[Robotics 2] Model-based collision detection and localization, addendum

September 2025

1 Transient in residual dynamics

The residual dynamics

$$r = K_o\Big(p(t) - p(0) - \int_0^t (\tau_m - \beta(q,\dot{q}) - r)dt\Big) \tag{1}$$

is discretized. Both the backwards Euler method and the Tustin/bilinear method are tested. The discretization results in a small transient in the residual dynamics. It is expected for the peaks in the transient to have smaller amplitude when using the bilinear method.

All simulations are done with $K_o = 10$ to ensure the dynamics stays within the stability limit imposed by the discretization. A feedback linearization scheme is used for control with $K_P = K_D = 200$, like in the report. The robot system is asymptotically stable and reaches steady state in about 5s. Both the control and the residual dynamics are computed in discrete time.

The simulations start from a rest configuration of $q_{0,i} = \frac{\pi}{2}$ (not an equilibrium) and perform a regulation task to $q_d = \begin{bmatrix}0 & \pi & 0 & -\frac{\pi}{2} & \pi & \pi & \pi\end{bmatrix}^T$ (not an equilibrium).

All simulations are in the nominal, collision free case.

1.1 Using approximate derivative vs factorization of $S(q,\dot{q})$

The term $\beta(q,\dot{q})$ in (1) is

$$\beta(q,\dot{q}) = n(q,\dot{q}) - \dot{M}(q)\dot{q} \tag{2}$$

Where $\dot{M}(q)$ can either be approximated as

$$\dot{M}_k(q) = \frac{M_k(q) - M_{k-1}(q)}{T_s} \tag{3}$$

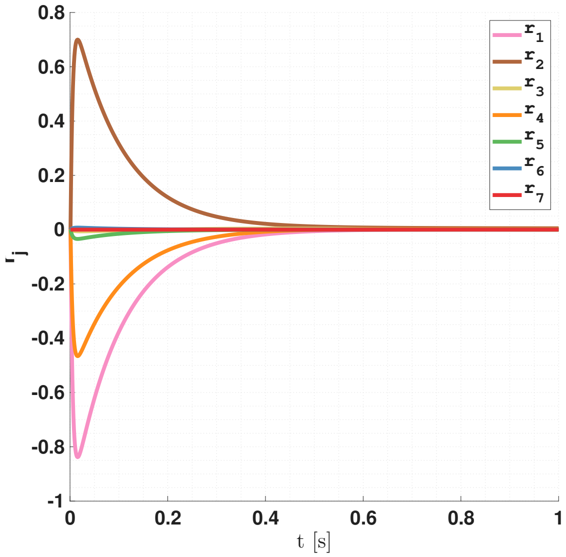
or computed exactly via a factorization of $S(q,\dot{q})$, resulting in

$$\beta(q,\dot{q}) = g(q) - S^T(q,\dot{q})\dot{q}. \tag{4}$$

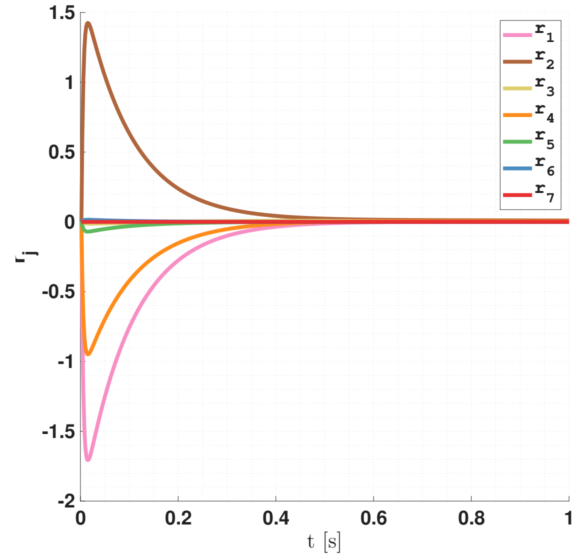
Ideally, using (4) should provide smaller transients as it is an exact computation of the term $\dot{M}(q)$.

Figures 1 and 2 show the behaviour in the two cases, with increasingly sampling times. It can be noted that in both cases, the amplitude of the peaks increases as the sampling time does, even resulting in some oscillations at $T_s = 8$s.

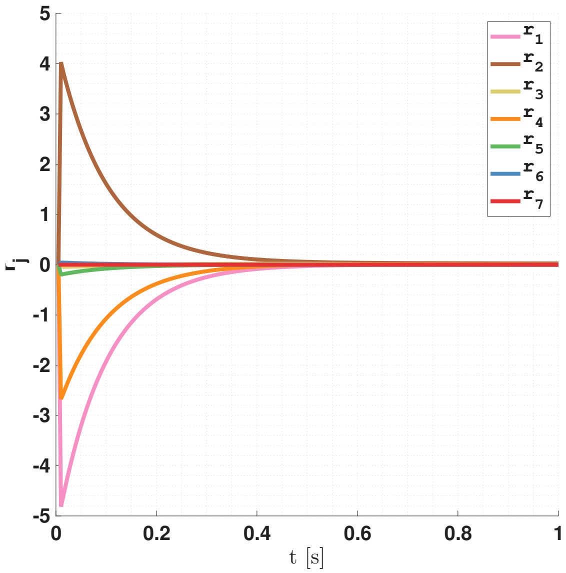
It can also be noted that there are no particular differences in transient behaviour when using (3) or (4), suggesting that (3) is a good approximation of the term $\dot{M}(q)$. The goodness of the approximation is because of the numerical conditioning of the inertia matrix: many elements much smaller than 1 and the inertia matrix is often very close to singularity. A less "singular" model would likely show more differences between using (3) and (4).



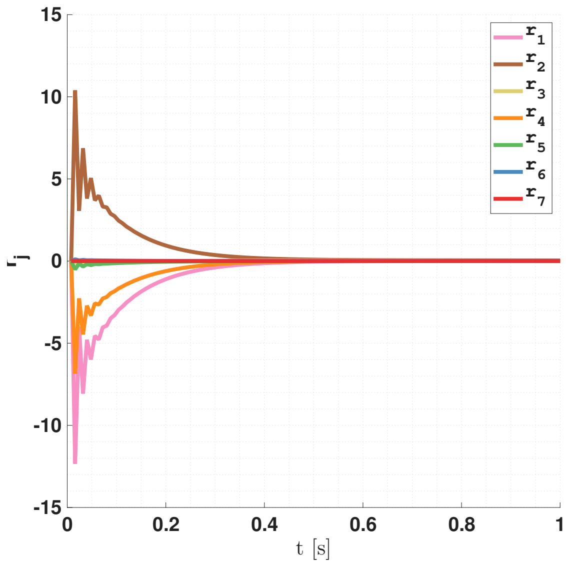
(a) T_s=1mS



(b) T_s=2mS

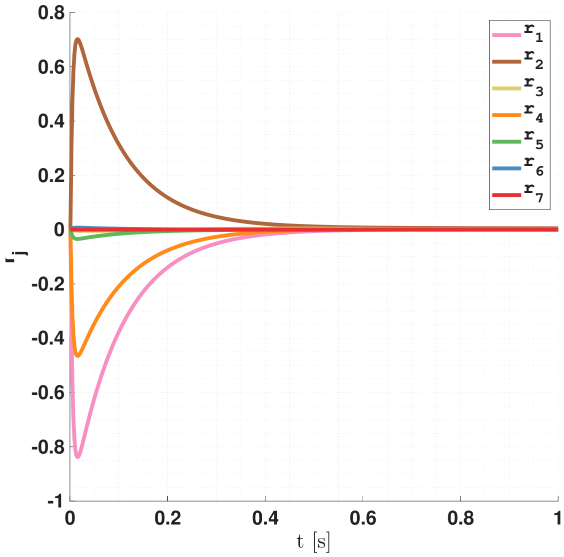


(c) T_s=5mS

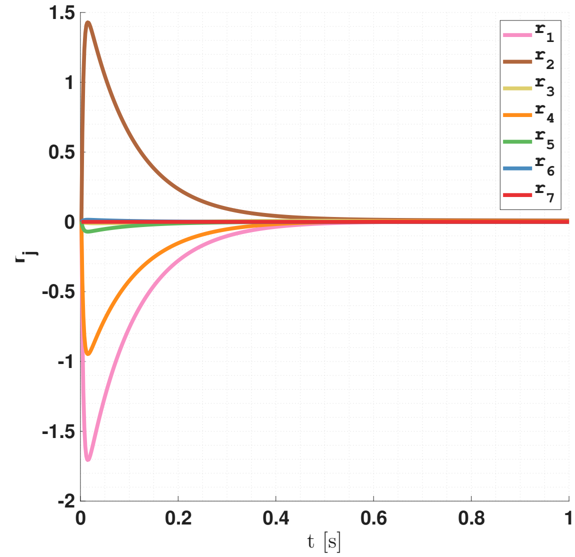


(d) T_s=8mS

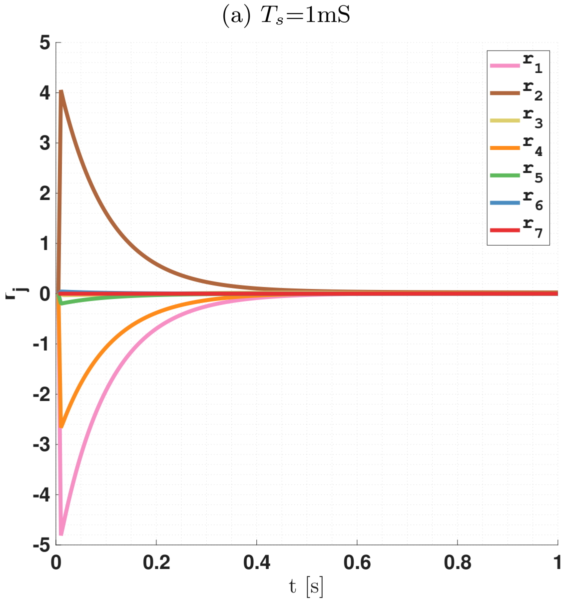
Figure 1: Residual dynamics with different sampling times, using approximate derivative (3)



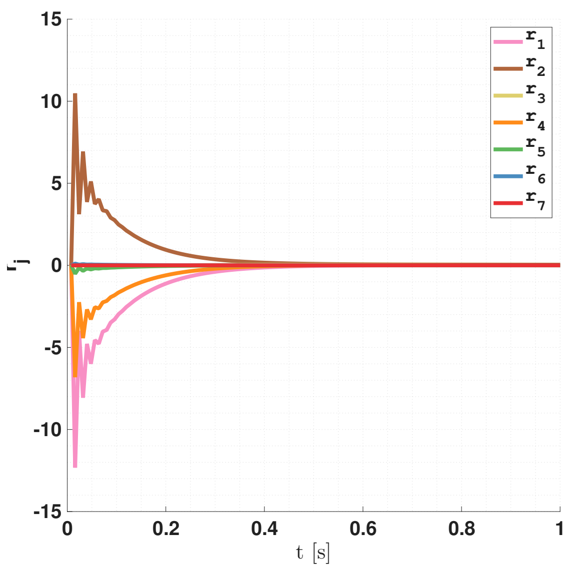
(a) T_s=1mS



(b) T_s=2mS

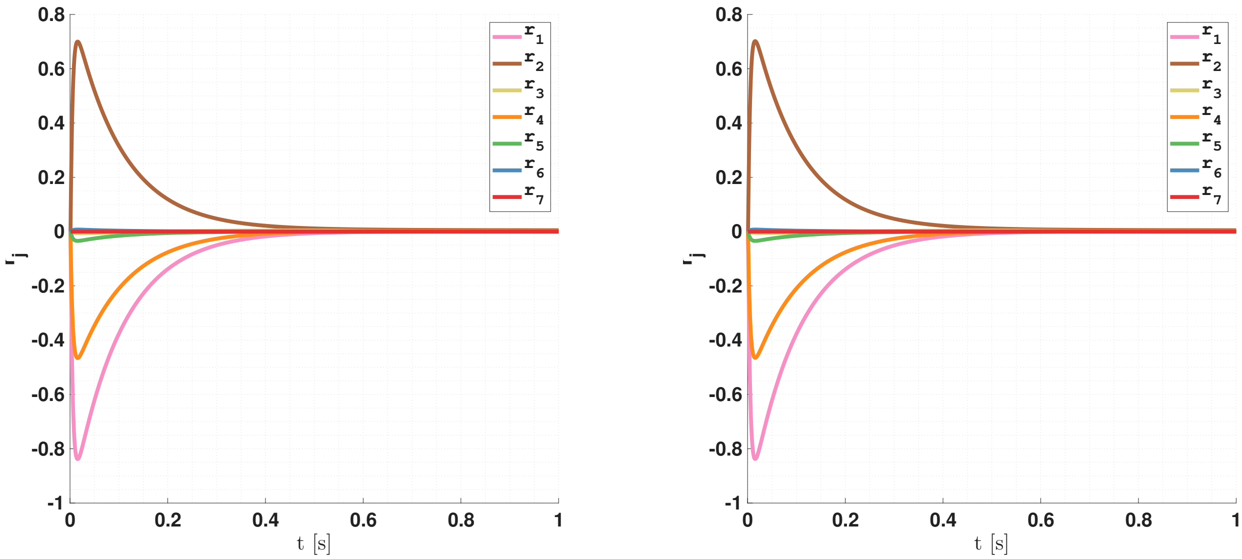


(c) T_s=5mS



(d) T_s=8mS

Figure 2: Residual dynamics with different sampling times, using $S(q, \dot{q})$ factorization (4)



(a) Using approximate derivative, T_s=1mS

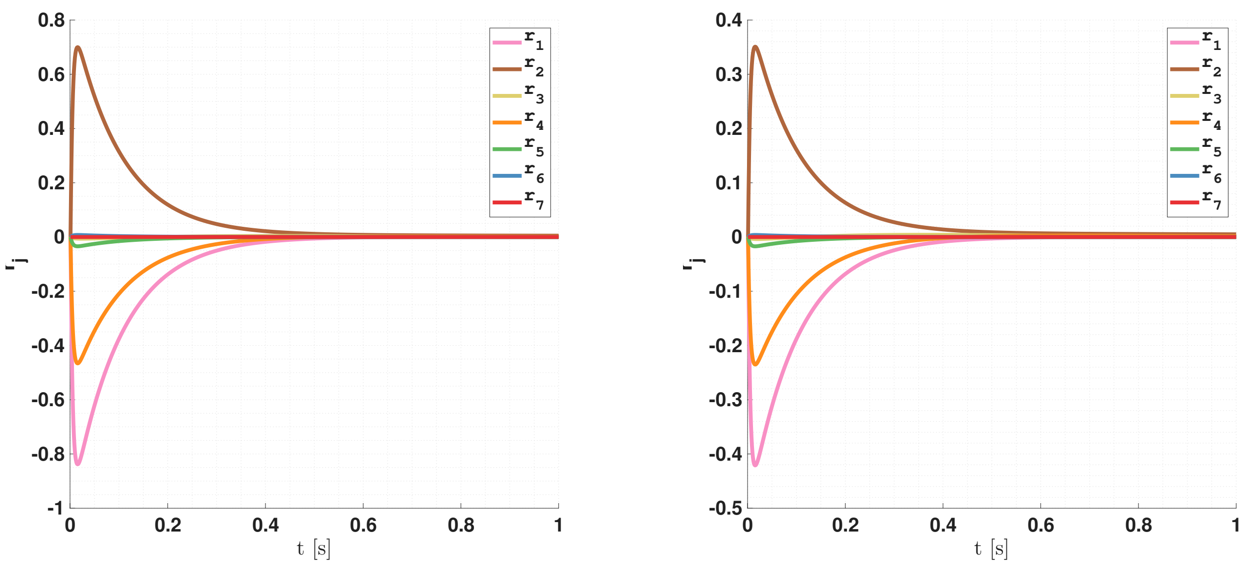
(b) Using factorization $S(q, \dot{q})$, T_s=1mS

Figure 3: Comparison between approximate derivative and $S(q, \dot{q})$. No notable differences between the two

2 Euler integration vs bilinear integration

With no particular differences between the two approaches, it is safe to deduce that the transient behaviour is due to the discretization of the integral term $\int_0^t (\tau_m - \beta(q, \dot{q}) - r)dt$

Hence, using a more precise integration method, like the Tustin/bilinear method, should result in better transient behaviour. This is in fact the case, as shown in Figure 4.



(a) Approximate derivative, T_s=1mS, backwards Euler method

(b) Approximate derivative, T_s=1mS, Tustin method

Figure 4: Comparison between backwards Euler and Tustin method

3 Sources of integration error

The two terms responsible for most of the integration error are τ_m and $\dot{q}$. When sharp and sudden changes happen in this two terms, the discrete integration methods fail to approximate the functions well, and the integration error makes the residual dynamics behave as if a collision happened.

In the report, this error is "concentrated" at the start of the simulation, as if the residual dynamics has a transient. In reality, this is due to a sudden change in τ_m and $\dot{q}$: in the regulation task, the robot starts from rest and has to move to a different joint configuration. At the start of the simulation, there is a large error between the current and the desired configuration, resulting in a high control effort, further accentuated by high gains in the controller.

For the tracking task, there is some error both in position and velocity, which again requires high control effort and a sharp change in both control effort and joint velocity to be recovered.

This is justified by the following figures. In Figure 5 a regulation tasks starts at $t = 0s$ from $q_{0,i} = \frac{\pi}{2}$ and at rest towards $q_d = \begin{bmatrix} 0 & \pi & 0 & -\frac{\pi}{2} & \pi & \pi & \pi \end{bmatrix}^T$. At $t = 5s$ however, the desired configuration changes back to q_0. The initial high control effort results in the transient-like behaviour already explored, but then the change in desired configuration at $t = 5s$ requires a sharp change in control effort (discontinuous by looking at the plots, which can indeed happen since the control is computed in discrete time) and results in a spike in the values of the residuals, as if a collision happened.

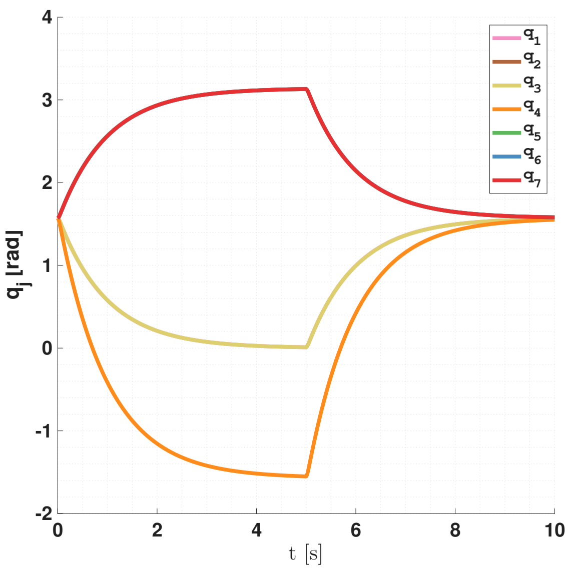
Figure 6 features a trajectory tracking task with

$$
\begin{aligned}
q_{d,i} &= \frac{\pi}{2} \cos 4t \\
\dot{q}_{d,i} &= -2\pi \sin 4t \\
\ddot{q}_{d,i} &= -8\pi \cos 4t
\end{aligned}
$$

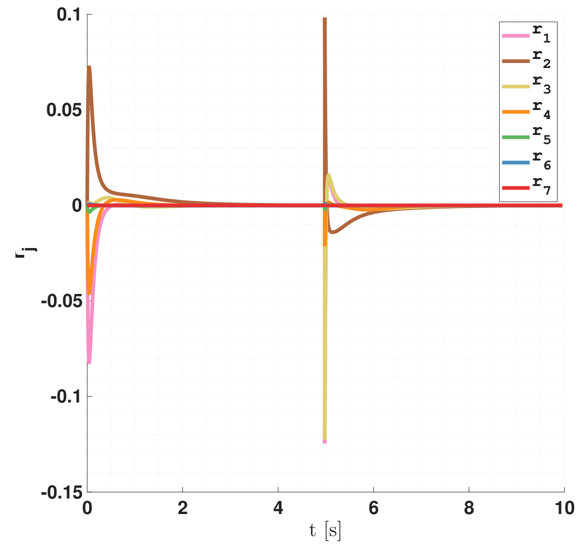
with no initial position and velocity error. The transient-like behaviour is absent here, but there is a spike in the residuals values each time the sine wave changes direction, as already featured in the report.

Finally, Figure 7 features the same regulation task as the report, now done instead by tracking a rest-to-rest trajectory described by a quintic polynomial in joint space. Here, no transient is present at all, but a spike is present when the control effort changes rapidly (eg. joints 7).

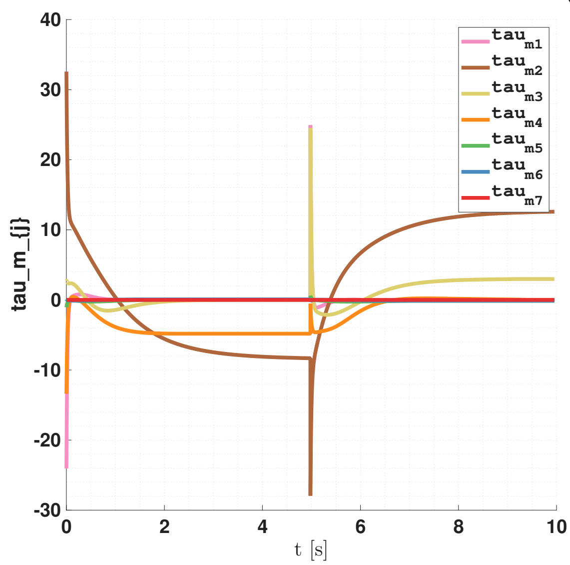
All three simulations used the feedback linearization scheme used in the report, with $K_P = K_D = 50$, the $S(q, \dot{q})$ factorization in 4 and Tustin's method for integration, with 1ms sampling time.



(a) Joints

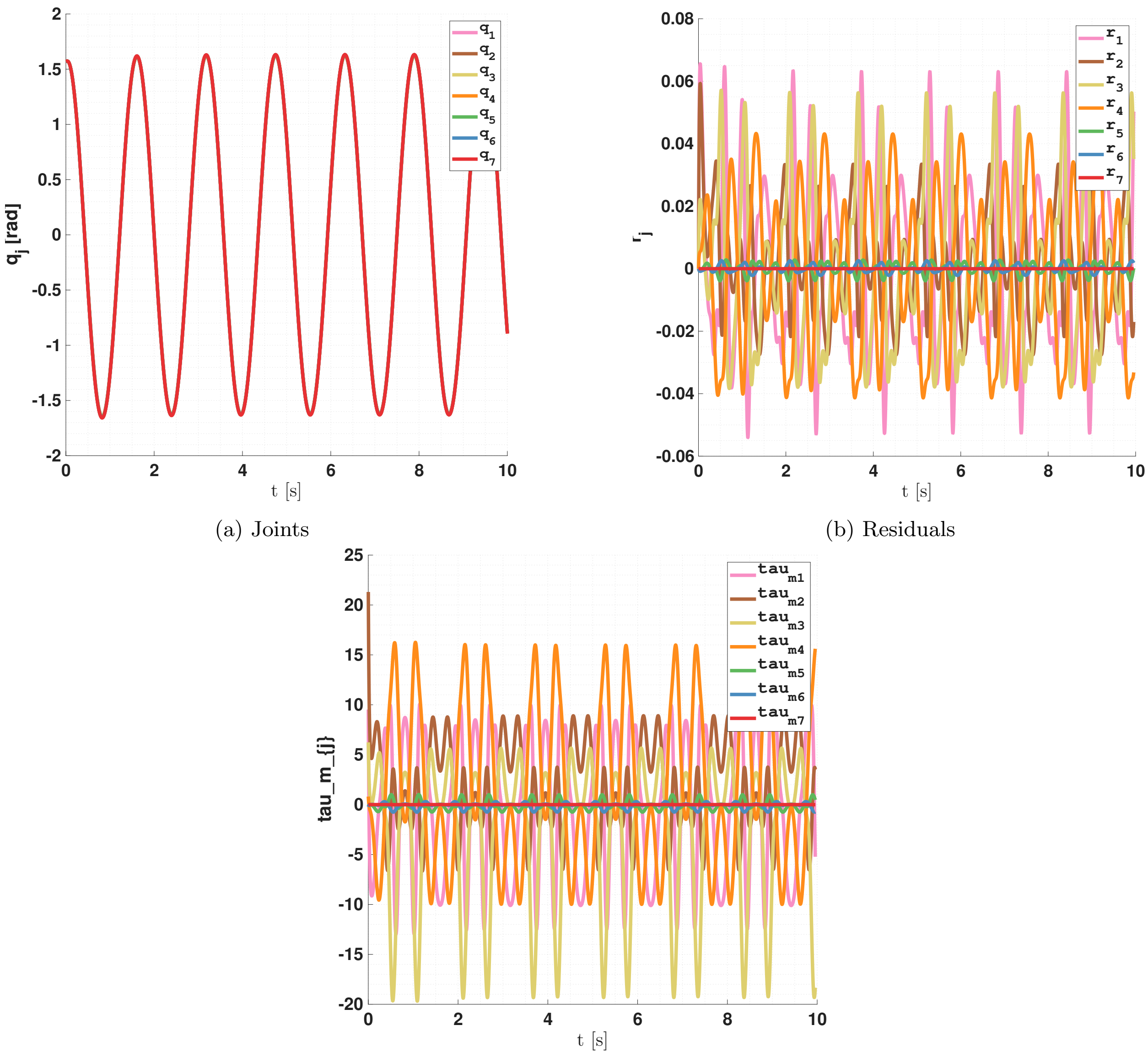


(b) Residuals



(c) Control effort

Figure 5: Double regulation task

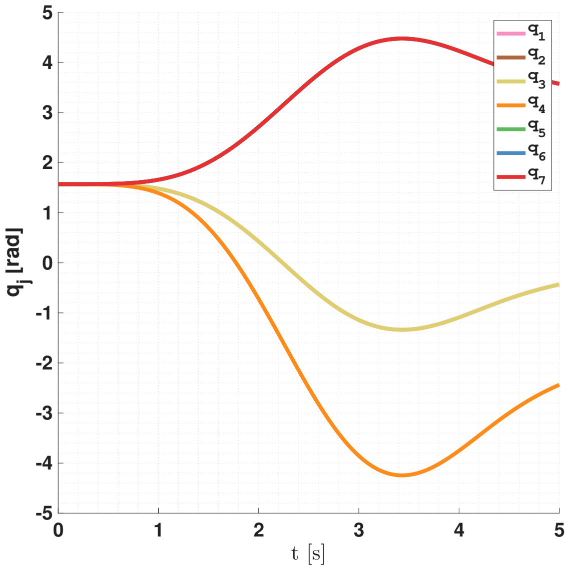


(a) Joints

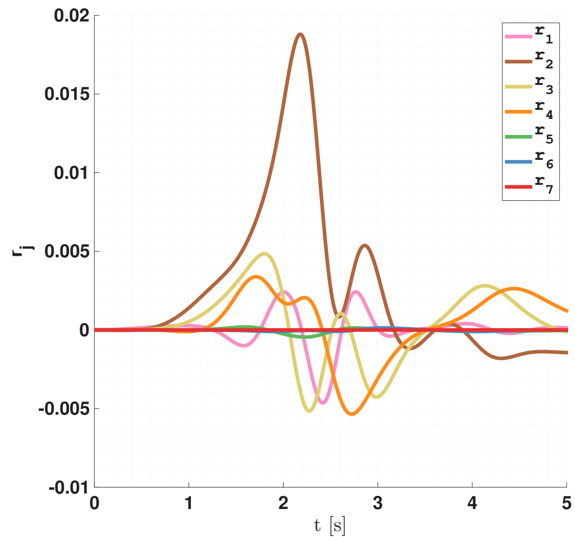
(b) Residuals

(c) Control effort

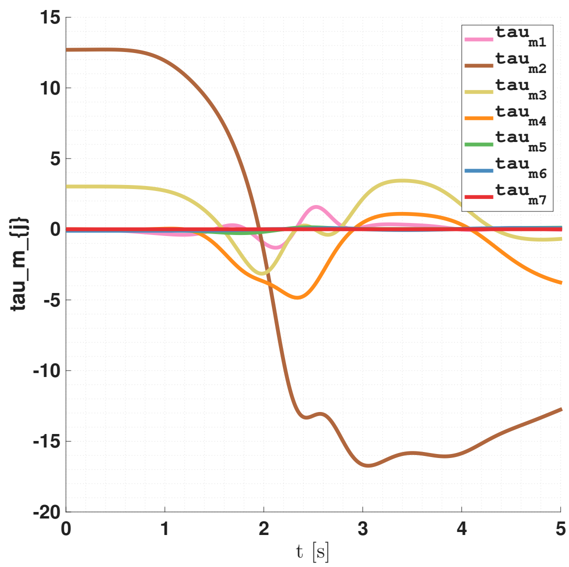
Figure 6: Trajectory tracking task with no initial error



(a) Joints



(b) Residuals



(c) Control effort

Figure 7: Regulation task by tracking a quintic polynomial