IMPERIAL

Robust Resolved-Rate Motion Control (RRMC) for Redundant Manipulators

Guan-Sheng Chen
Supervisor: Dr Giordano Scarciotti
MathWorks: Roberto G. Valenti
George Amarantidis
Rory Adams

Back & Motivation

- **Background:** Modern robots (UR5) must achieve high precision and safety across diverse tasks and dynamic environments.
- **Challenge:** Redundant manipulators suffer severe instability (torque/velocity spikes) near kinematic singularities.
- **Engineering Aim:** Deliver a plug-and-play velocity control architecture that is robust, modular, and hardware-ready.

Core Equations

RRMC:

$$v_{des} = v_{traj} + K_p(p_{des},- p_{current})$$

Inverse Kinematics:

- Moore-Penrose (MP)

$$\dot{q}_{des} = J^{+} v_{des}$$

- Damped Least Squares (DLS)

$$\dot{q}_{des} = J^{T}(JJ^{T} + \lambda^{2} I)^{-1} v_{des}$$

MP: simple but fragile at singularities.
DLS: robust, tunable by λ.

Optimised Parameter

Parameter	Recommended Value
Inverse Kinematics	$DLS\ (\lambda = 0.03)$
RRMC K_p	$[2.2;\ 2.2;\ 2.2;\ 0.5;\ 0.5;\ 0.5]$
PID (P/I/D/N)	7.5 / 0.01 / 1.3 / 60
Rate Limiter	$\pm[0.3, 0.3, 0.3, 0.8, 0.8, 0.8]$
Velocity Sat.	$1.6\ rad/s$
Torque Sat.	$\pm 50\ Nm$

Table 1: Optimal settings for robust, hardware-friendly control

Model Limitations

- Simulation only—no real robot validation yet.
- No null-space/secondary objectives implemented.
- Friction, delay, noise are not modelled.

Sustainability

- All results: simulation only, minimal energy, no data privacy concerns.
- Open-source workflow, full reproducibility.
- EDI/Accessibility: methods directly support rehab/assistive robotics.
- Engineering process aligns with safety, legal, and ethical guidelines.

Resolved-Rate Motion Control Framework Works

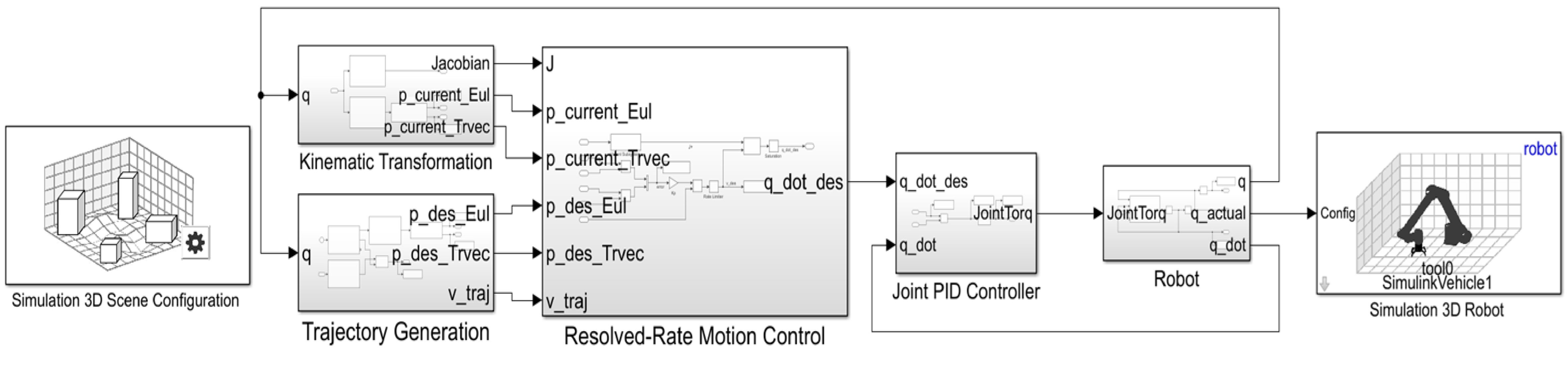

Figure 1: Overall RRMC Simulink framework: Trajectory Generation Kinematics → RRMC (MP/DLS) → Joint PID → UR5 Plant.

- **Key Modules:** All blocks are modular and support batch parameter sweeps.
- **Core Signals:** $p_{des}, p_{current}, v_{des}, \dot{q}_{des}, joint\ torque$
- **Plug-and-Play:** Safety blocks (rate limiter, saturation) ensure hardware transferability.

Key Simulation Insights

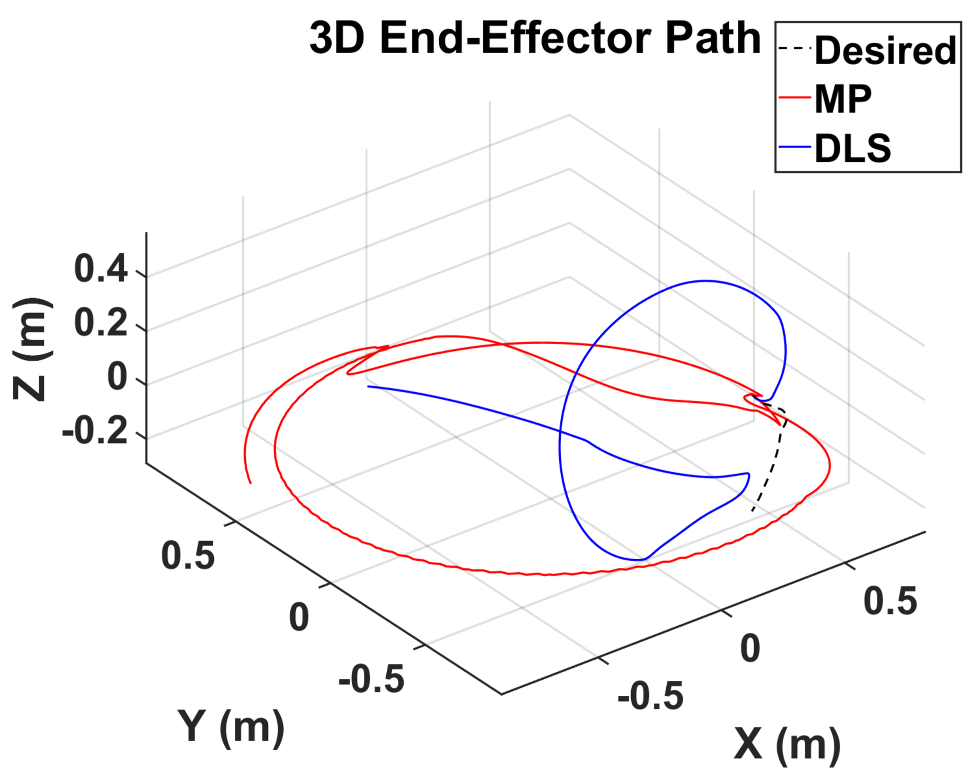

Figure 2: 3D End-Effector Path (MP V.S. DLS) with desired trajectory

- DLS closely follows the desired 3D path, while MP diverges when approaching singularities.
- Robust tracking with DLS, especially in complex, multi-axis trajectories.
- Instability and path deviation for MP.

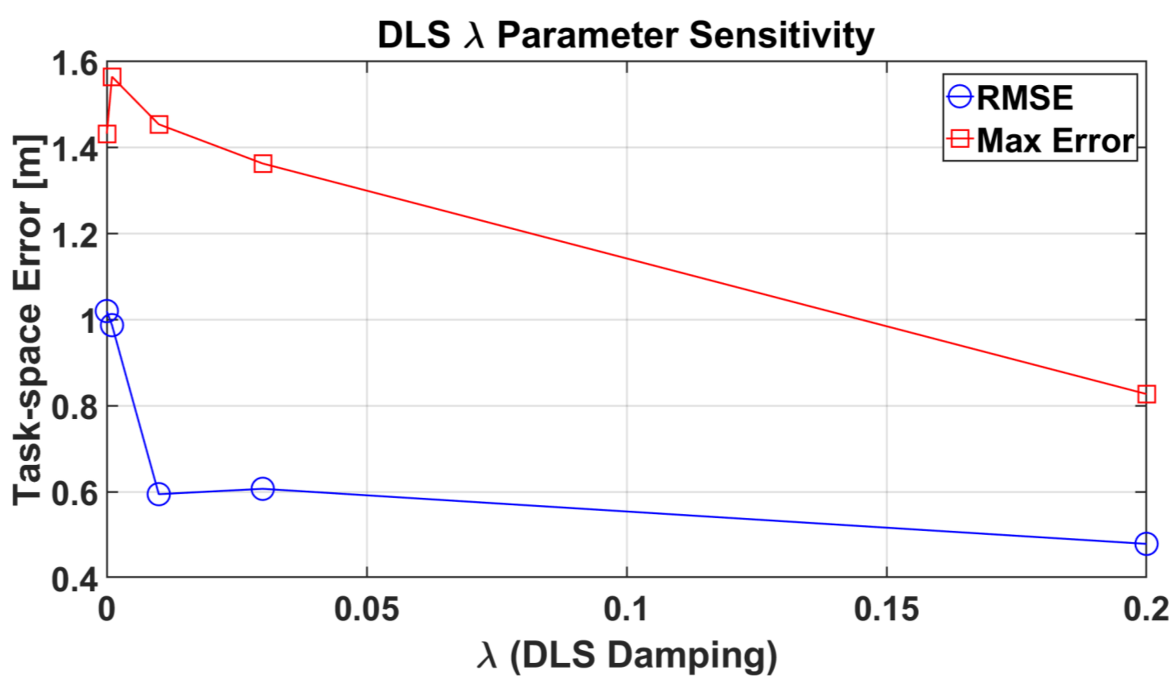

Figure 3: DLS λ Parameter Sensitivity

- Increasing λ reduces RMSE and max error (λ > 0.01 is stable).
- A moderate λ yields robust performance, while too little damping risks large errors.
- Under-regularization of λ leads to instability.
- DLS with tuned λ achieves safe, consistent tracking even near singularities.

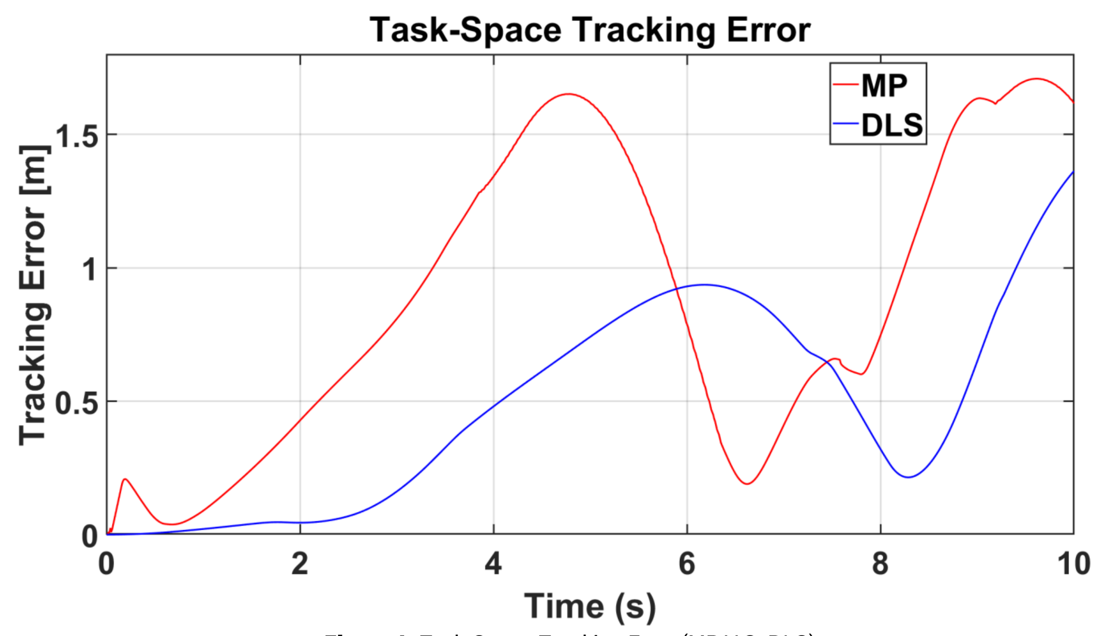

Figure 4: Task-Space Tracking Error (MP V.S. DLS)

- DLS (blue) maintains lower task-space tracking error than MP (red) throughout the trajectory.
- MP is highly sensitive to parameter choice, with error spikes in challenging sections.
- DLS is less sensitive to gain tuning and always maintains safer, more reliable error profiles.

Future Work

- Hardware tests & real robot integration.
- Null-space extension (obstacle/joint limit avoidance).
- Adaptive/robust PID.
- Modelling real-world uncertainty/friction.

Conclusion

- DLS makes velocity control practical for real robots.
- MP is only reliable in simple, slow trajectories.
- Robust tuning and safety blocks are key to safe, transferable robot control.
- Grouped K_p allocation minimizes error and actuator effort.
- Higher gains for major axes, lower for the wrist, strikes a balance between tracking and smoothness.

MathWorks®