init()
- controller state $\mathbf{C} = \mathbf{TRAJ}$
- number of joints J
- noise threshold ϵ
- joint times $\mathbf{t}_{i,0} = 0 \; \forall i \in J$
- joint states $\mathbf{j}_{i,0} = \text{NO\_CONTACT} \; \forall i \in J$
- forces $\mathbf{f}_{i,0}$ s $\mathbf{f}_{i,t-1} = 0 \; \forall i \in J$
- feather constant estimator $\bar{k}_{i,0} = 0 \; \forall i \in J$

update(Period p)

update_sensors();

// update joint states & timing
for i in $[0...J-1]$ **do**
    if $\mathbf{f}_{i,t-1} \leq \epsilon$ **&&** $\mathbf{f}_{i,t} > \epsilon$ **then**
        $\mathbf{s}_{i,t} = \text{GOT\_CONTACT}$
    else if $\mathbf{f}_{i,t-1} > \epsilon$ **&&** $\mathbf{f}_{i,t} \leq \epsilon$ **then**
        $\mathbf{s}_{i,t} = \text{LOST\_CONTACT}$
    else if $\mathbf{f}_{i,t-1} > \epsilon$ **&&** $\mathbf{f}_{i,t} > \epsilon$ **then**
        $\mathbf{s}_{i,t} = \text{IN\_CONTACT}$
    else
        $\mathbf{s}_{i,t} = \text{NO\_CONTACT}$
    end

    if $\mathbf{s}_{i,t} == \text{NO\_CONTACT} \,||\, \mathbf{s}_{i,t} == \text{LOST\_CONTACT}$ **then**
        $\mathbf{t}_{i,t} \leftarrow \mathbf{t}_{i,t} + p$
end

// update controller state
if $\mathbf{C} == \mathbf{TRANS}$ **then**
    $\mathbf{C} = \mathbf{F\_CTRL}$
else if $\mathbf{C} == \mathbf{TRAJ}$ **then**
    if **check_controller_transition()** **then**
        $\mathbf{C} = \mathbf{TRANS}$
        for i in $[0, J-1]$ **do**
            $\mathbf{f}_{i,T} \leftarrow \mathbf{f}_{i,t}$
            $\mathbf{j}_{i,T} \leftarrow \mathbf{j}_{i,t}$
        end

// joint position calculations
for i in $[0, J-1]$ **do**
    if $\mathbf{C} == \mathbf{TRAJ} \,||\, \mathbf{C} == \mathbf{TRANS}$ **then**
        $\mathbf{j}_{\text{des}_{i,t}} \leftarrow$ **sample_trajectory_for_joint**(Joint i, Time $\mathbf{t}_{i,t}$)
    else
        // calculate new deltas, estimate $\bar{k}$ and calculate new $\mathbf{j}_{\text{des}_{i,t}}$
        TIAGo

$$\Delta\mathbf{f}_{i,t} = \mathbf{f}_{i,t} - \mathbf{f}_{i,T}$$
$$\Delta\mathbf{j}_{i,t} = \mathbf{j}_{i,t} - \mathbf{j}_{i,T}$$

$$k_{i,t} = \frac{\mathbf{f}_{i,t}}{\mathbf{j}_{i,t}}$$
$$\bar{k}_{i,t} = (1-\lambda)k_{i,t} + \lambda\bar{k}_{i,t}$$

$$\mathbf{f}_{\text{des}_{i,t}} = 1.1 * \mathbf{f}_{\text{max}_i} - \mathbf{f}_{i,t}$$
$$\mathbf{j}_{\text{des}_{i,t}} = \frac{\mathbf{f}_{\text{des}_{i,t}}}{\bar{k}_{i,t}} + \mathbf{j}_{i,t}$$

        Shadow

$$\Delta\mathbf{j}_{i,t} = \mathbf{j}_{i,t} - \mathbf{j}_{i,T}$$

$$k_{i,t} = \frac{\mathbf{f}_{i,t}}{\Delta\mathbf{j}_{i,t}}$$
$$\bar{k}_{i,t} = (1-\lambda)k_{i,t} + \lambda\bar{k}_{i,t}$$

$$\mathbf{f}_{\text{des}_{i,t}} = \mathbf{f}_{\text{max}_i} - \mathbf{f}_{i,t}$$
$$\mathbf{j}_{\text{des}_{i,t}} = \frac{\mathbf{f}_{\text{des}_{i,t}}}{\bar{k}_{i,t}} + \mathbf{j}_{i,t}$$

    end
end

publish_debug_info();

// check if the goal is finished
if $\mathbf{C} == \mathbf{TRAJ}$ **then**
    if last Segment successfully executed **then**
        goal_succeeded();
else
    if $\mathbf{f}_{i,t} \geq \mathbf{f}_{\text{max}_i} \forall i \in [0, J-1]$ **then**
        force_finished();
        goal_succeeded();
end

// store data for next loop
for i in $[0, J-1]$ **do**
    $\mathbf{f}_{i,t-1} \leftarrow \mathbf{f}_{i,t}$
    $\mathbf{s}_{i,t-1} \leftarrow \mathbf{s}_{i,t}$
end

Algorithm 1: Platform-Independent Force Controller