How Two Robots Can Share a Common Workspace and Perform Online Collision Avoidance While Executing a Shared Trajectory

Floris Jousselin

To begin with, it is valuable to examine how this issue is approached in the current state of the art:
The following research works are highly relevant:

- Claes, D., Tuyls, K.
  Multi-robot collision avoidance in a shared workspace.
  https://doi.org/10.1007/s10514-018-9726-5
- Mohammad Safeea, Pedro Neto, Richard Bearee
  On-line collision avoidance for collaborative robot manipulators by adjusting off-line generated paths: An industrial use case.
  https://doi.org/10.1016/j.robot.2019.07.013
- Argtim Tika, Naim Bajcinca
  Predictive Control of Cooperative Robots Sharing a Common Workspace.
  https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10322775

From these papers, the key aspects emerge and lead me to those 5 points:

1. Shared High-Level Trajectory

A common high-level trajectory defines the global task objective. This trajectory can be:

- A spatial path (e.g., a line or surface in 3D space),
- A sequence of waypoints,
- Or a continuous task-space function (e.g., object pose over time).

Both robots interpret this shared trajectory according to their respective roles (e.g., leader/follower, mirrored agents, or equal collaborators).

2. Local Trajectory Deformation

Each robot uses a local trajectory planner to adapt the shared reference trajectory using:

- Velocity-rescaled path following,
- Elastic band deformation, or
- Model Predictive Control (MPC) with dynamic constraints.

The local planner ensures:

- Adherence to the shared objective,
- Safety through collision-free motion,
- Smooth, physically feasible trajectories.

3. Real-Time Mutual Prediction

Each robot maintains a real-time estimate of the other's pose and velocity, and makes short-term predictions (e.g., 0.5–1.0 s horizon) using:

- Monte Carlo localization models(AMCL),
- Kalman filters or learned motion models,
- Dead reckoning or joint state broadcasting.

These predictions help determine a dynamic safety margin, which adjusts based on proximity and speed.

4. Communication and Coordination

A low-latency communication system (e.g., ROS 2 DDS or shared memory) enables:

- Periodic broadcasting of intentions (e.g., short-term waypoints, motion vectors),
- Reactive synchronization,
- Role arbitration in case of conflict (e.g., "yielding" behavior or predefined priorities).

If communication fails, robots switch to conservative fallback strategies, such as halting or retracting their paths.

5. Safety via Dynamic Repulsion and Constraints

Each robot treats the other as a moving obstacle, generating a repulsive field or constraint within its motion optimization:

- Real-time enforcement of minimum safe distance constraints,
- Dynamic adaptation of these constraints based on the context (e.g., tighter bounds during low-speed collaboration),
- Real-time solvers (e.g., CHOMP, TEB, or custom MPC) handle the balance between trajectory fidelity and collision avoidance.

Key Benefits:

- Decentralized yet coordinated behavior,
- Task-level synchronization without rigid coupling,
- Robustness to communication delays or physical disturbances,
- Scalability for more than two agents.