*)^T W_{motion}(\mathbf{j} - \mathbf{j}^*) \tag{27}$$

Similar to the acceleration level, this yields:

$$H^{jerk}_{motion} = \begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}^T \mathbf{W}_{motion}\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix} \qquad c^{jerk}_{motion} = \left(-\mathbf{a}^{*T}\mathbf{W}_{motion}\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\right)^T \tag{28}$$

Equation 2 with $\dot{\mathbf{w}}_i = \mathbf{0}$ is used to write the equality constraint:

$$\begin{bmatrix}\mathbf{M} & -\dot{\mathbf{G}}_v\end{bmatrix}\begin{bmatrix}\mathbf{j} \\ \dot{\omega}^2_{prop}\end{bmatrix} = \dot{\mathbf{w}}_c + \dot{\mathbf{G}}_v\omega^2_{prop} \tag{29}$$

There are no bounds on the jerk and there are lower- and upperbounds on $\dot{\omega}^2_{prop}$ for 2 reasons.

- First there are practical limitations that relate to the hardware. There are bounds on $\dot{\omega}_{prop}$, which may depend on ω_{prop}, for example, because of a limit to the current going to an electric motor.
- Completely separate from this are bounds that ensure that propeller speeds themselves don't break their bounds. So the bounds of $\dot{\omega}^2_{prop}$ depend on ω^2_{prop}. In order to express these bounds, consider that at the jerk level, time integration of the solution is required to obtain an expression for ω_{prop}:

$$\omega^2_{prop}(t) = \omega^2_{prop}(0) + \int_0^T \dot{\omega}^2_{prop}dt \tag{30}$$

When the forward Euler method is used in the controller as an approximation one may write:

$$\omega_{prop}^{2}(t) \approx \omega_{prop}^{2}(t-1) + \dot{\omega}_{prop}^{2}(t)\,\Delta T \tag{31}$$

Where ΔT is the time-step of the controller. By rewriting, one obtains:

$$\dot{\omega}_{prop}^{2}(t) \approx \frac{\omega_{prop}^{2}(t) - \omega_{prop}^{2}(t-1)}{\Delta T} \tag{32}$$

This expression can then be use to express to lower and upperbounds as:

$$\dot{\omega}_{min}^{2}(t) = \frac{\omega_{min}^{2} - \omega_{prop}^{2}(t-1)}{\Delta T} \tag{33}$$

$$\dot{\omega}_{max}^{2}(t) = \frac{\omega_{max}^{2} - \omega_{prop}^{2}(t-1)}{\Delta T} \tag{34}$$

These 2 effects each produce a set of lower and upperbounds. The lowerbound to be applied is the highest of the 2 lowerbounds and the upperbound to be applied is the lowest of the 2 upperbounds.

3.3.2 Constrained formulation

In the constrained case the search variable is defined as $\mathbf{u} = \begin{bmatrix}\mathbf{j} & \dot{\omega}_{prop}^{2} & \dot{\mathbf{w}}_{con}\end{bmatrix}^{T}$. The cost function for the motion task remains the same and the Hessian and gradient are given by:

$$H_{motion}^{jerk} = \begin{bmatrix}\mathbf{I} & \mathbf{0} & \mathbf{0}\end{bmatrix}^{T} \mathbf{W}_{motion} \begin{bmatrix}\mathbf{I} & \mathbf{0} & \mathbf{0}\end{bmatrix} \qquad c_{motion}^{jerk} = \left(-\mathbf{j}^{*T} \mathbf{W}_{motion} \begin{bmatrix}\mathbf{I} & \mathbf{0} & \mathbf{0}\end{bmatrix}\right)^{T} \tag{35}$$

Equation 2 is again used to write the equality constraint:

$$\begin{bmatrix}\mathbf{M} & -\mathbf{G}_{v} & -\mathbf{J}_{con}^{T}\end{bmatrix} \begin{bmatrix}\mathbf{j} \\ \dot{\omega}_{prop}^{2} \\ \dot{\mathbf{w}}_{con}\end{bmatrix} = \dot{\mathbf{w}}_{c} + \dot{\mathbf{G}}_{v}\mathbf{w}_{prop} + \dot{\mathbf{J}}_{con}\mathbf{w}_{con} \tag{36}$$

Because $\mathbf{w}_{con}$ is no longer a search variable, it is not straightforward to fullfill the contact stability conditions. One may attempt the same strategy involving ΔT as is applied in equation 33 to describe the bounds on $\dot{\mathbf{w}}_{con}$ in terms of $\mathbf{w}_{con}$. Alternatively, it is proposed to use to parametrization of the contact wrench, introduced in section 8.4.5. In order to accomplish this, the search variable is changed into: $\begin{bmatrix}\mathbf{j} & \dot{\omega}_{prop}^{2} & \dot{\xi}\end{bmatrix}^{T}$ and the equality constraint is instead written as:

$$\begin{bmatrix}\mathbf{M} & -\mathbf{G}_{v} & -\mathbf{J}_{con}^{T}\Phi(\xi)\end{bmatrix} \begin{bmatrix}\mathbf{j} \\ \dot{\omega}_{prop}^{2} \\ \dot{\xi}\end{bmatrix} = \dot{\mathbf{w}}_{c} + \dot{\mathbf{G}}_{v}\omega_{prop}^{2} + \dot{\mathbf{J}}_{con}\mathbf{w}_{con} \tag{37}$$

The invertability of $\Phi(\xi)$ plays a role in solving this QP formulation. To compute it, the variable ξ is required, which may be obtained via integration of $\dot{\xi}$ or via inversion of the parametrization: $\xi^{meas} = \gamma^{-1}(\mathbf{w}_{con}^{meas})$. The latter option is preferred, as it allows further information from the FT sensor to be included.
Similar to the constrained formulation at the acceleration level, even more feedback from the FT sensor can be included if a task relating to the wrench is added:

$$\text{cost}_{wrench}^{jerk}(\mathbf{u}) = \left(\dot{\xi} - \dot{\xi}^{*}\right)^{T} W_{wrench} \left(\dot{\xi} - \dot{\xi}^{*}\right) \tag{38}$$

$$H_{wrench}^{jerk} = \begin{bmatrix}\mathbf{0} & \mathbf{0} & \mathbf{I}\end{bmatrix}^{T} \mathbf{W}_{wrench} \begin{bmatrix}\mathbf{0} & \mathbf{0I}\end{bmatrix} \tag{39}$$

$$c_{wrench}^{jerk} = \left(-\dot{\xi}^{*T} \mathbf{W}_{wrench} \begin{bmatrix}\mathbf{0} & \mathbf{0} & \mathbf{I}\end{bmatrix}\right)^{T} \tag{40}$$

The desired parameter relating to the parametrization, $\dot{\xi}^{*}$ is computed by using the inverse of the parametrization:

$$\dot{\xi}^{*} = K_{contact}\left(\xi^{meas} - \xi^{ref}\right) \quad \text{with} \quad \xi^{meas} = \gamma^{-1}\left(\mathbf{w}_{con}^{meas}\right) \quad \text{and} \quad \xi^{ref} = \gamma^{-1}\left(\mathbf{w}_{con}^{ref}\right) \tag{41}$$

3.3.3 Required information

The desired jerk $\mathbf{j}^{*}$ may be computed via a simple PD+ controller with an additional feed-forward term:

$$\mathbf{j}^{*} = \begin{bmatrix}\dddot{p}_{ref} + K_{a}(a_{ref} - a_{meas})\,K_{v}(v_{ref} - v_{meas}) + K_{p}(p_{ref} - p_{meas}) \\ \ddot{\omega}_{ref} + K_{\dot{\omega}}(\dot{\omega}_{ref} - \dot{\omega}_{meas})\,K_{\omega}(\omega_{ref} - \omega_{meas}) + K_{R}\text{vmap}(R_{ref}, R_{meas})\end{bmatrix} \tag{42}$$

The reference signal must have a finite third order derivative for $\mathbf{j}^{*}$ to be finite. Similar to the acceleration level, $\mathbf{j}^{*}$ may have to be adapted for the holonomic constraint to not be violated. When only the position of the end-effector along the x-axis is constrained:

$$\mathbf{j}_{x}^{*} = K_{a}\left(-a_{meas}^{x}\right) K_{v}\left(-v_{meas}^{x}\right) K_{p}\left(p_{con}^{x} - p_{meas}^{x}\right) \tag{43}$$

The variables used for feedback when solving this QP formulation at the jerk level are thus:

- Propeller speeds: ω_{prop}: to compute $\omega^2_{prop} = \omega^2_{prop}$
- Contact wrench: $\mathbf{w}^{meas}_{con}$ to compute ξ and $\dot{\xi}^*$
- Position p_{meas}: to compute $\mathbf{j}^*$
- Velocity v_{meas}: to compute $\mathbf{j}^*$
- Acceleration a_{meas} to compute $\mathbf{j}^*$
- Orientation R_{meas}: to compute $\mathbf{j}^*$, $\mathbf{G}_v$ and $\dot{\mathbf{G}}_v$
- Angular velocity: ω_{meas}: to compute $\mathbf{j}^*$ and $\dot{\mathbf{w}}_c$ and $\dot{\mathbf{G}}_v$
- Angular acceleration: $\dot{\omega}_{meas}$: to compute $\mathbf{j}^*$ and $\dot{\mathbf{w}}_c$

When no measurement of $\dot{\omega}$ is available, it may be obtained by inverting equation 1, which would require ω, ω^2_{prop} and $\mathbf{w}^{meas}_{con}$

3.3.4 Summary

At the acceleration level, the solution is discontinuous in ω^2_{prop}. At the jerk level the solutions are continuous in ω^2_{prop}, even if they may be discontinuous in $\dot{\omega}^2_{prop}$. IMU information is not included at the acceleration level and it is included in the computation of $\mathbf{j}^*$ at the jerk level. FT sensor information is included at the acceleration level when computing $\mathbf{w}^*_{con}$. At the jerk level, it is used to compute ξ, $\dot{\xi}^*$ and may be used to compute $\dot{\omega}$. The latter of these is used to compute $\mathbf{j}^*$ and thus, both the motions and wrench task would benefit from the FT sensor.

4 Simulation

4.1 Motivation

The acceleration- and jerk level controllers were developed and employed in a simulation environment. There are several advantages to doing things in simulation, rather than in real life. First of all, in a simulator there is more control over the measurement noise. In contrast with real life, simulation environment may contain little to no noise and additional noise of various types may be added to measurements with ease. Second, in a simulator there is more control over modelling errors, while in real life experiments, calibration procedures are required. Lastly, simulations may be cheaper, faster and safer.

4.2 Simulink

The controllers are implemented in Matlab Simulink by using qpOASES[3] in an S-function. The plant is also implemented in Matlab Simulink and it consist of the equations of motion and a contact model. In this model, the environment is modelled as a spring with high stiffness, is connected based on the end-effector position. There is also viscous friction when the end-effector moves parallel to the environment. Static, coulomb and any form of torsional friction are not implemented. The mission is defined by a set of position, orientation and force waypoints. The reference signal is computed by interpolating between the waypoints with a quintic polynomial.

5 Results

Several experiments where performed. 3 tasks were tested:

- Unconstrained. In this task there is no contact with the environment; the UAV is in free flight and the task is motion tracking
- Constrained: In this task the UAV end-effector is to make stable contact with the environment and to exert a specified force.
- Push-and-slide: In this task, the UAV end-effector is to slide along the surface while maintaining stable contact. There is thus both motion and force tracking.

3 types of noise were tested:

- No additional noise
- FT: white noise on the FT sensor with $power = 0.01$
- IMU: white noise on the IMU sensor with $power = 0.001$

3 types of modelling errors were tested:

- No modelling errors

- M: The mass matrix used in the controller has a maximum deviation of 7% from the actual mass matrix used in the plant
- G: The wrench map used in the controller has a maximum deviation of 7% from the actual wrench map used in the plant

The results of the acceleration-level experiments can be found in 1 and those of the jerk-level experiments in 2. First results were gather for each task, without noise and without modelling errors. This provides a baselines with which other results can be compared. Then, for each combination of task, noise and modelling error, 25 simulations were ran and the results from each simulation were combined into a single result. This was done using the root mean square (RMS) of the error. The investigated error signals are the error in position, error in orientation and the error relating to the wrench task.

Task	**Noise**	**Model**	RMS_x	RMS_y	RMS_z	RMS_ϕ	RMS_θ	RMS_ψ	RMS_{wrench}
Unconstrained			7.6160e-05	7.7074e-05	7.9361e-05	1.4281e-06	1.4281e-06	1.4281e-06	
Unconstrained		M	3.4465e-04	3.5089e-04	0.0390	3.5083e-06	2.9975e-06	2.4185e-06	
Unconstrained		G	0.0021	0.0023	0.0092	0.0220	0.0278	0.0058	
Constrained			0.0721	1.2981e-04	6.9846e-05	6.1939e-09	1.1782e-07	2.2882e-07	0.4820
Constrained	FT		0.0721	0.0721	0.0721	1.0186e-07	4.8758e-06	3.4030e-06	0.9773
Constrained		M	0.0721	9.0662e-05	0.0383	2.5929e-08	3.6809e-05	2.0635e-07	0.5086
Constrained		G	0.0721	0.0048	0.0089	0.0234	0.0301	0.0058	0.5153
Push-and-slide			0.0822	0.0015	5.4404e-04	1.4074e-07	1.4100e-06	3.7469e-04	0.6484
Push-and-slide	FT		0.0822	0.0019	0.0025	3.1741e-07	7.7199e-06	3.8939e-04	1.2351
Push-and-slide		M	0.0822	0.0014	0.0378	3.9899e-07	4.6644e-05	3.7784e-04	0.7129
Push-and-slide		G	0.0822	0.0077	0.0090	0.0241	0.0315	0.0058	0.7978

Table 1: Results for acceleration-level

Task	**Noise**	**Model**	RMS_x	RMS_y	RMS_z	RMS_ϕ	RMS_θ	RMS_ψ	RMS_{wrench}
Unconstrained			0.0097	0.0097	0.0097	3.5943e-04	3.5943e-04	3.5943e-04	
Unconstrained	IMU		0.0102	0.0115	0.0117	0.0017	0.0016	0.0016	
Unconstrained		M	0.0097	0.0097	0.0103	3.5943e-04	3.5943e-04	3.5943e-04	
Unconstrained		G	0.0098	0.0098	0.0100	0.0081	0.0082	0.0021	
Constrained			0.0721	2.3970e-15	0.0027	1.8334e-15	3.5860e-10	6.9624e-16	0.4244
Constrained	FT		0.0721	6.2053e-04	0.0027	6.9051e-10	2.0350e-06	3.4139e-06	0.5346
Constrained	IMU		0.0721	0.0048	0.0059	0.0016	0.0013	0.0016	0.4974
Constrained		M	0.0721	2.5201e-15	0.0035	1.6267e-15	4.8870e-10	7.1695e-16	0.4315
Constrained		G	0.0721	5.9002e-04	0.0028	0.0103	0.0104	0.0026	0.4267
Push-and-slide			0.0822	0.0048	0.0033	3.3051e-07	1.1002e-05	5.4574e-04	0.7320
Push-and-slide	FT		0.0822	0.0050	0.0033	5.1955e-07	1.3260e-05	6.5297e-04	0.7872
Push-and-slide	IMU		0.0823	0.0053	0.0066	0.0013	0.0012	0.0017	0.8086
Push-and-slide		M	0.0822	0.0046	0.0043	1.1999e-07	1.0256e-05	5.5056e-04	0.7715
Push-and-slide		G	0.0822	0.0051	0.0035	0.0119	0.0120	0.0031	0.7315

Table 2: Results for jerk-level

6 Conclusion

6.1 FT noise

Consider the rightmost column of 1 for the constrained and push-and-slide tasks. By comparing the baseline value with the value where FT noise is included, one finds that the RMS value is almost double. When doing the same comparison for the jerk level and the rightmost column of 2, one finds that while the RMS value is higher for the experiments with FT noise, the decrease in performance is not as bad as in at the acceleration level.

6.2 IMU noise

The introduction of IMU noise decreases the tracking of the orientation task and negatively effects the tracking along the z axis. Furtermore, it has a negative effect on the wrench task.

6.2.1 Mass matrix

At the acceleration level, it can be seen that for each of the tasks, performance is worse when there are modelling errors in M. Both the RMS values relating to relating to translational along z and the RMS values relating to the wrench are bigger when there are modelling error in M. At the jerk level, the performance of motion tracking along the z axis is hardly affected. The performance of the wrench task at the jerk level is affected by the modelling errors in M, just as much as at the acceleration level.

6.2.2 Wrench map

At the acceleration level, modelling errors in the wrench map lead to tracking error in all of the relevant variables. At the jerk level, only the orientation tracking is effected.

7 Discussion

7.1 Experiments

When this project was set up, the initial plan was to use Gazebo as a simulator. This was because it has a more realistic contact model compared to what was implemented in Simulink and because Gazebo comes with a visualization. Furthermore, it was planned perform experiments in real life and not just in simulation. As time went on, the realization dawned that experiments in Gazebo and in real life would take too much time and would fall outside of the scope of this thesis.

7.2 Future work

Future work could include such experiments. Furthermore, the tasks were defined such that only the position along the x-axis was constrained. Future work could include different and more types of interaction, such as having constraints on the orientation. Furthermore, in these experiments, the end-effector was a rigid stick, which meant moments could not be exerted on the environment. If the end-effector were different, one could investigate moment tracking. Lastly, one may investigate having a non-rigid or multi-link manipulator. Having either of these options would mean that $\dot{M} \neq \mathbf{0}$ and $\dot{I} \neq \mathbf{0}$, which would have negative computational and practical connotations.

8 Appendix

8.1 Gravity

The wrench due to gravitational effects and it's derivative are given by:

$$\mathbf{w}_g = \begin{bmatrix} -\mathbf{mg} \\ \mathbf{0} \end{bmatrix} \quad \text{and} \quad \dot{\mathbf{w}}_g = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \tag{44}$$

The inertial frame is oriented in such a way that the gravitational acceleration is $\mathbf{g} = \begin{bmatrix} 0 & 0 & 9.81 \end{bmatrix}^T$.

8.2 Coriolis

The wrench due to Coriolis effects is given by:

$$\mathbf{w}_c = \begin{bmatrix} \mathbf{0} \\ -\omega \times \mathbf{I}\omega \end{bmatrix} \tag{45}$$

Because $\dot{\mathbf{I}} = \mathbf{0}$, the derivative of this wrench with respect to time is simply given by:

$$\dot{\mathbf{w}}_c = \begin{bmatrix} \mathbf{0} \\ -\dot{\omega} \times \mathbf{I}\omega - \omega \times \mathbf{I}\dot{\omega} \end{bmatrix} \tag{46}$$

8.3 Actuation

It is assumed that the wrench due to actuation, expressed in the body-fixed frame, is proportional to the propeller speed squared:

$$\mathbf{w}_a^{bf} = \mathbf{G}\omega_{prop}^2 \tag{47}$$

In this expression, the wrench map $\mathbf{G} \in \mathbb{R}^{6\times 6}$ is full rank and $\dot{\mathbf{G}} = 0$. The wrench due to actuation, where the translational dynamics are expressed in the inertial frame via:

$$\mathbf{w}_a = \mathbf{G}_v\omega_{prop}^2 \quad \text{with} \quad \mathbf{G}_v = \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \mathbf{G} \tag{48}$$

Where $\mathbf{R} \in \mathbb{R}^{3\times 3}$ is the orientation of the body-fixed frame with respect to the inertial frame. The derivative of this actuation wrench with respect to time is given by:

$$\dot{\mathbf{w}}_a = \mathbf{G}_v\dot{\omega}_{prop}^2 + \dot{\mathbf{G}}_v\omega_{prop}^2 \quad \text{with} \quad \dot{\mathbf{G}}_v \begin{bmatrix} \dot{\mathbf{R}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad \text{and} \quad \dot{\mathbf{R}} = \mathbf{R}\tilde{\omega} \tag{49}$$

The tilde-operator is defined as:

$$\tilde{p} = \begin{bmatrix} 0 & -p_z & p_y \\ p_z & 0 & -p_x \\ -p_y & p_x & 0 \end{bmatrix} \quad \text{with} \quad p = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \tag{50}$$

8.4 Contact

It is assumed that the robot interacts with a rigid and planar environment, which imposes constraints on the position and orientation of the end-effector. The vector $\mathbf{s} = \begin{bmatrix} c_x & c_y & c_z & c_\phi & c_\theta & c_\psi \end{bmatrix}$, is introduced where $c_i = 0$ if i is not constrained and $c_i = 1$ if i is constrained. Examples of these different types of interaction include

- Fully constrained: $\mathbf{s} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$: the end-effector position and orientation remain constant w.r.t the inertial frame
- Position constrained $\mathbf{s} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$: the end-effector position is constant w.r.t the inertial frame and it can freely rotate. This case applies to a point contact
- Normal translation constrained: $\mathbf{s} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$: in this case the end-effector position is constrained in the direction of motion normal to the plane spanning the environment. The end-effector may still move parallel to the plane and it may rotate
- Orientations and normal translation constrained $\mathbf{s} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$: the end-effector position is constrained the same as the previous case. The end-effector can rotate about the axis normal to the plane spanning the environment.

8.4.1 Holonomic constraint

To model the interaction in terms of $\mathbf{s}$ a diagonal selector matrix is introduced: $S_{con} = \text{diag}(\mathbf{s})$. The constraints on the end-effector velocities are then described with the selector matrices via:

$$S_{con}\nu_{ee}^{in} = \mathbf{0} \tag{51}$$

In this expression ν_{ee}^{in} are the end-effector velocities expressed in the inertial frame. The Jacobian J maps the robot velocities to the end-effector velocities such that:

$$\nu_{ee}^{in} = \mathbf{J}\nu_{bf}^{in} \tag{52}$$

With:

$$\mathbf{J} = \begin{bmatrix} \mathbf{I} & \mathbf{R}\tilde{\mathbf{p}}_{ee}^{bf} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \quad \text{and} \quad \dot{\mathbf{J}} = \begin{bmatrix} \mathbf{0} & \dot{\mathbf{R}}\tilde{\mathbf{p}}_{ee}^{bf} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{53}$$

$$\tag{54}$$

In this expression ν_{bf}^{in} are the UAV velocities expressed in the inertial frame and $\mathbf{p}_{ee}^{bf}$ is the position of the end-effector expressed in the body-fixed frame with $\dot{\mathbf{p}}_{ee}^{bf} = \mathbf{0}$. Because the rotational dynamics are expressed in the body-fixed frame an extra term is needed to write the end-effector velocities in terms of the system state:

$$\nu_{bf}^{in} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{bmatrix} \nu \tag{55}$$

Combining these expression yields the expression for the holonomic constraint:

$$\mathbf{J}_{con}\nu = \mathbf{0} \quad \text{with} \quad \mathbf{J}_{con} = S_{con}\mathbf{J} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{bmatrix} \tag{56}$$

Taking the derivative yields:

$$\mathbf{J}_{con}\mathbf{a} + \dot{\mathbf{J}}_{con}\nu = \mathbf{0} \tag{57}$$

8.4.2 Wrench

The holonomic constraints are enforced by a contact wrench $\mathbf{w}_{con}$. This wrench acts on the end-effector and has translational and rotational dynamics expressed in the inertial frame. The Jacobian transposed is used to express the wrench on the UAV, with translational and rotational dynamics expressed in the inertial frame:

$$\mathbf{w}_{contact}^{CoM} = (S_{con}\mathbf{J})^T \mathbf{w}_{con} \tag{58}$$

Where $\mathbf{w}_{contact}^{CoM}$ is the wrench on the CoM with both translational and rotational dynamics expressed in the inertial frame. Similar to the previous section an extra term is required to obtain the wrench on the CoM with rotational dynamics expressed in the body-fixed frame:

$$\mathbf{w}_{contact} = \mathbf{J}_{con}^T\mathbf{w}_{con} \quad \text{with} \quad \mathbf{J}_{con}^T = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}^T \end{bmatrix} (S_{con}\mathbf{J})^T \tag{59}$$

There may also be a wrench relating to the unconstrained directions of motion, for example viscous friction. To describe this, the complimentary selector matrix $S_{free} = \text{diag}(\neg\mathbf{s})$ is introduced, where $\neg$is the logical NOT operation:

$$\mathbf{w}_{free} = \mathbf{J}_{free}^T\mathbf{w}_{visc} \quad \text{with} \quad \mathbf{J}_{free}^T = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}^T \end{bmatrix} (S_{free}\mathbf{J})^T \tag{60}$$

The total wrench due to interaction is simply the sum of these components:

$$\mathbf{w}_i = \mathbf{J}_{con}^T\mathbf{w}_{con} + \mathbf{J}_{free}^T\mathbf{w}_{visc} \tag{61}$$

The derivative of this wrench is given by:

$$\mathbf{w}_i = \dot{\mathbf{J}}_{con}^T\mathbf{w}_{con} + \mathbf{J}_{con}^T\dot{\mathbf{w}}_{con} + \dot{\mathbf{J}}_{free}^T\mathbf{w}_{visc} + \mathbf{J}_{free}^T\dot{\mathbf{w}}_{visc} \tag{62}$$

8.4.3 Contact Stability Constraints

In case of stable contact, additional constraints apply. The contact wrench is defined as $\mathbf{w}_{con} = \begin{bmatrix} f_x & f_y & f_z & M_x & M_y & M_z \end{bmatrix}^T$ where f_i, M_i are the corresponding forces and moments in the i direction and along the i axis respectively. Maintaining stable planar unilateral contact is tantamount to enforcing the following constraints:

$$f_x > f_x^{min} \geq 0 \tag{63}$$

$$\sqrt{f_y^2 + f_z^2} < \mu_s f_x \tag{64}$$

$$z_c^{min} < \frac{M_y}{f_x} < z_c^{max} \tag{65}$$

$$y_c^{min} < -\frac{M_z}{f_x} < y_c^{max} \tag{66}$$

$$|\frac{M_x}{f_x}| < \mu_x \tag{67}$$

In which y_c^{min}, y_c^{max}, z_c^{min}, z_c^{max} are the dimensions of the herein assumed rectangular contact surface and μ_s, μ_x are the static and torsional friction coefficients respectively. The different types of interaction relate to different subsets of these equations:

- Fully constrained: eqs. (63) to (67)
- Position constrained: eqs. (63) and (64)
- Normal translation constrained: eq. (63)
- Orientations and normal translation constrained: eqs. (63), (65) and (66)

The constraints define a set $\mathcal{K}$ and thus a wrench obeying to the constraints can be written as $\mathbf{w}_{con}^{ee} \in \mathcal{K}$.

8.4.4 Linearized contact stability conditions

Equation 64 is not linear in f_y and f_z. For reasons that will become clear later, it is desirable to approximate this nonlinear constraint with a set of linear constraints. One can imagine this equation to describe a circle in the f_y, f_z plane with radius $\mu_s f_x$ and the approximation to be a regular polygon, such that the circle circumscribes the polygon.

8.4.5 Parametrization

Again, for reason that will become clear later, it is desirable to find a parametrization $\mathbb{R}^6 \mapsto \hat{\mathcal{K}}$ that approximates the contact stability conditions.

$$\mathbf{w}_{con} = \gamma(\xi) \tag{68}$$

With properties:

- $\hat{\mathcal{K}} \subset \mathcal{K}$, or equivalently $\gamma(\xi) \in \mathcal{K} \forall \xi \in \mathbb{R}^6$
- The function is a bijection, namely a one-to-one correspondence from $\mathbb{R}^6$ to $\hat{\mathcal{K}}$.

was proposed ensuring the satisfaction of eqs. (63) to (67), where

$$\gamma(\xi) = \begin{bmatrix} e^{\xi_1} + f_x^{min} \\ \mu_s \frac{\tanh \xi_2 (e^{\xi_1} + f_x^{min})}{\sqrt{1 + \tanh^2 \xi}} \\ \mu_s \frac{\tanh \xi (e^{\xi_1} + f_x^{min})}{\sqrt{1 + \tanh^2 \xi_2}} \\ (\delta_y \tanh \xi_4 + \delta_{y_0})(e^{\xi_1} + f_x^{min}) \\ (\delta_x \tanh \xi_5 + \delta_{x_0})(e^{\xi_1} + f_x^{min}) \\ \mu_x \tanh \xi_6 (e^{\xi_1} + f_x^{min}) \end{bmatrix} \tag{69}$$

Differentiating 68 yields:

$$\dot{\mathbf{w}}_{con} = \Phi(\xi)\dot{\xi} \tag{70}$$

With:

$$\Phi(\xi) = \left[\frac{\delta\gamma}{\delta\xi_1}, \ldots, \frac{\delta\gamma}{\delta\xi_6}\right] \tag{71}$$

This gradient is invertible $\forall \xi \in \mathbb{R}^6$.

8.5 Cost function manipulation

Imagine a cost function with a symmetric weight matrix W:

$$\mathrm{cost}(u) = (\mathbf{s} - \mathbf{s}^*)^T W (\mathbf{s} - \mathbf{s}^*) \tag{72}$$

Which has to be rewritten in the form:

$$\mathrm{cost}(u) = \frac{1}{2} u^T H u + c^T u \tag{73}$$

Expanding the cost function yields:

$$\mathrm{cost}(u) = \mathbf{s}^T W \mathbf{s} - \mathbf{s}^T W \mathbf{s}^* - \mathbf{s^*}^T W \mathbf{s} + \mathbf{s^*}^T W \mathbf{s}^* \tag{74}$$

Imagine s can be written as $s = \alpha + \beta u$. Inserting this into the cost function yields:

$$\mathrm{cost}(u) = (\alpha + \beta u)^T W (\alpha + \beta u) - (\alpha + \beta u)^T W s^* - {s^*}^T W (\alpha + \beta u) + {s^*}^T W s^* \tag{75}$$

Expanding this yields:

$$\mathrm{cost}(u) = \alpha^T W \alpha + \alpha^T W \beta u + (\beta u)^T W \alpha + (\beta u)^T W \beta u - \alpha^T W s^* - (\beta u)^T W s^* - {s^*}^T W \alpha - {s^*}^T W \beta u + {s^*}^T W s^* \tag{76}$$

The terms without u in them can be removed without changing the solution to the minimization problem:

$$\mathrm{cost}(u) = \alpha^T W \beta u + (\beta u)^T W \alpha + (\beta u)^T W \beta u - (\beta u)^T W s^* - {s^*}^T W \beta u \tag{77}$$

Factoring terms together yields:

$$\mathrm{cost}(u) = (\beta u)^T W \beta u + (\alpha - s^*)^T W \beta u + (\beta u)^T W (\alpha - s^*) \tag{78}$$

Factoring further, using the property that W is symmetric and dividing by 2 yields:

$$\mathrm{cost}(u) = \frac{1}{2} (\beta u)^T W \beta u + (\alpha - s^*)^T W \beta u \tag{79}$$

Thus, the Hessian and gradient are given by:

$$H = \beta^T W \beta \tag{80}$$

$$c = \left((\alpha - s^*)^T W \beta \right)^T \tag{81}$$

References

[1] Davide Bicego, Jacopo Mazzetto, Ruggero Carli, Marcello Farina, and Antonio Franchi. Nonlinear model predictive control with enhanced actuator model for multi-rotor aerial vehicles with generic designs. *J. Intell. Robot. Syst.*, 100(3-4): 1213–1247, December 2020.

[2] Marie Charbonneau, Francesco Nori, and Daniele Pucci. On-line joint limit avoidance for torque controlled robots by joint space parametrization. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE, November 2016.

[3] H.J. Ferreau, C. Kirches, A. Potschka, H.G. Bock, and M. Diehl. qpOASES: A parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation*, 6(4):327–363, 2014.

[4] Ahmad Gazar, Gabriele Nava, Francisco Javier Andrade Chavez, and Daniele Pucci. Jerk control of floating base systems with contact-stable parameterized force feedback. *IEEE Trans. Robot.*, 37(1):1–15, February 2021.

[5] Mahmoud Hamandi, Federico Usai, Quentin Sablé, Nicolas Staub, Marco Tognon, and Antonio Franchi. Design of multirotor aerial vehicles: A taxonomy based on input allocation. *Int. J. Rob. Res.*, 40(8-9):1015–1044, August 2021.

[6] Gabriele Nava, Quentin Sable, Marco Tognon, Daniele Pucci, and Antonio Franchi. Direct force feedback control and online multi-task optimization for aerial manipulators. *IEEE Robot. Autom. Lett.*, 5(2):331–338, April 2020.

[7] Anibal Ollero, Juan Cortes, Angel Santamaria-Navarro, Miguel Angel Trujillo Soto, Ribin Balachandran, Juan Andrade-Cetto, Angel Rodriguez, Guillermo Heredia, Antonio Franchi, Gianluca Antonelli, Konstantin Kondak, Alberto Sanfeliu, Antidio Viguria, J Ramiro Martinez-de Dios, and Francesco Pierri. The AEROARMS project: Aerial robots with advanced manipulation capabilities for inspection and maintenance. *IEEE Robot. Autom. Mag.*, 25(4):12–23, December 2018.

[8] Markus Ryll, Giuseppe Muscio, Francesco Pierri, Elisabetta Cataldi, Gianluca Antonelli, Fabrizio Caccavale, Davide Bicego, and Antonio Franchi. 6D interaction control with aerial robots: The flying end-effector paradigm. *Int. J. Rob. Res.*, 38(9):1045–1062, August 2019.