Multi-Sensor-Based Predictive Control for Autonomous Backward Perpendicular and Diagonal Parking

David Pérez-Morales,
Olivier Kermorgant,
Salvador Domínguez-Quijada,
Philippe Martinet

Laboratoire des Sciences du Numérique de Nantes

October 1, 2018

Car-Like Robot Rear-Wheel Driving

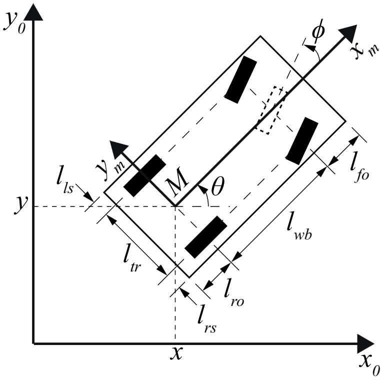


Figure: Kinematic model diagram for a car-like rear-wheel driving robot

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ \tan\phi / l_{wb} \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \dot{\phi} \tag{1}$$

Where v and $\dot{\phi}$ are the driving and steering velocities.

Multi-sensor modeling

In a static environment, the sensor feature derivative can be expressed as[1]:

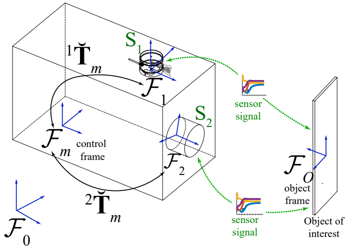


Figure: Multi-sensor model

$$\dot{\mathbf{s}}_i = \breve{\mathbf{L}}_i \breve{\mathbf{v}}_i = \underset{(\mathbb{d}_i \times 6)}{\breve{\mathbf{L}}_i} \underset{(6\times 6)}{{}^i\breve{\mathbf{T}}_m} \underset{(6\times 1)}{\breve{\mathbf{v}}_m} \quad (2)$$

[1] Kermorgant and Chaumette, "Dealing with constraints in sensor-based robot control".

Multi-sensor modeling

In a static environment, the sensor feature derivative can be expressed as[1]:

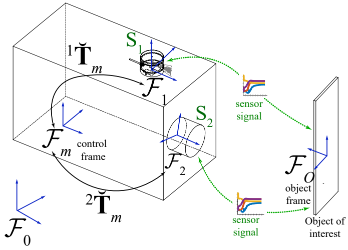


Figure: Multi-sensor model

$$\underset{(d\times 6)}{\breve{\mathbf{L}}_{\mathbf{s}}} = \breve{\mathbf{L}}\breve{\mathbf{T}}_m = \underset{(d\times 6k)}{\begin{bmatrix} \breve{\mathbf{L}}_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \breve{\mathbf{L}}_k \end{bmatrix}} \underset{(6k\times 6)}{\begin{bmatrix} {}^1\breve{\mathbf{T}}_m \\ \vdots \\ {}^k\breve{\mathbf{T}}_m \end{bmatrix}} \quad (3)$$

$$\dot{\mathbf{s}} = \breve{\mathbf{L}}_{\mathbf{s}}\breve{\mathbf{v}}_m \quad (4)$$

$$\dot{\mathbf{s}}_i = \breve{\mathbf{L}}_i\breve{\mathbf{v}}_i = \underset{(d_i\times 6)}{\breve{\mathbf{L}}_i}\ \underset{(6\times 6)}{{}^i\breve{\mathbf{T}}_m}\ \underset{(6\times 1)}{\breve{\mathbf{v}}_m} \quad (2)$$

[1] Kermorgant and Chaumette, "Dealing with constraints in sensor-based robot control".

Multi-sensor modeling

In a static environment, the sensor feature derivative can be expressed as[1]:

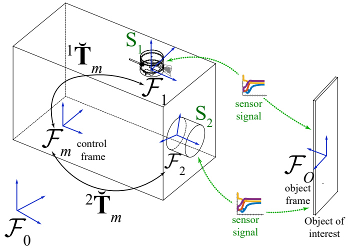


Figure: Multi-sensor model

$$\dot{\mathbf{s}}_i = \check{\mathbf{L}}_i \check{\mathbf{v}}_i = \underset{(\mathbb{d}_i \times 6)}{\check{\mathbf{L}}_i} \underset{(6\times 6)}{{}^{i}\check{\mathbf{T}}_m} \underset{(6\times 1)}{\check{\mathbf{v}}_m} \quad (2)$$

$$\underset{(d\times 6)}{\check{\mathbf{L}}_{\mathbf{s}}} = \check{\mathbf{L}}\check{\mathbf{T}}_m = \underset{(\mathbb{d}\times 6k)}{\begin{bmatrix} \check{\mathbf{L}}_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \check{\mathbf{L}}_k \end{bmatrix}} \underset{(6k\times 6)}{\begin{bmatrix} {}^{1}\check{\mathbf{T}}_m \\ \vdots \\ {}^{k}\check{\mathbf{T}}_m \end{bmatrix}} \quad (3)$$

$$\dot{\mathbf{s}} = \check{\mathbf{L}}_{\mathbf{s}} \check{\mathbf{v}}_m \quad (4)$$

Under a planar world assumption:

$$\dot{\mathbf{s}}_i = \check{\mathbf{L}}_i \check{\mathbf{v}}_i = \underset{(\mathbb{d}_i \times 3)}{\check{\mathbf{L}}_i} \underset{(3\times 3)}{{}^{i}\check{\mathbf{T}}_m} \underset{(3\times 1)}{\check{\mathbf{v}}_m} \quad (5)$$

where $\check{\mathbf{v}}_m = [v_{x_m}, v_{y_m}, \dot{\theta}]^T$

[1] Kermorgant and Chaumette, "Dealing with constraints in sensor-based robot control".

Multi-sensor modeling

Assuming $v_{y_m} = 0$ (no slipping nor skidding)

$$\mathbf{v}_m = [v_{x_m}, \dot{\theta}]^T \tag{6}$$

with $\dim(\mathbf{L}_s) = (\mathbb{d}_i \times 2)$ and $v_{x_m} = v$.

Multi-sensor modeling

Assuming $v_{y_m} = 0$ (no slipping nor skidding)

$$\mathbf{v}_m = [v_{x_m}, \dot{\theta}]^T \tag{6}$$

with $\dim(\mathbf{L}_s) = (\mathbb{d}_i \times 2)$ and $v_{x_m} = v$.

Control input

$$\mathbf{v}_r = [v, \phi]^T \tag{7}$$

with

$$\dot{\theta} = \frac{v \tan(\phi)}{l_{wb}} \tag{8}$$

Interaction Model

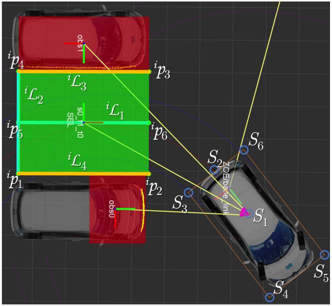


Figure: Sensors' configuration and sensor features

The sensor signals $\mathbf{s}_{i_{\mathcal{L}_j}}$ and reduced interaction matrix $\check{\mathbf{L}}_{i_{\mathcal{L}_j}}$ are defined respectively as [2]:

$$\mathbf{s}_{i_{\mathcal{L}_j}} = \left[{}^i\underline{\mathbf{u}}_j(1), {}^i\underline{\mathbf{u}}_j(2), {}^i\mathbf{h}_j(3)\right]^T \quad (9)$$

$$\check{\mathbf{L}}_{i_{\mathcal{L}_j}} = \begin{bmatrix} 0 & 0 & {}^i\underline{\mathbf{u}}_j(2) \\ 0 & 0 & -{}^i\underline{\mathbf{u}}_j(1) \\ -{}^i\underline{\mathbf{u}}_j(2) & {}^i\underline{\mathbf{u}}_j(1) & 0 \end{bmatrix} \quad (10)$$

[2] Andreff, Espiau, and Horaud, "Visual Servoing from Lines".

Task

Task sensor features

$$\mathbf{s}^t = [s_1^t, \ldots, s_9^t]^\mathsf{T} = [\mathbf{s}_1, \mathbf{s}_2]^\mathsf{T} = [\mathbf{s}_{1_{\mathcal{L}_1^{\text{off}}}}, \mathbf{s}_{2_{\mathcal{L}_1}}, \mathbf{s}_{2_{\mathcal{L}_2}}]^\mathsf{T} \qquad (11)$$

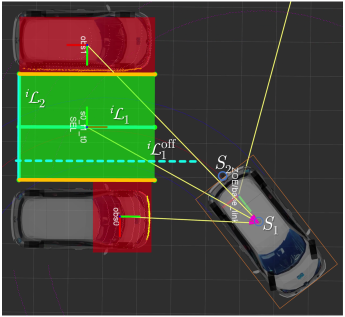


Figure: Task features used

Task

Task sensor features

$$\mathbf{s}^t = [s_1^t, \ldots, s_9^t]^\mathsf{T} = [\mathbf{s}_1, \mathbf{s}_2]^\mathsf{T} = [\mathbf{s}_{1_{\mathcal{L}_1^{\mathrm{off}}}}, \mathbf{s}_{2_{\mathcal{L}_1}}, \mathbf{s}_{2_{\mathcal{L}_2}}]^\mathsf{T} \tag{11}$$

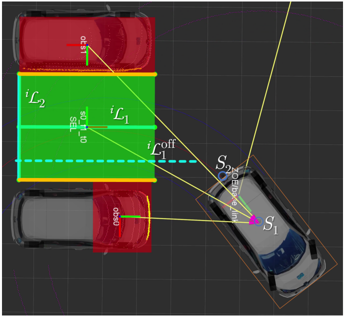


Figure: Task features used

- $\check{\mathbf{L}}_1^t$ is computed at each iteration.

Task

Task sensor features

$$\mathbf{s}^t = [s_1^t, \ldots, s_9^t]^\top = [\mathbf{s}_1, \mathbf{s}_2]^\top = [\mathbf{s}_{1_{\mathcal{L}_1^{\text{off}}}}, \mathbf{s}_{2_{\mathcal{L}_1}}, \mathbf{s}_{2_{\mathcal{L}_2}}]^\top \qquad (11)$$

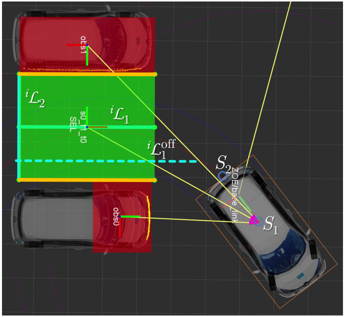
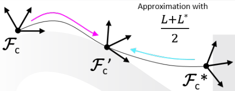


Figure: Task features used

- $\check{\mathbf{L}}_1^t$ is computed at each iteration.
- $\check{\mathbf{L}}_2^t$ is computed by a 2nd order approximation.



Task

Task sensor features

$$\mathbf{s}^t = [s_1^t, \ldots, s_9^t]^\mathsf{T} = [\mathbf{s}_1, \mathbf{s}_2]^\mathsf{T} = [\mathbf{s}_{1_{\mathcal{L}_1^{\text{off}}}}, \mathbf{s}_{2_{\mathcal{L}_1}}, \mathbf{s}_{2_{\mathcal{L}_2}}]^\mathsf{T} \quad (11)$$

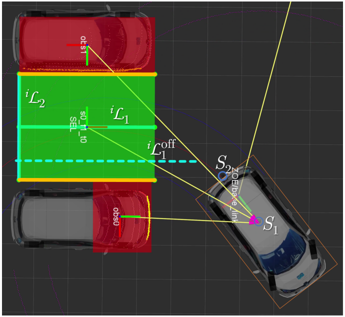
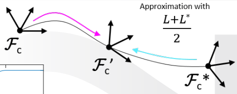
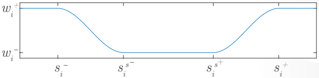


Figure: Task features used

- $\check{\mathbf{L}}_1^t$ is computed at each iteration.
- $\check{\mathbf{L}}_2^t$ is computed by a 2nd order approximation.





$$\mathbf{W}^t = \text{diag}(w_1^t, \ldots, w_9^t) \quad (12)$$

Task

Task sensor features

$$\mathbf{s}^t = [s_1^t, \dots, s_9^t]^\mathsf{T} = [\mathbf{s}_1, \mathbf{s}_2]^\mathsf{T} = [\mathbf{s}_{1_{\mathcal{L}_1^{\text{off}}}}, \mathbf{s}_{2_{\mathcal{L}_1}}, \mathbf{s}_{2_{\mathcal{L}_2}}]^\mathsf{T} \quad (11)$$

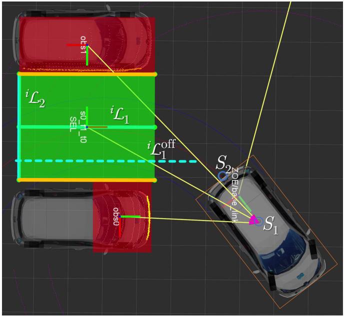
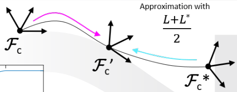
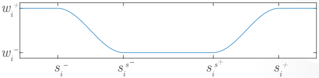


Figure: Task features used

- $\check{\mathbf{L}}_1^t$ is computed at each iteration.
- $\check{\mathbf{L}}_2^t$ is computed by a 2nd order approximation.





$$\mathbf{W}^t = \text{diag}(w_1^t, \dots, w_9^t) \quad (12)$$

where w_1^t-w_3^t, w_6^t and w_9^t are constant while the values of w_i^t $\forall i = \{4, 5, 7, 8\}$ are computed using a smooth weighting function.

Constrained sensor features

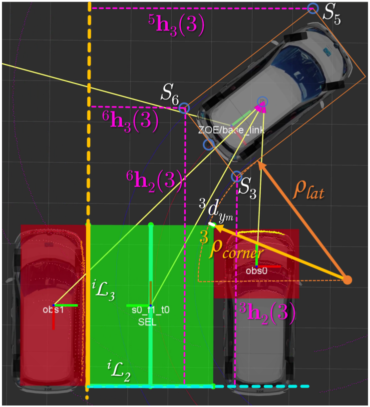


Figure: Constraints for backward non-parallel parking maneuvers

Constrained sensor features

$$\mathbf{s}^c = [s^c_1, \ldots, s^c_{10}]^\mathsf{T} = [\mathbf{s}_3, \mathbf{s}_5, \mathbf{s}_6]^\mathsf{T} \quad (13)$$

with

$$\mathbf{s}_3 = [{}^3\mathbf{h}_2(3), {}^3\mathbf{h}_4(3), {}^3X_2, {}^3Y_2, {}^3d_{lat_2}]^\mathsf{T} \quad (14a)$$

$$\mathbf{s}_5 = {}^5\mathbf{h}_3(3) \quad (14b)$$

$$\mathbf{s}_6 = [{}^6\mathbf{h}_2(3), {}^6\mathbf{h}_3(3), {}^6X_3, {}^6Y_3]^\mathsf{T}. \quad (14c)$$

Control structure

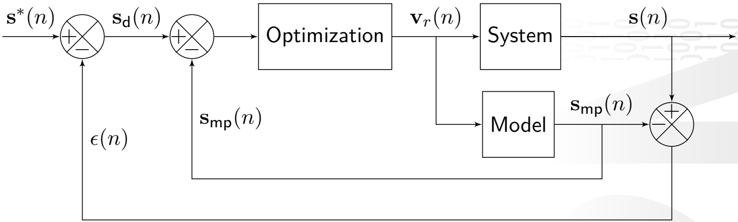


Figure: Control structure[3]

$$\mathbf{s}_{\mathsf{d}}(n) - \mathbf{s}_{\mathsf{mp}}(n) = \mathbf{s}^*(n) - \mathbf{s}(n) \tag{15}$$

[3] Guillaume Allibert, Estelle Courtial, and François Chaumette. "Predictive Control for Constrained Image-Based Visual Servoing". In: *IEEE Trans. on Robotics* 26.5 (2010), pp. 933–939.

Constraint handling

$$\mathbf{s}^c_{\mathsf{min}} \leq \mathbf{s}^c \leq \mathbf{s}^c_{\mathsf{max}} \tag{16}$$

$$|v| < v_{\mathsf{max}} \tag{17}$$

$$|\phi| < \phi_{\mathsf{max}} \tag{18}$$

$$(v_{n-1} - \Delta_{dec}) \leq v_n \leq (v_{n-1} + \Delta_{acc}) \tag{19}$$

$$(\phi_{n-1} - \Delta_{\phi}) \leq \phi_n \leq (\phi_{n-1} + \Delta_{\phi}) \tag{20}$$

$$(\dot{\phi}_{n-1} - \Delta_{\dot{\phi}}) \leq \dot{\phi}_n \leq (\dot{\phi}_{n-1} + \Delta_{\dot{\phi}}). \tag{21}$$

By writing the constraints (16)-(21) as nonlinear functions:

$$C(\mathbf{v}_r) \leq 0 \tag{22}$$

a constraint domain $\mathbb{C}$ can be defined.

Mathematical formulation

Control law

$$\min_{\tilde{\mathbf{v}}_r \in \mathbb{C}} J(\mathbf{v}_r) \tag{23}$$

with

$$J(\mathbf{v}_r) = \sum_{j=n+1}^{n+N_p} [\mathbf{s}_{\mathsf{d}} - \mathbf{s}^t_{\mathsf{mp}}(j)]^T \mathbf{Q}(j) [\mathbf{s}_{\mathsf{d}} - \mathbf{s}^t_{\mathsf{mp}}(j)] \tag{24}$$

and

$$\tilde{\mathbf{v}}_r = \{\mathbf{v}_r(n), \mathbf{v}_r(n+1), \ldots, \mathbf{v}_r(n+N_c), \ldots, \mathbf{v}_r(n+N_p-1)\} \tag{25}$$

subject to

$$\mathbf{s}^t_{\mathsf{mp}}(j) = \mathbf{s}^t_{\mathsf{mp}}(j-1) + \mathbf{L}^t_{\mathbf{s}}(j-1) T_s \mathbf{v}_m(j-1) \tag{26a}$$

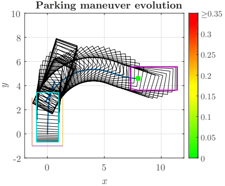
$$\mathbf{s}^c_{\mathsf{mp}}(j) = \mathbf{s}^c_{\mathsf{mp}}(j-1) + \mathbf{L}^c_{\mathbf{s}}(j-1) T_s \mathbf{v}_m(j-1) \tag{26b}$$

Mathematical formulation (contd)

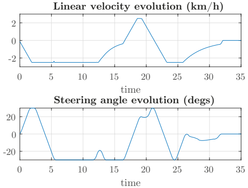
$$\mathbf{Q} = \left[\begin{array}{c|c} Q_1 \, \mathsf{diag}(w_1^t, \ldots, w_3^t) & 0_{3\times 6} \\ \hline 0_{6\times 3} & Q_2 \, \mathsf{diag}(w_4^t, \ldots, w_9^t) \end{array} \right] \tag{27}$$

It should be noted that, from $\mathbf{v}_r(n + N_c)$ to $\mathbf{v}_r(n + N_p - 1)$, the control input is constant and is equal to $\mathbf{v}_r(n + N_c)$, where N_c is the control horizon.

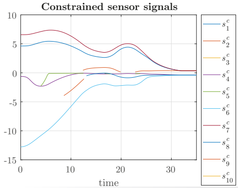
Individual cases - MATLAB simulations



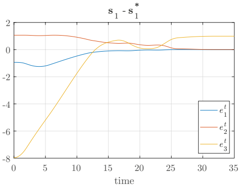
(a) Performed maneuver



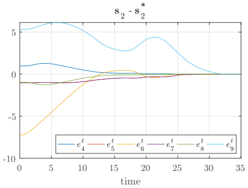
(b) Control signals



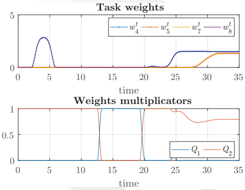
(c) Constrained sensor signals



(d) Task error signal: $\mathbf{e}_1^t$



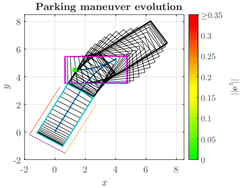
(e) Task error signal: $\mathbf{e}_2^t$



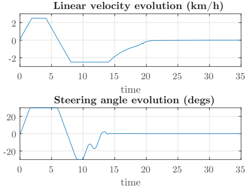
(f) Weighting signals

Figure: Constrained $\perp$ backward parking maneuver. Initial pose = (8m, 4.6m, 0°)

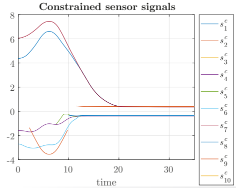
Individual cases - MATLAB simulations



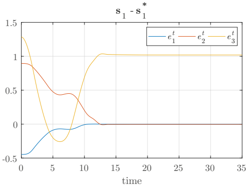
(a) Performed maneuver



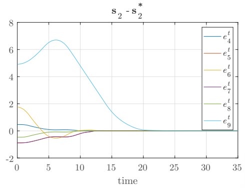
(b) Control signals



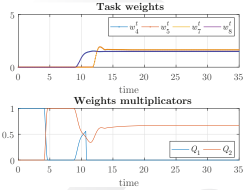
(c) Constrained sensor signals



(d) Task error signal: $\mathbf{e}_1^t$



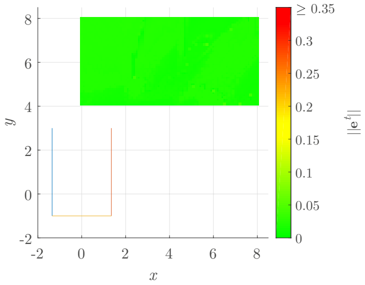
(e) Task error signal: $\mathbf{e}_2^t$



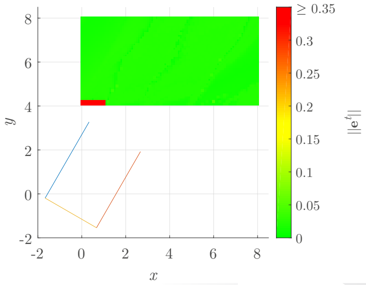
(f) Weighting signals

Figure: Constrained diagonal backward parking maneuver. Initial pose = (1.3m, 4.5m, 0°)

Exhaustive simulations



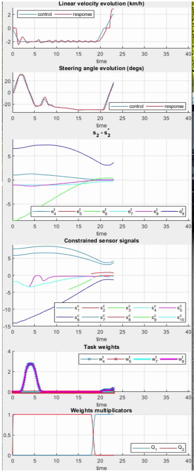
(a) Backward perpendicular case

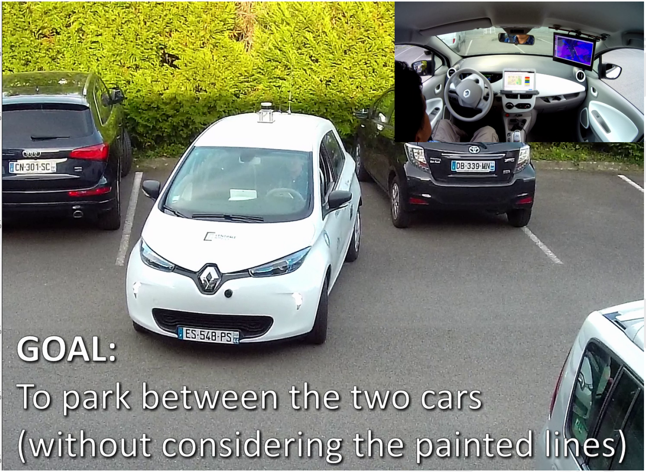


(b) Backward diagonal case

Figure: Initial orientation ($\theta_{T=0} = 0$). Parking spot length = 4m and width = 2.7m

Real experimentation results





Conclusions

- The changes in the interaction model with respect to an approach without prediction are small.

Conclusions

- The changes in the interaction model with respect to an approach without prediction are small.
- Exhaustive simulations show that the MSBPC is able to park the vehicle successfully from virtually any sensible initial position.

Conclusions

- The changes in the interaction model with respect to an approach without prediction are small.
- Exhaustive simulations show that the MSBPC is able to park the vehicle successfully from virtually any sensible initial position.
- The presented approach has been tested several times using real vehicles with positive results.