# Increased Productivity of an Automated Tape Winding System: SPIDE-TP Platform

Divya Shah (Master Student), Jiuchun Gao (PhD Student),
Stéphane Caro (CNRS-HDR), Anatol Pashkevich (Professor),
Benoît Courtemanche (CETIM-Engineer)

## Abstract

- Recent increase in use of fibre reinforced composite materials in **aerospace, automotive** and other industries.
- Composite materials offer attractive properties like **high strength-to-weight ratio, flexibility of shaping, corrosion resistance**.

## Automated Tape Winding

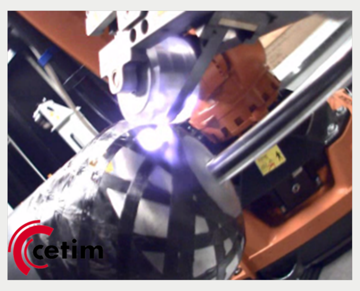


Figure: Tape winding process.

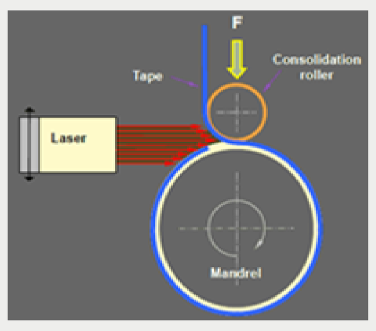



Figure: Winding head schematic.

- Automated Tape Winding (ATW) is a fabrication technique for composite components.
- Workpiece liner is mounted onto a rotating mandrel.
- Heated fibre tows are consolidated over the liner in desired paths using a compaction roller.

## SPIDE-TP Platform

- ATW system installed at `CETIM`.
- KUKA KR210 R3100 ultra robot (Payload: **210*kg***, Reach: **3095*mm***).
- KUKA KL-2000 linear axes (Range: **4500*mm***).
- 2 external AFPT winding axis (Workpiece diameters from **25*mm*** to **2500*mm*** and lengths up to **3500*mm***).
- AFPT laser-assisted tape winding head (**4*kW*** power).
- Processes all glass to carbon reinforced fibres.
- Applications: **Energy storage tanks, cryogenic tanks**, and others.

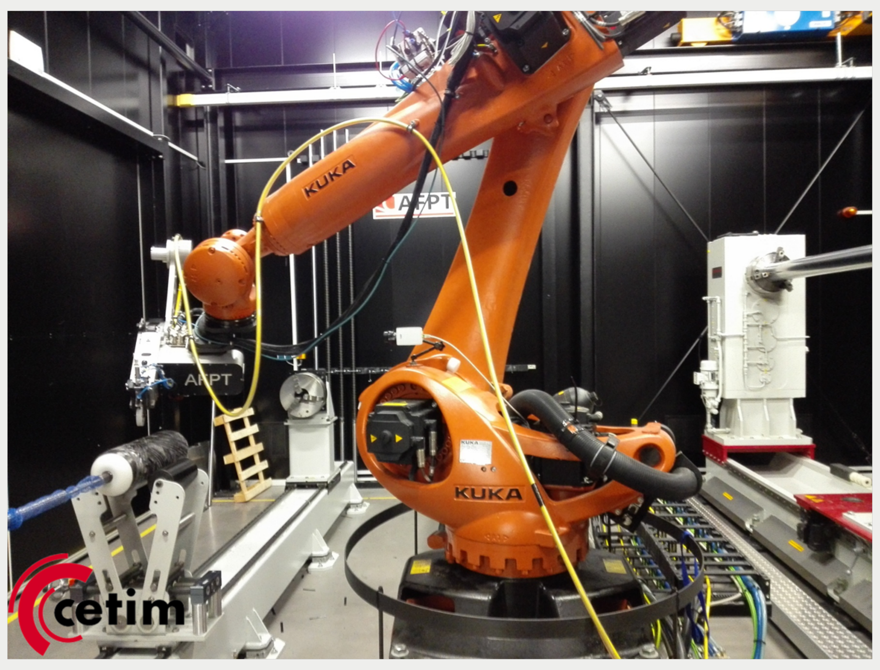


Figure: SPIDE-TP platform.

## Key Challenges

- Trajectories generated from Composicad software.
- **Kinematic redundancy** of the system not fully exploited.
- **Discontinuities** in the trajectory (near the domes).
- High raw material cost.

## Objective

To **increase the productivity** by optimizing the robot and positioner trajectories with improved management of kinematic redundancy.

## Practical Considerations

- **Additional joint limits** due to optical fibre over-bending.
- **Adjusting TCP frame** to accommodate ideal laser incidence angle.
- **Modification of shaft geometry** to avoid possible collisions.

## Further Work

- Robot programming with different motion strategies.
- Actual implementation on the SPIDE-TP platform.
- Comparison and analysis with previous methods.
- Working with more complex components & trajectories.

## System & Task Modelling

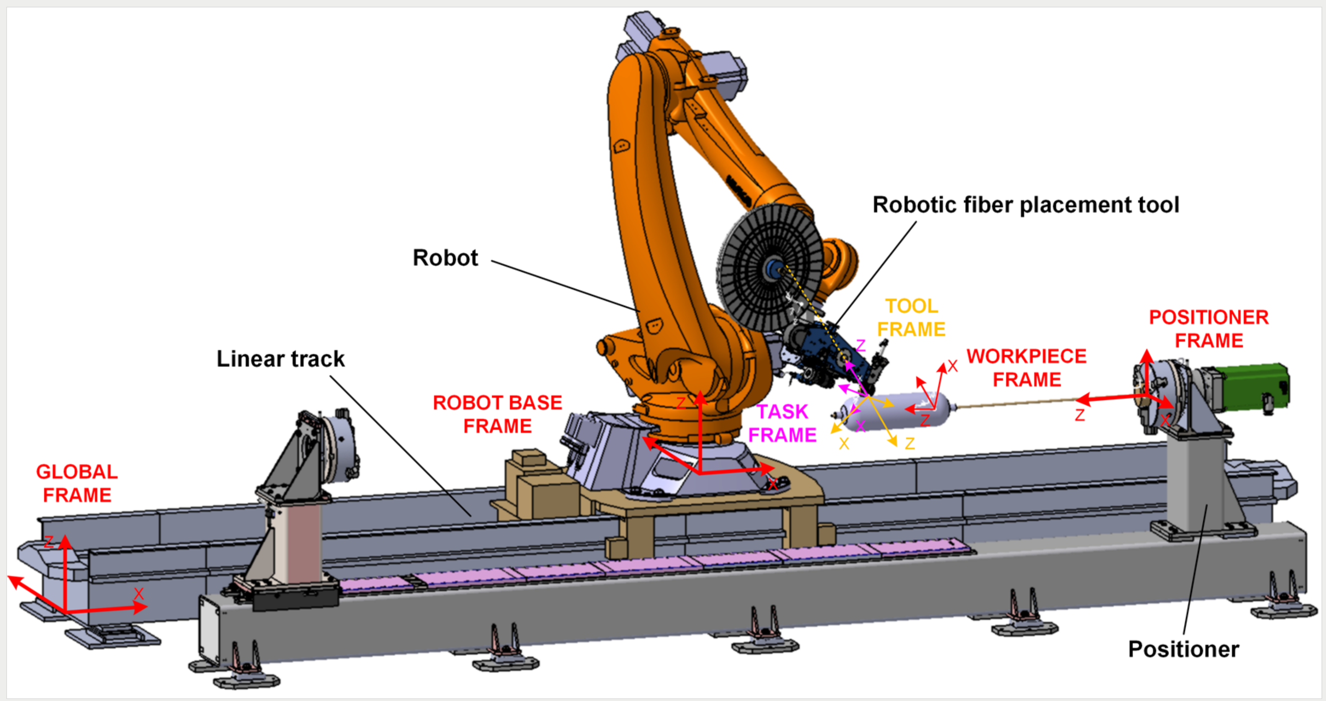



Figure: CAD model of the SPIDE-TP Platform with assigned frames.

**CAD Modelling**: 3D model of the SPIDE-TP Platform created in **CATIA V5** environment with joint-space kinematic simulation.

**Kinematic Modelling**:

Robot Model:

$$rob(\vec{q}_r) = {}^{RB}T_1(q_1).{}^{1}T_2(q_2).{}^{2}T_3(q_3).{}^{3}T_4(q_4).{}^{4}T_5(q_5).{}^{5}T_{RF}(q_6) \quad (1)$$

Positioner Model:

$$pos(q_p) = Rot_z(q_p) \quad (2)$$

Task Model:

$${}^{W}T_{TL_i} = \begin{bmatrix} \vec{n}_i & \vec{s}_i & \vec{a}_i & \vec{p}_i \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R(\vec{\varphi}_i) & \vec{p}_i \\ 0\,0\,0 & 1 \end{bmatrix} \; | i = 1, 2, ..., n \quad (3)$$

Closed loop kinematic chain:

$${}^{0}T_{RB}.rob(\vec{q}_r).{}^{RF}T_{T}ool.{}^{Tool}T_{TL_i} = {}^{0}T_{PB}.pos(q_p).{}^{PF}T_W.{}^{W}T_{TL_i} \quad (4)$$

## Optimal Trajectory Generation

### 1. Task Graph Generation

Positioner coordinate discretization:

$$q_p^k = q_p^{min} + \Delta q_p.k; \; k = 0, 1...(q_p^{max} - q_p^{min})/\Delta q_p \quad (5)$$

Computing solutions for each candidate:

$$q_r^k(t_i) = g_r^{-1}(g_p(q_p^{(k)}(t_i)), \mu); \; k = 0, .., m; \; i = 1, .., n \quad (6)$$

Location cell representing different joint configurations:

$$L_c^{(k,i)} = (q_r^{(k)}(t_i), q_p^{(k)}(t_i)) \quad (7)$$

Inter-nodal distance corresponds to the travelling time:

$$dist(L_c^{(k_i,i)}, L_c^{(k_{i+1},i+1)}) = \max_{j=0,..,6}\left(\frac{|q_{j,i}^{(k_i)} - q_{j,i+1}^{(k_{i+1})}|}{\dot{q}_j^{max}}\right) \quad (8)$$

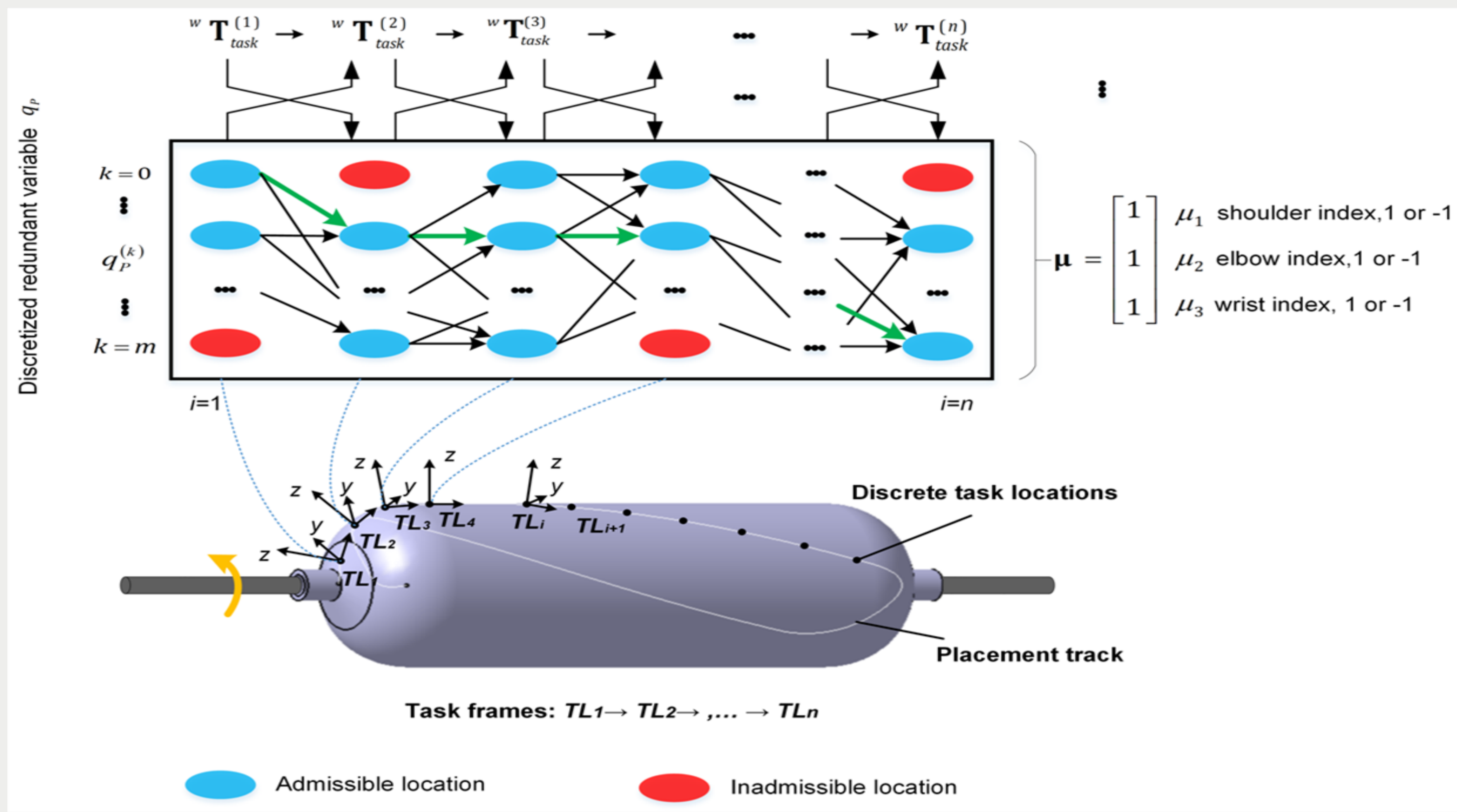
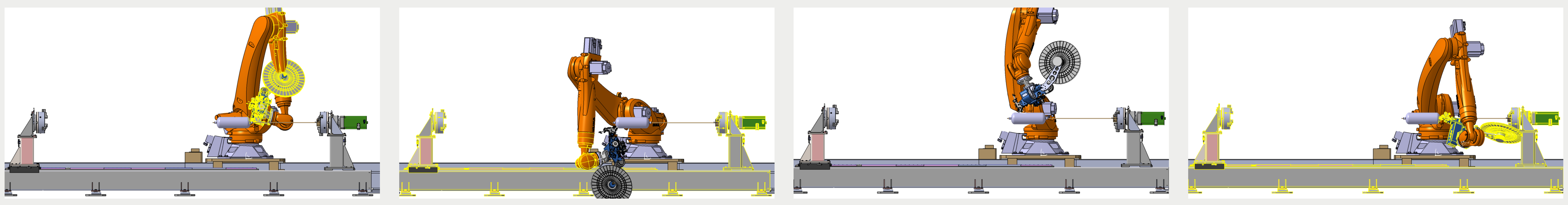



Figure: Graph based representation of discrete search space.

### 2. Collision Detection

Running the task graph on the CAD model. Checks for defined interferences between workcell components, highlights the detected collisions and returns inadmissible locations.



(a) Tool with robot. (b) Robot with positioner. (c) Collision-free candidate. (d) Tool with positioner.

Figure: Examples of collisions detected in the CAD model.

### 3. Optimal Path Planning

Objective function: total travelling time to be minimized

$$T \min_{q_r(t), q_p(t)} \; ; \; T = \sum_{j=1}^{n-1} dist(L_c^{(k_i,i)}, L_c^{(k_{i+1},i+1)}) \quad (9)$$

Path planning using the Dynamic Programming principle:

$$d_{k,i+1} = \min_{k'}\{d_{k',i} + dist(L_c^{(k,i+1)}, L_c^{(k',i)})\} \quad (10)$$

## Simulation Results & Analysis

- Determining optimal robot base location on linear axis.

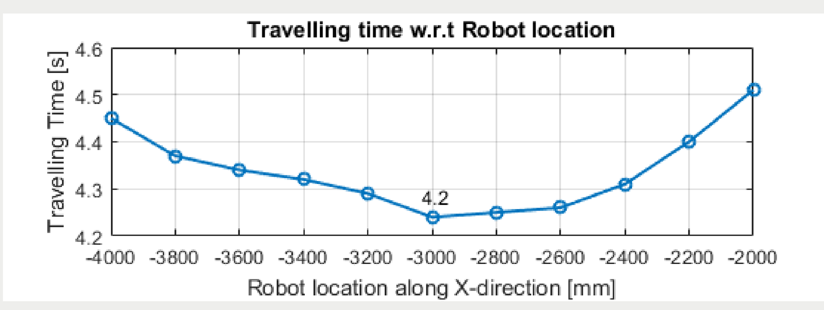



Figure: Optimal robot base location.

- For linear axis coordinate value of $-3000mm$ and with a $1deg$ discretization step for the positioner coordinate, the results obtained are as follows:

| No. of Task Locations | Travelling Time |
|---|---|
| 127 | 3.0*sec* |
| 200 | 3.2*sec* |

- Significant reduction in travelling time, compared to initial **14*sec***.
- Smooth trajectories.
- Higher quality and increased productivity.

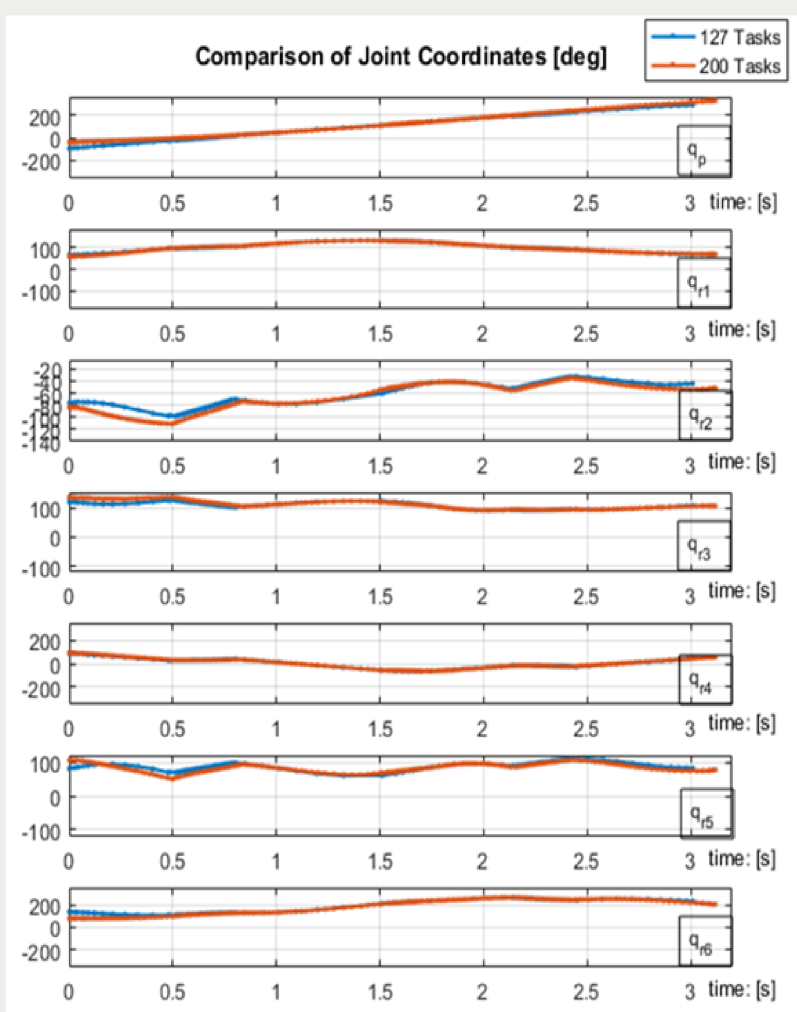



Figure: Comparison of Joint coordinates.

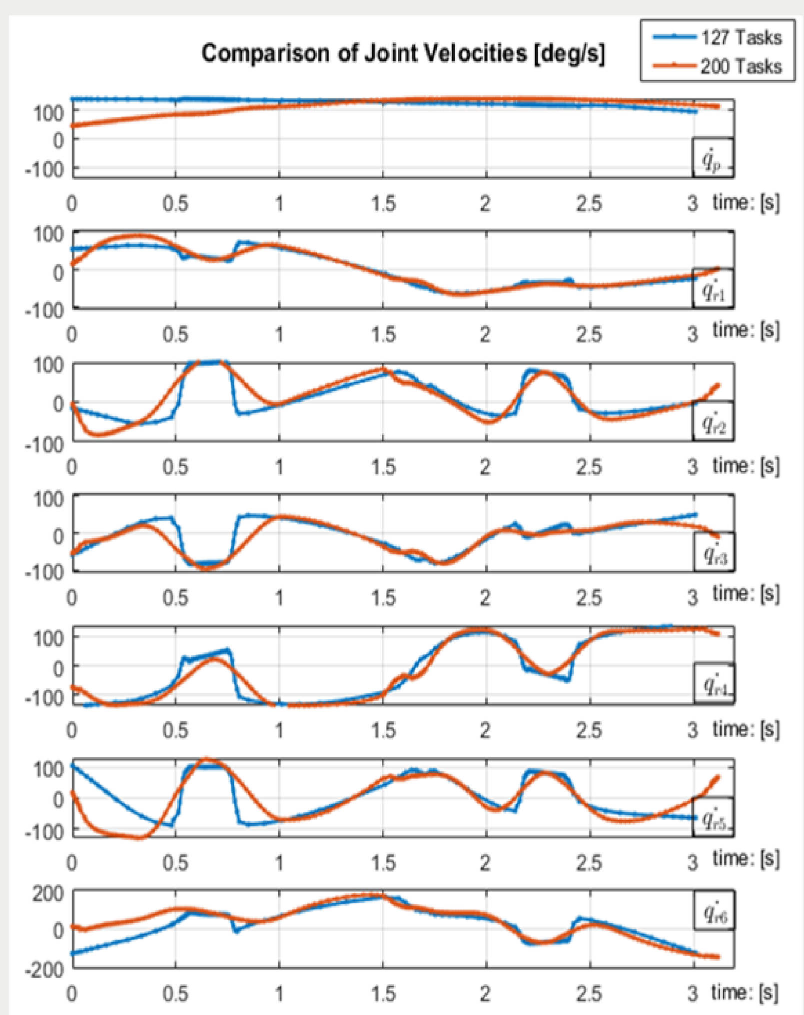



Figure: Comparison of Joint Velocities.

divya-haresh.shah@eleves.ec-nantes.fr
jiuchun.gao@ls2n.fr