MOTORSPORTS

A 2-DOF MultiBody Track Joint for Race Car Optimization

Srikanth Sivaramakrishnan, Behzad Danaei, Adwait Verulkar, Ryan Trinter
General Motors

Agenda

- Formula 1 context: why the track model matters
- Motivation
- Surface model and joint formulation
- Modelica implementation and FMI 3.0 export
- Results: time-domain validation, minimum lap time, benchmarks
- Limitations and conclusions

Why the Track Model Matters for F1 Lap-Time Optimization

MOTORSPORTS

Setup and energy decisions are made in simulation, on a model of every track.

1. **Every circuit, before the car gets there**

   Few test days and a cost cap: all 23 circuits are simulated, so each track model has to come straight from a LiDAR scan.

2. **The “perfect driver” needs a real road**

   Minimum-lap-time optimal control finds the fastest lap; camber, crown and elevation change the loads it can use.

3. **Tight corners decide lap time**

   Monaco and Singapore hairpins are tighter than the road is wide, exactly where centre-line track models break down.

4. **One model, not two**

   The production Modelica car drives the simulators; the track must plug in as a standard MultiBody component.

USE CASE 1 · SETUP PER CIRCUIT

Which car for which track?

- Ride heights and aero balance respond to bank, grade and bumps
- Lap-time sensitivities need a road that is smooth to differentiate

USE CASE 2 · ENERGY MANAGEMENT

Where to deploy, where to harvest?

- Deployment on corner exits depends on corner geometry
- Speed along the track sets the time–distance transform

Goal: optimize the production Modelica car on a measured 3D track

Vehicle + TrackJoint exported once from Dymola as FMI 3.0 Model Exchange with LS-DAE; run on all 23 circuits.

Minimum-Lap-Time Optimal Control Problem

Posed in distance s: the track supplies ṡ, the bounds on w and the road surface.

$$\min_{x(\cdot),\,z(\cdot),\,u(\cdot)} \quad t(L) \;+\; \varepsilon\, R(u)$$

lap time + small control regularization

$$\text{s.t.} \quad \frac{dx}{ds} = \frac{1}{\dot{s}(x)}\, f(x, z, u, p)$$

vehicle dynamics from the FMU, s = track distance

$$0 = g(x, z, u, p)$$

LS-DAE residuals: closed suspension loops

$$w_R(s) \le w(s) \le w_L(s)$$

track limits

$$h(x, z, u, p) \le 0$$

tyre saturation, MGU-K power & speed derate

$$0 \le \mathrm{SoC}(s) \le 1, \quad E_{\mathrm{harv}}(L) \le 1$$

battery window 4 MJ, harvest 8.5 MJ / lap

$$0 \le a(s) \le a_{\max}(s)$$

active-aero zones

$$x(0) \approx x(L)$$

periodic, flying lap

$$u_{\min} \le u \le u_{\max}, \quad p = \bar{p}$$

actuator limits; setup offsets fixed

TRANSCRIPTION (FMIOPT + CASADI)

- Direct collocation in s, IPOPT + MUMPS
- FMU Jacobians via FMI 3 directional derivatives
- LS-DAE: algebraic variables and residuals become NLP variables and constraints
- Singapore: 324 collocation points, 264 IPOPT iterations

WHERE THE TRACK ENTERS

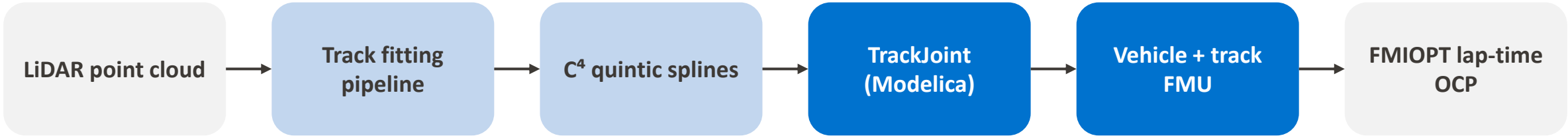
- ṡ = vsTrack, exported by TrackJoint, rescales every derivative
- Track limits w_R(s) ≤ w ≤ w_L(s) from the fitted width splines
- Height, bank and grade act through the 3D surface

One NLP per circuit, solved from a static initial guess

Track-relative chassis coordinates make it trivial: s is a linear ramp, every other chassis state is zero.

From LiDAR Scan to Lap Time

The same chain runs for every circuit on the 2026 F1 calendar; these two papers build the track part of it.



Input

- Surveyed LiDAR scan of each circuit: x, y, z and surface normals
- 23 circuits, from Monaco (3.3 km) to Spa (6.9 km)

Output for the team

- Optimal racing line, speed, aero mode and energy use from one solve
- Setup comparisons and lap-time sensitivities per circuit

Without a usable 3D track, none of this runs

The track sets the time-to-distance transform, the track limits and the road loads in every lap.

What an F1 Track Model Must Deliver

Requirement	Why it matters in Formula 1
Accurate 3D surface	Camber, crown and grade change loads and grip; Zandvoort banks up to 16.2°
Regular in tight corners	Monaco and Singapore have corners tighter than the road is wide
Smooth (C^4)	The DAE solver and the NLP differentiate the road at every collocation point
Reusable	One compiled FMU must run all 23 circuits without rebuilding
Cheap to evaluate	About two thirds of a 464 s Singapore solve is spent inside the FMU

This paper: the track as a reusable Modelica MultiBody joint

TrackJoint turns six splines into a 2-DOF joint used for both simulation and lap-time optimization.

MOTORSPORTS

Motivation

Why a Track Joint?

- Race-car simulation needs road **position, orientation and angular velocity** at every contact point
- In a MultiBody model this is naturally a **joint**: keep the chassis on the road, leave the in-plane coordinates free
  - Flat roads: Modelica.Mechanics.MultiBody.Joints.Planar
- 3D road ribbons in earlier race-car work are hand-coded inside surrogate optimization models, not reusable components
- Claytex **Rail** joint: 1 DOF along a B-spline rail
  - Orientation slaved to the rail tangent or given by 3 angle tables

Contribution

- **2-DOF** Monge-patch surface joint in (s, w)
- Orientation triad built **analytically** from surface partials
- Wrench closed by **two d'Alembert relations**, no Lagrange multipliers
- **One FMU** for time-domain simulation and CasADi/IPOPT minimum lap time

Why Not a Cartesian Road Constraint?

Constraining the chassis to z = h(x, y) works for simulation, but not for minimum lap-time optimization.

	Cartesian joint (x, y, z)	TrackJoint (s, w)
Independent variable	Time t, free final time T	Distance s ∈ [0, L], fixed
Progress along the lap	Not a state: needs a closest-point projection onto the line	s is a joint state; ṡ = vsTrack is exported
Time-to-distance transform	Not available without that projection	dt/ds = 1/ṡ directly
Track limits	Non-convex polygon constraints in (x, y)	Bounds w_R(s) ≤ w ≤ w_L(s)
Mesh	Collocation points in time, not tied to the track	Collocation points fixed along the track

$$\frac{\mathrm{d}t}{\mathrm{d}s} = \frac{1}{\dot{s}}, \qquad \dot{s} = \frac{v_t \cos\chi - v_n \sin\chi}{1 - \kappa(s)\, w}, \qquad \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}s} = \frac{\dot{\boldsymbol{x}}}{\dot{s}}$$

The time-to-distance transform needs s and ṡ in the model

TrackJoint provides both as joint states, so the same FMU is optimized in distance.

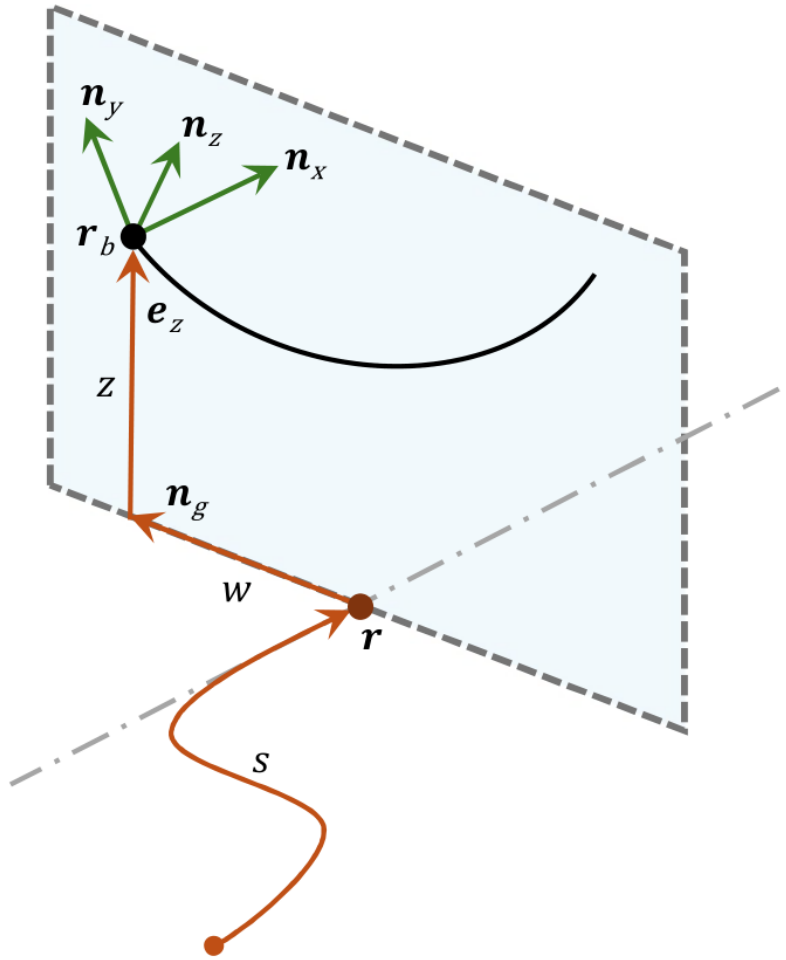
Surface Model and Joint Formulation

Surface Model: Monge Patch

$$\boldsymbol{r}_b(s,w) = \boldsymbol{r}(s) + w\,\boldsymbol{n}_g(s) + z(s,w)\,\boldsymbol{e}_z$$

$$z(s,w) = r_z(s) + z_1(s)\,w + z_2(s)\,w^2 + z_3(s)\,w^3$$

- r(s) = (r_x, r_y, 0): planar reference line; n_g(s): its ground-plane left normal
- z(s, w): lateral cubic for elevation (Limebeer and van de Merwe, 2025)
- Six channel splines **{r_x, r_y, r_z, z_1, z_2, z_3}**, periodic piecewise quintic, C^4 at knots
- Splines come from the companion fitting pipeline, anchored on the **minimum-curvature line**



Surface Tangents and Orientation Triad

$$\partial_s \boldsymbol{r}_b = \left(\partial_s r_x - w\,\partial_{ss} r_y\right)\boldsymbol{e}_x + \left(\partial_s r_y + w\,\partial_{ss} r_x\right)\boldsymbol{e}_y + \partial_s z\,\boldsymbol{e}_z$$
$$\partial_w \boldsymbol{r}_b = -\partial_s r_y\,\boldsymbol{e}_x + \partial_s r_x\,\boldsymbol{e}_y + \partial_w z\,\boldsymbol{e}_z$$

$$\boldsymbol{n}_z = \frac{\partial_s \boldsymbol{r}_b \times \partial_w \boldsymbol{r}_b}{\|\partial_s \boldsymbol{r}_b \times \partial_w \boldsymbol{r}_b\|}, \qquad \boldsymbol{n}_x = \frac{\partial_s \boldsymbol{r}_b}{\|\partial_s \boldsymbol{r}_b\|}, \qquad \boldsymbol{n}_y = \boldsymbol{n}_z \times \boldsymbol{n}_x$$

- Joint states: **(s, ṡ, w, ẇ)**; everything else is evaluated from the splines at the current s
- Right-handed surface triad (n_x, n_y, n_z): n_x follows the reference line, n_z is the surface normal
- Lateral offset is along the ground-plane normal, so no yaw spline and **no Euler-angle or quaternion integration**

Frame Outputs and Parametric Rotation Rates

$$\mathbf{R}_b = [\, \boldsymbol{n}_x \;\; \boldsymbol{n}_y \;\; \boldsymbol{n}_z \,]^\top$$

$$\boldsymbol{\omega}_b = \mathbf{R}_b \left(\partial_s \boldsymbol{\phi}\, \dot{s} + \partial_w \boldsymbol{\phi}\, \dot{w} \right)$$

$$\partial_s \boldsymbol{\phi} = \underbrace{\frac{\boldsymbol{n}_z \times \partial_s \boldsymbol{c}}{\|\boldsymbol{c}\|}}_{\text{roll / pitch}} + \underbrace{\frac{(\partial_{ss} \boldsymbol{r}_b) \cdot \boldsymbol{n}_y}{\|\partial_s \boldsymbol{r}_b\|}\, \boldsymbol{n}_z}_{\text{yaw}}$$

$$\partial_w \boldsymbol{\phi} = \frac{\boldsymbol{n}_z \times \partial_w \boldsymbol{c}}{\|\boldsymbol{c}\|} + \frac{(\partial_{sw} \boldsymbol{r}_b) \cdot \boldsymbol{n}_y}{\|\partial_s \boldsymbol{r}_b\|}\, \boldsymbol{n}_z, \qquad \boldsymbol{c} = \partial_s \boldsymbol{r}_b \times \partial_w \boldsymbol{r}_b$$

- Triad is orthonormal for every (s, w), so its parametric derivative is skew-symmetric and defines a unique rate vector ∂ξϕ
- Built from pieces the surface partials already provide: **cheaper and better conditioned** than ½Σ n_i × ∂n_i
- In-plane part (roll/pitch) from the normal; yaw part from the tangent column
- Needs spline derivatives up to the **third s-derivative** of r_x and r_y

D'Alembert Constraint

$$0 = \boldsymbol{f}_b^w \cdot \partial_s \boldsymbol{r}_b + \boldsymbol{\tau}_b^w \cdot \partial_s \boldsymbol{\phi}$$
$$0 = \boldsymbol{f}_b^w \cdot \partial_w \boldsymbol{r}_b + \boldsymbol{\tau}_b^w \cdot \partial_w \boldsymbol{\phi}$$

$$\mathbf{A} = \begin{bmatrix} (\partial_s \boldsymbol{r}_b)^\top & (\partial_s \boldsymbol{\phi})^\top \\ (\partial_w \boldsymbol{r}_b)^\top & (\partial_w \boldsymbol{\phi})^\top \end{bmatrix}, \qquad \operatorname{rank} \mathbf{A} = 2 \iff \|\partial_s \boldsymbol{r}_b \times \partial_w \boldsymbol{r}_b\| \neq 0$$

- Generalized forces conjugate to (s, w) are set to zero (virtual work); wrench resolved to world with resolve1
- Two algebraic equations close the **index-1 DAE** in (ṡ, ẇ): no Lagrange multipliers or auxiliary variables
- Solvability depends **only on the surface geometry**: A has full rank iff the surface tangents are independent

Regularity: Where the Joint Breaks Down

$$\|\partial_s \boldsymbol{r}_b \times \partial_w \boldsymbol{r}_b\| \approx \left|1 - \kappa(s)\, w\right| \sqrt{1 + \|\nabla z\|^2}$$

$$w_{\text{fold}}(s) = \frac{1}{\kappa(s)}$$

$$\min_s \frac{1}{|\kappa(s)|\, b(s)} > 1$$

Continuity required of the splines

- Rotation rates need third s-derivatives; index reduction differentiates once more → 4th derivative
- **C^4 piecewise quintic** is the lowest degree that stays continuous across knots; cubic splines make the integrator chatter at segment boundaries

- Patch folds at the **centre of curvature** of the reference line
- With the geometric centre-line, slow corners with radius below the half-width put the fold **inside the drivable strip**
  - Two points for the same (s, w); A loses rank; DAE cannot be solved
- Anchoring on the **minimum-curvature line** lowers curvature peaks and moves the fold outside the strip
- Margin checked for every circuit used (all 23 > 1)

Time-to-Space Transformation

$$\frac{\mathrm{d}t}{\mathrm{d}s} = \frac{1}{\dot{s}}, \qquad \dot{s} = \frac{v_t \cos\chi - v_n \sin\chi}{1 - \kappa(s)\, w}, \qquad \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}s} = \frac{\dot{\boldsymbol{x}}}{\dot{s}}$$

- Minimum lap time is posed with arc length s as the independent variable: fixed domain [0, L], cost T = ∫ dt
- ṡ is the **single quantity** through which the road geometry enters the transformation
- Joint exports **vsTrack = der(sTrack)** with StateSelect.always, so it survives as a continuous FMU output
- FMIOPT reads vsTrack from the FMI 3.0 interface and rescales every state derivative, **without modifying the model**
- The denominator 1 − κw is the same factor that sets the rank of the constraint Jacobian A

MOTORSPORTS

Modelica Implementation and FMI Export

Equation Block of TrackJoint

```
// Height z(s,w) in Horner form
z    = rz + wTrack*(z1 + wTrack*(z2 + wTrack*z3));
dzdw = z1 + wTrack*(2*z2 + wTrack*(3*z3));
// Surface position and partial derivatives
rTrack = {rx - wTrack*dryds, ry + wTrack*drxds, z};
drds   = {drxds - wTrack*d2ryds2, dryds + wTrack*d2rxds2, dzds};
drdw   = {-dryds, drxds, dzdw};
// Track triad: tangent, normal, binormal
nxTrack = normalize(drds);
nzTrack = normalize(cross(drds, drdw));
nyTrack = cross(nzTrack, nxTrack);
// Parametric angular-rate vectors
cTrack = cross(drds, drdw);
dphids = cross(nzTrack, dcTrackds)/length(cTrack)
       + ((d2rds2*nyTrack)/length(drds))*nzTrack;
dphidw = cross(nzTrack, dcTrackdw)/length(cTrack)
       + ((d2rdsdw*nyTrack)/length(drds))*nzTrack;
derPhiTrack = dphids*vsTrack + dphidw*vwTrack;
// D'Alembert: virtual work in s and w directions
0 = fFrameB*drds + tFrameB*dphids;
0 = fFrameB*drdw + tFrameB*dphidw;
// Frame_b: position, rotation matrix, angular velocity
frame_b.r_0 = rTrack;
frame_b.R.T = transpose([nxTrack, nyTrack, nzTrack]);
frame_b.R.w = frame_b.R.T*derPhiTrack;
```

- **Ground-plane lateral direction**: orientation follows from the spline 1st/2nd derivatives
- **Horner evaluation**: backward stable, keeps relative error small in the high-order derivatives
- **D'Alembert**: two projections of the resolved wrench close the 2-DOF DAE
- **C^4 requirement**: the solver differentiates the angular-velocity equation during index reduction

Spline Storage, MultiBody Assembly and FMI 3.0 Export

MOTORSPORTS

Spline storage and evaluation

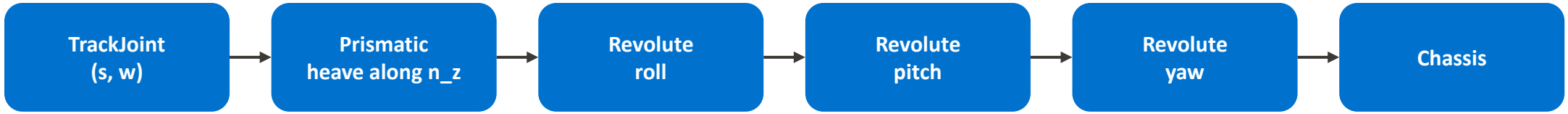
- One parameter record: six coefficient matrices N_seg × 6 and one shared break vector
- Uniform breaks: **constant-time segment lookup**; one Horner function with derivative order q ∈ {0, ..., 3}

$$y^{(q)}(s) = \sum_{j=q}^{d} \frac{j!}{(j-q)!}\, a_{\mathrm{ix},j}\, (s - s_{\mathrm{ix}})^{\,j-q}, \qquad \mathrm{ix} = \min\left(\left\lfloor \frac{s \bmod L}{\Delta s} \right\rfloor,\ N_{\mathrm{seg}} - 1\right)$$

MultiBody and FMI

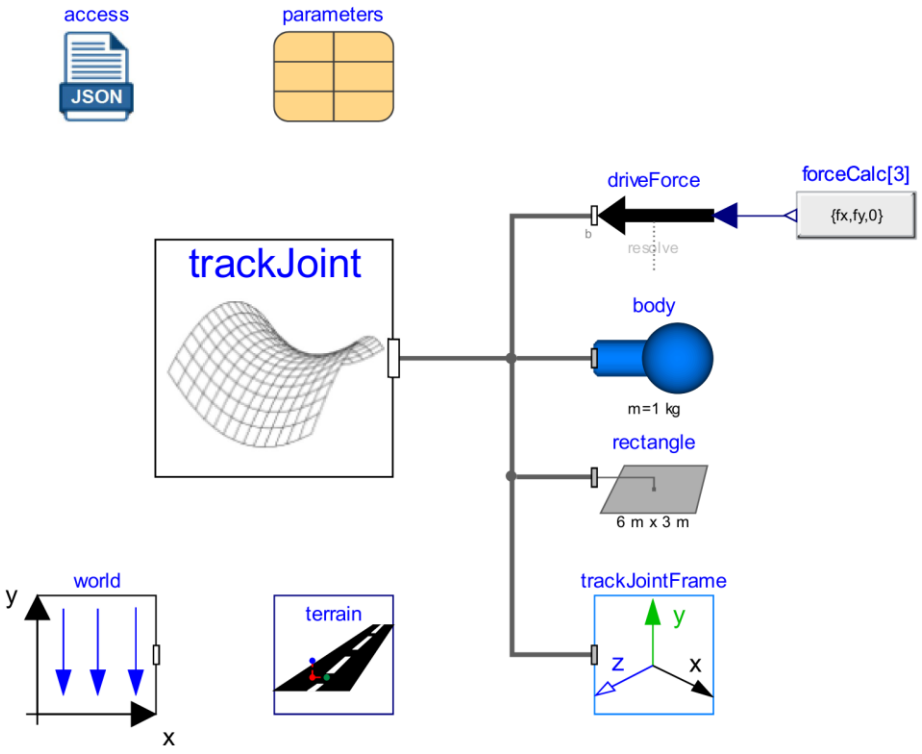
- (s, ṡ, w, ẇ) declared StateSelect.always; Connections.root(frame_b.R)
- Chassis keeps 6 DOF, each **relative to the track triad**
- Vehicle + joint exported as **one FMI 3.0 model-exchange FMU**
- Spline coefficients are **tunable**: one binary, any circuit, no recompilation

Vehicle assembly on frame_b

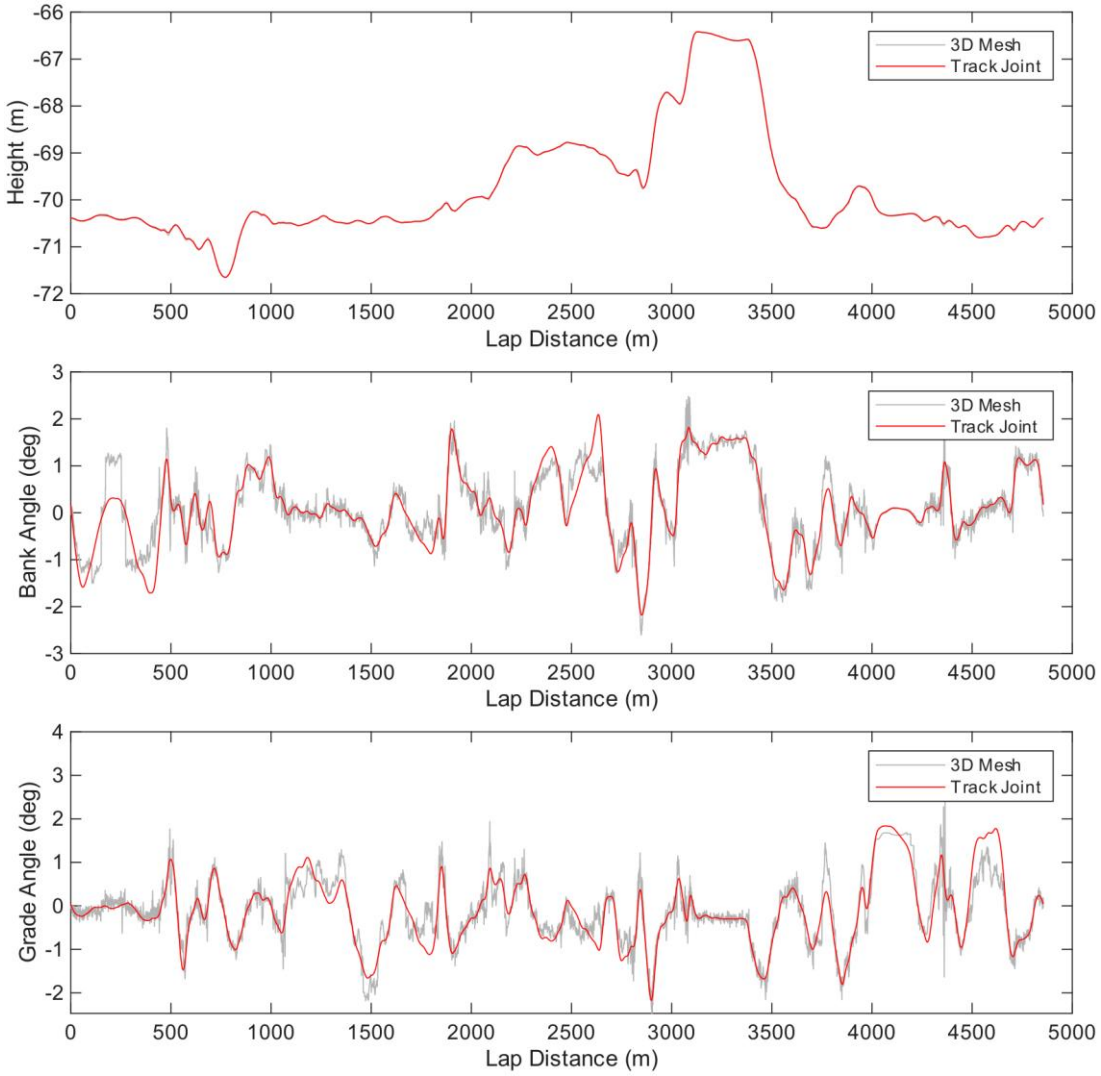


Results

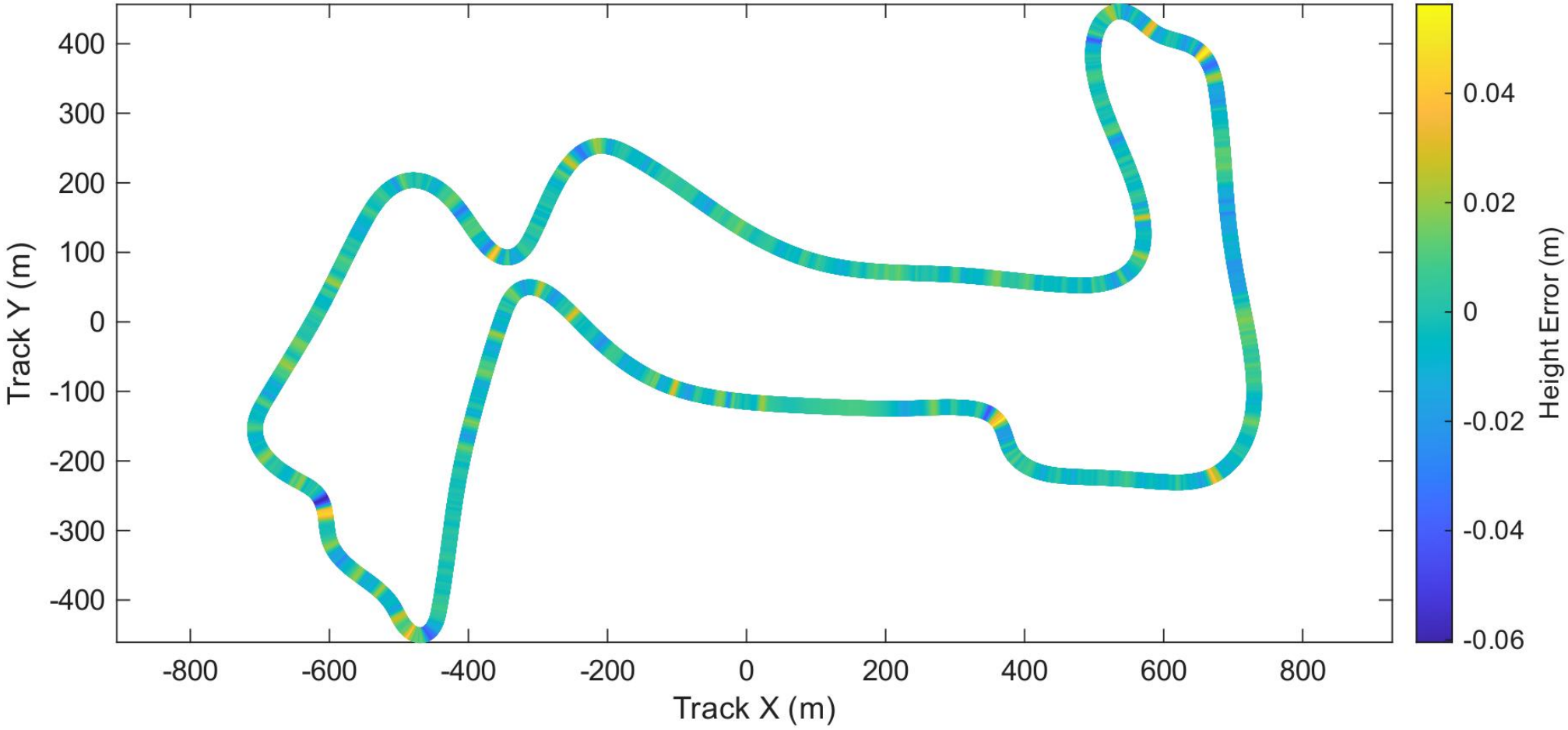
Time-Domain Validation: Singapore



- Point mass driven along the minimum-curvature line with the joint as the only constraint
- Surface compared with the rFPro Terrain Server mesh of the LiDAR point cloud

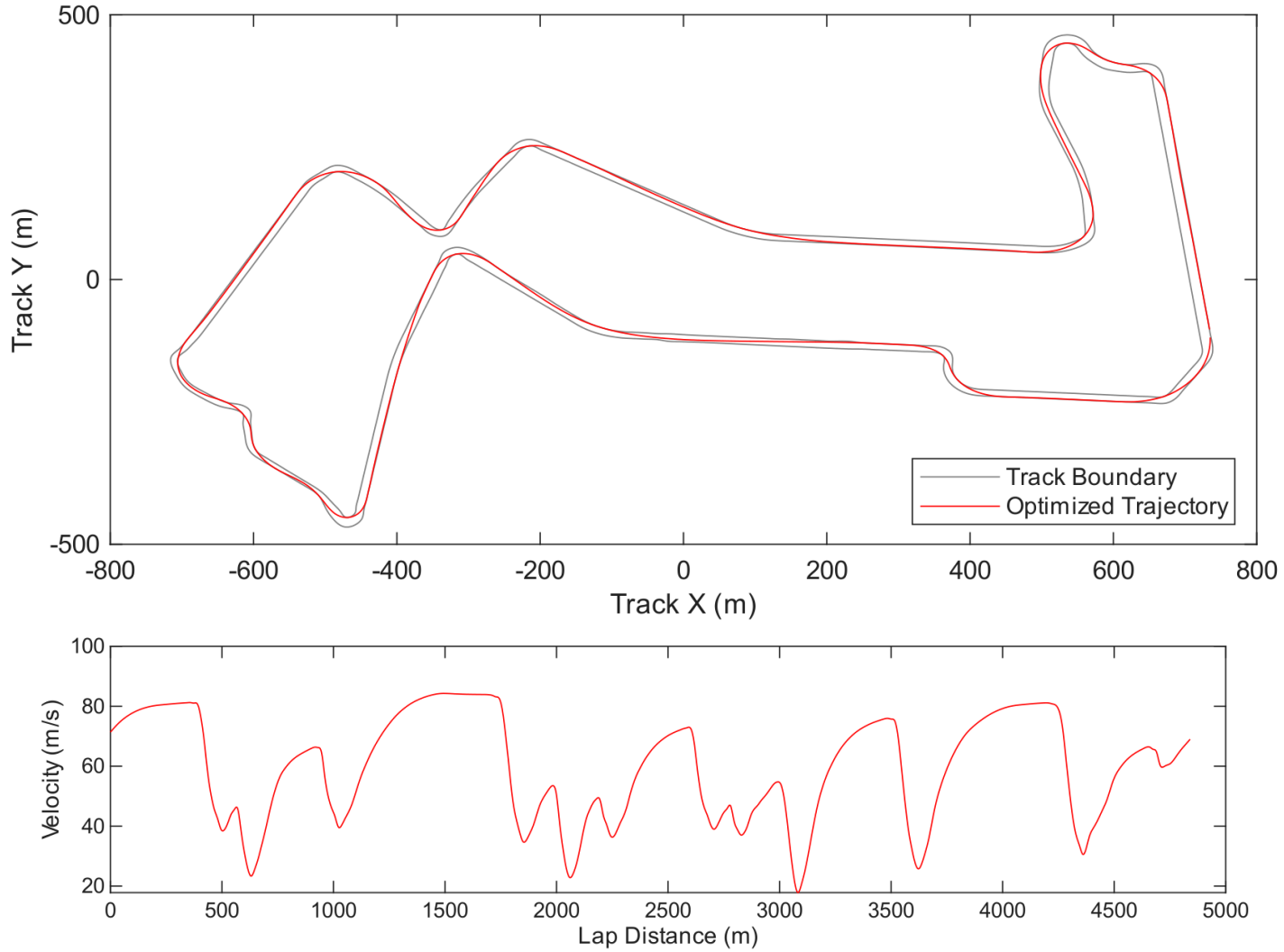


Reconstructed Surface vs. LiDAR Mesh



Path of the point mass through TrackJoint and absolute height error against the 3D mesh of the Singapore circuit (Dymola 2026x Refresh 1)

Minimum Lap Time: Singapore



- Vehicle + TrackJoint FMU → FMIOPT → CasADi direct collocation → IPOPT (MUMPS)
- 323 intervals, 324 collocation points
- **Optimal Solution Found** in 264 iterations
- Lap time **90.65 s**
- Converges on every circuit anchored on the minimum-curvature line; centre-line anchoring reproduces the failure of the companion paper

A Trivial, Feasible Initial Guess

- Chassis coordinates are relative to the track triad (heave prismatic + roll, pitch, yaw revolutes)
- Guess for a full lap:
  - Every chassis coordinate (heave, roll, pitch, yaw, w) and every velocity except $\dot{s}$ set to **zero**
  - $s(\tau) = \tau L$: a **linear ramp** over the normalized horizon $\tau \in [0, 1]$
- Vehicle sits on $r_b(s, 0)$ with R_b equal to the surface triad at every collocation point
  - Both d'Alembert relations hold at the **first iterate**; only vehicle dynamics remain
- No preliminary simulation; only the track length is needed
- Avoids the orientation-singular regions of world-frame chassis parametrizations

Circuit Coverage: 2026 F1 Calendar

Circuit	L [m]	N_seg	R_min [m]	Bank [°]	Margin
Monaco	3283	657	9.3	5.2	1.03
Zandvoort	4185	837	31.2	16.2	2.32
Singapore	4858	972	23.0	2.1	1.56
Spa	6934	1387	22.5	4.6	1.42
Silverstone	5798	1160	41.2	1.9	2.38
All 23 (range)	3283–6934	657–1387	9.3–41.2	1.5–16.2	**1.03**–2.89

Margin = min over s of 1 / (|κ| b)

- 23 circuits, uniform **5 m knot spacing**
- 657–1387 quintic segments; ≈ 1.2 MB JSON per circuit
- Injectivity margin **> 1 on all 23** circuits
- Tightest: **Monaco 1.03** (9.3 m reference-line radius)
- Steepest camber: Zandvoort, 16.2° bank

Evaluation Cost and NLP Performance

MOTORSPORTS

Per-step cost (unit point mass, RK4, 720,000 evaluations)

- Horner: 3.54 s → **4.9 µs per evaluation** of the full 2-DOF joint
- Expanded powers: 2.63 s (**−26%**), bit-identical trajectories; Dymola shares $x^2 \ldots x^5$ across channels

Accuracy vs. 60-digit reference

- Max. relative error 2.6×10^{-12} (Horner) vs 5.0×10^{-12} (expanded)

Where the time goes (464 s solve)

- 167 s objective gradients, 120 s constraints, 23 s Jacobians: **the FMU dominates**

Singapore NLP	Horner	Expanded
IPOPT status	optimal	optimal
Iterations	264	249
IPOPT time [s]	464.4	442.5
Final NLP error	8.0e−4	3.8e−4
Lap time [s]	90.6462	90.6429

- Same track, setup, guess, mesh, solver options
- Lap times agree to 3.3 ms (4×10^{-5}); evaluation form makes no measurable difference to convergence
- FMU: 35.5 MB shared library (10.9 MB archived)

Comparison with the Claytex Rail Joint

	Rail	TrackJoint
Free DOF	1 (s)	2 (s, w)
Geometry	Space curve	Monge-patch surface
Orientation	Rail tangent or 3 angle tables	Analytic triad from surface partials
Input data	xyz or xyzϕ sample table	6 quintic splines + shared knots
Constraint	1 projection	2 d'Alembert relations
Camber / crown	Constant (centre-line)	z(s, w) polynomial

Different problems, so no timing comparison: the second DOF, the analytic triad and the second d'Alembert relation all come from six scalar splines and no extra tabulated data.

Limitations and Conclusions

Limitations and Conclusions

MOTORSPORTS

Limitations

- Relies on the fitting pipeline; a broken injectivity margin is **not detected at run time**
- Validated on one circuit against LiDAR; one vehicle, setup and guess per circuit
- Uniform breaks, closed circuits, single frame_b
- Proprietary component; formulation published in full

Next steps

- Assert the injectivity margin at initialization
- Adaptive knots; four-contact-patch variant sharing one record

Conclusions

- A race-track surface packages cleanly as an **ordinary MultiBody joint**: frame_b, two coordinates, six splines
- **Regularity is solvability**: the constraint Jacobian loses rank exactly on the Frenet fold
- **Track-relative chassis coordinates** give a trivial feasible initial guess
- ≈ 2/3 of the solve is spent in the FMU: **generated code matters** (26% from polynomial form alone)


MOTORSPORTS