Supplement B
Comparison of Robust Input Bound

ARMOUR's robust input (given in (33)) is inspired by [4], [5]. We claim that the proposed controller improves on these published controllers because it achieves the same uniform bound with a smaller robust input bound. From [4, Thm. 1, (10)], the robust input given in previous work is

$$v(q_A(t),\Delta_0,[\Delta]) = -\Big(\kappa(t)\,\|w_M(q_A(t),\Delta_0,[\Delta])\| + \phi(t)\Big)r(t), \tag{S.19}$$

where κ and ϕ are positive increasing functions with $\kappa_P \geq 1$ and $\phi_P \geq 1$ as their respective minimums, and $w_M(q_A(t),\Delta_0,[\Delta])$ as in (31). With this choice of robust input, [4, Thm. 1] proves that the trajectories of r are ultimately uniformly bounded by $\|r(t)\| \leq \frac{1}{\kappa_P}\sqrt{\frac{\sigma_M}{\sigma_m}} \quad \forall t \geq t_1$, where t_1 is some finite time. In the context of Rem. 12, this bound holds for all time if (S.19) were used in the current framework. This uniform bound can be made identical to (35) by choosing $\kappa_P = \sqrt{\frac{\sigma_M}{2V_M}}$.

To bound the magnitude of the robust input similarly to App. E, note

$$|v(q_A(t),\Delta_0,[\Delta])|_j = \Big(\kappa(t)\,\|w_M(q_A(t),\Delta_0,[\Delta])\| + \phi(t)\Big)\,\big|r_j(t)\big|. \tag{S.20}$$

The smallest possible bound on $\big|r_j(t)\big|$ is given by $\|r(t)\| \leq \frac{1}{\kappa_P}\sqrt{\frac{\sigma_M}{\sigma_m}}$, which yields $\big|r_j(t)\big| \leq \|r(t)\| \leq \frac{1}{\kappa_P}\sqrt{\frac{\sigma_M}{\sigma_m}}$. Defining $w_M(*) = w_M(\mathbf{q_A}(\mathbf{T_i};\mathbf{K}),\Delta_0,[\Delta])$ as in (65), we have

$$|v(q_A(t),\Delta_0,[\Delta])|_j \leq \Big(\kappa(t)\,\|w_M(*)\| + \phi(t)\Big)\frac{1}{\kappa_P}\sqrt{\frac{\sigma_M}{\sigma_m}} \tag{S.21}$$

for all $t \in \mathbf{T_i}$. Next, we show that a lower bound on the right hand side of this equation (i.e., a lower bound on the robust input bound) is larger than the robust input bound (66).

Using the properties of κ and ϕ,

$$\Big(\kappa_P\,\|w_M(*)\|\Big)\frac{1}{\kappa_P}\sqrt{\frac{\sigma_M}{\sigma_m}} \leq \Big(\kappa(t)\,\|w_M(*)\| + \phi(t)\Big)\frac{1}{\kappa_P}\sqrt{\frac{\sigma_M}{\sigma_m}}. \tag{S.22}$$

Cancelling terms on the left hand side gives

$$\|w_M(*)\|\sqrt{\frac{\sigma_M}{\sigma_m}} \leq \Big(\kappa(t)\,\|w_M(*)\| + \phi(t)\Big)\frac{1}{\kappa_P}\sqrt{\frac{\sigma_M}{\sigma_m}}. \tag{S.23}$$

The robust input bound for (S.19) is larger than $\|w_M(*)\|\sqrt{\frac{\sigma_M}{\sigma_m}}$.

We compare this value to ARMOUR's robust input bound in (66). Under what conditions is (66) smaller, i.e.

$$\frac{\alpha_c\varepsilon(\sigma_M - \sigma_m) + \|w_M(*)\| + w_M(*)_j}{2} \overset{?}{\leq} \|w_M(*)\|\sqrt{\frac{\sigma_M}{\sigma_m}}. \tag{S.24}$$

This relation depends on the user-specified constants α_c and ε and σ_M and σ_m, but generally the following are true. First, $\sigma_M \geq \sigma_m$ by definition, and usually σ_M is much larger than σ_m. Second, ε is the user-specified uniform bound and a small constant is desired to minimize tracking error. Third, the j^{th} component of the worst case disturbance $w_M(*)_j$ is smaller than $\|w_M(*)\|$. Therefore, ignoring the term involving ε (which is small), ARMOUR's robust input bound (66) is smaller than (S.21) by at least a factor of $\sqrt{\frac{\sigma_M}{\sigma_m}}$.

To give an idea of the differences, consider the Kinova Gen3 robot as reported in Sec. VIII-A1: $\alpha_c = 1$, $\varepsilon = 0.049089$, $\sigma_M = 18.2726$, and $\sigma_m = 8.2993$. Plugging in yields

$$\frac{\alpha_c\varepsilon(\sigma_M - \sigma_m) + \|w_M(*)\| + w_M(*)_j}{2} \leq 0.4895 + \|w_M(*)\|, \tag{S.25}$$

and

$$\|w_M(*)\|\sqrt{\frac{\sigma_M}{\sigma_m}} = 1.48380\,\|w_M(*)\|. \tag{S.26}$$

A. Intuition Behind the Robust Input

The key intuition behind why ARMOUR's robust controller outperforms that of Giusti et al. [4] lies in how the disturbance vector $w_M(q_A(t),\Delta_0,[\Delta])$ is coupled to the modified tracking error $r(t)$. In the robust input from [4, Thm. 1, (10)], each component of the error vector is scaled by the worst-case disturbance. This means that even if uncertainty exists in only a subset of links, the robust input incorporates this worst-case disturbance at every joint, weighted by its local error. As a result, joints which actuate links that are not affected by model mismatch can still receive large robust inputs which may ultimately degrade performance. In ARMOUR's robust input (33), each component of the error vector is only scaled by the corresponding disturbance at the joint. This results in the tighter, more efficient control showing in Fig. 5.

B. Additional Comparisons Between Robust Controllers

We further compare ARMOUR's robust controller to the controller developed in [4] by including two additional experiments. First, we compare the tracking performance of both controllers under varying model uncertainties. Assuming both controllers start with the same initial tracking error, we demonstrate that both controllers converge at a similar rate (Fig. 1). Second, we compare the robust control inputs of both controllers while tracking a reference trajectory under varying levels of model uncertainty. As the model uncertainty increases, the peak control inputs required by both controllers also increase. However, ARMOUR's control input grows 2-5x more slowly than that of [4] (Fig. 2). Together, these experiments demonstrate that ARMOUR achieves comparable convergence to [4] performance with substantially less control effort.

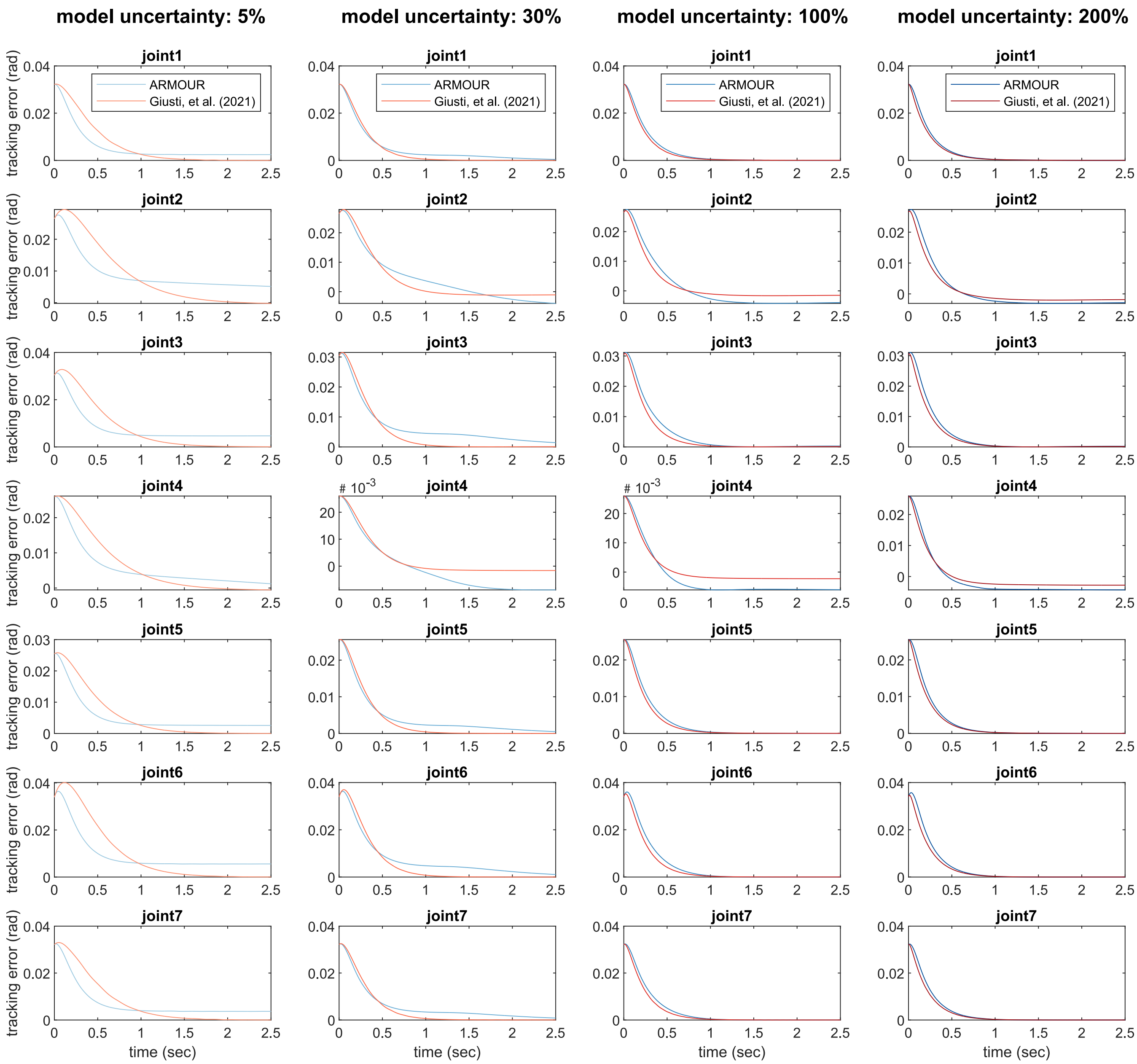

Fig. 1: This figure illustrates the tracking error of both controllers under different model uncertainties. Although both controllers start with the same initial tracking error, they converge at a similar rate, demonstrating the robustness and effectiveness of each controller.

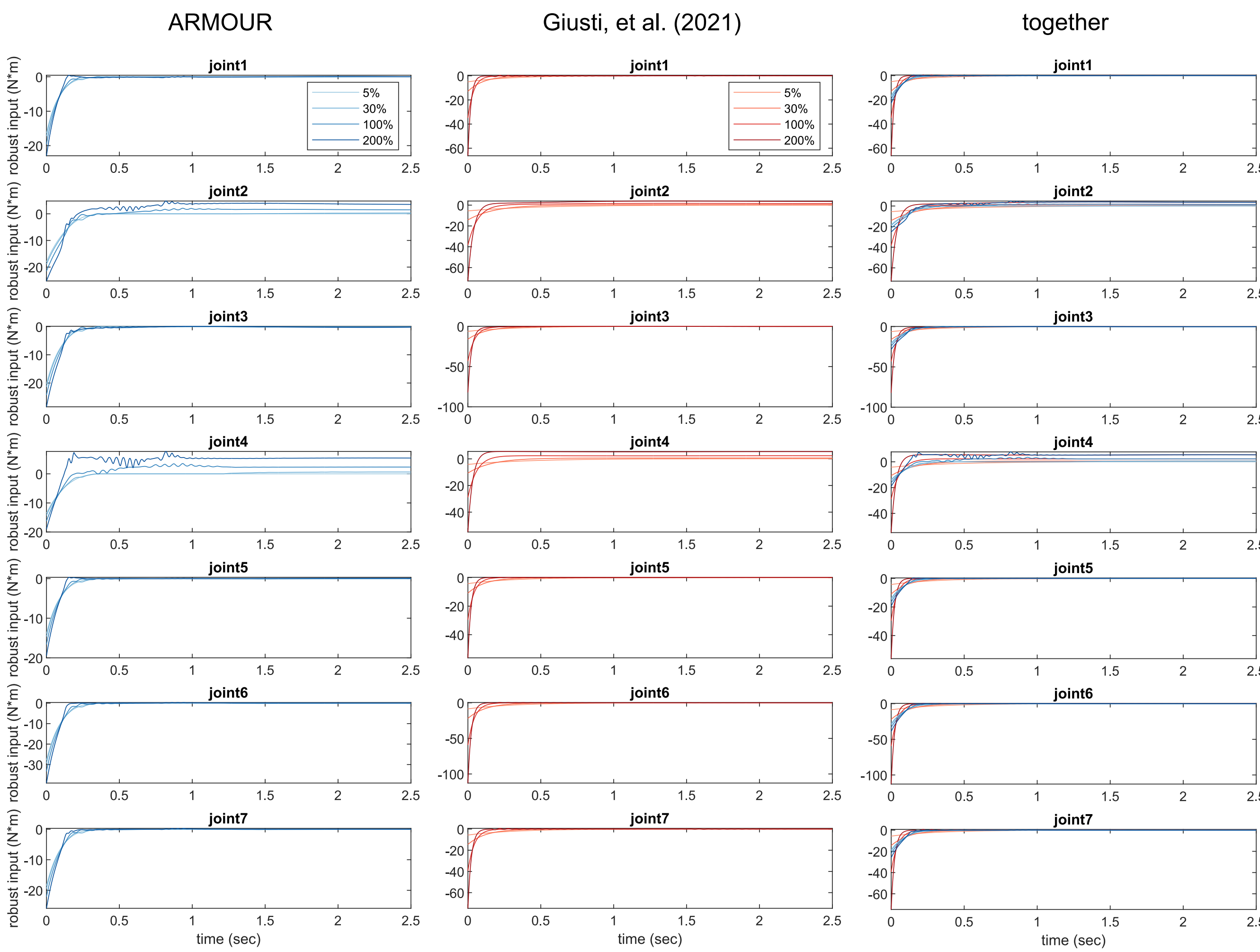

Fig. 2: This figure illustrates the robust control inputs of both controllers while tracking a reference trajectory under varying levels of model uncertainty. As the model uncertainty increases, the peak control inputs required by both controllers also increase. However, ARMOUR's control input grows 2-5x more slowly than that of [4]. This demonstrates that ARMOUR achieves comparable convergence performance with substantially less control effort.