

---

# Procedura predykująca VFO dla algorytmów planowania ruchu robotów mobilnych z ograniczoną krzywizną\*

Tomasz Gawron<sup>1</sup>, Maciej Marcin Michałek<sup>1</sup>

---

## Streszczenie

W niniejszym artykule opisano nową procedurę predykującą wykorzystującą sterownik VFO (z ang. *Vector Field Orientation*) do globalnego planowania ruchu robota mobilnego przy ograniczonej krzywiznie ruchu z wykorzystaniem algorytmów próbkujących takich jak RRT i RRT\*. Opracowano wydajne analityczne procedury sprawdzające spełnienie ograniczeń na sygnały sterujące i stan robota mobilnego bez uciekania się do wykorzystania predefiniowanego zbioru prymitywów ruchu. Efektywność proponowanej procedury predykującej sprawdzono podczas planowania ruchu w wybranych środowiskach kolizyjnych. Zaplanowane sekwencje punktów przejazdowych zrealizowano podczas eksperymentów na laboratoryjnej platformie mobilnej MTracker. Uzyskane wyniki porównano ilościowo z procedurą predykującą wykorzystującą znane z literatury  $\eta^3$ -krzywe.

## 1. Wprowadzenie

Algorytmy globalnego planowania ruchu robotów mobilnych przy obecności ograniczeń na stan i sygnały sterujące stanowią istotną część nowoczesnych systemów automatycznego sterowania robotami mobilnymi. Zwinne manewrowanie robotem z ograniczoną krzywizną ruchu w środowisku kolizyjnym uzależnione jest od dokładności algorytmu planowania, jego wydajności oraz jego kompatybilności z algorytmami sterowania ze sprzężeniem zwrotnym stosowanymi podczas realizacji zaplanowanego ruchu. W poniższej pracy zajmujemy się tzw. *procedurą predykującą* wykorzystywaną w próbkujących algorytmach planowania ruchu, która odpowiada za odzwierciedlenie powyższych ograniczeń robota w generycznym próbkującym algorytmie planowania. Aby zdefiniować zadanie planowania i proponowaną procedurę predykującą, rozważmy kinematykę monocykla z ograniczoną krzywizną ruchu:

$$\dot{\mathbf{q}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ \cos \theta \\ \sin \theta \end{bmatrix} u_2 = [\mathbf{g}_1 \ \mathbf{g}_2(\theta)] \mathbf{u}, \quad (1)$$

$$\forall t \geq 0 \quad |\kappa(t)| \triangleq \frac{|u_1(t)|}{|u_2(t)|} \leq \kappa_b, \quad 0 < \kappa_b < \infty, \quad (2)$$

---

\*Praca finansowana w ramach DS nr 09/93/DSPB/0611.

<sup>1</sup>Katedra Sterowania i Inżynierii Systemów, Politechnika Poznańska, ul. Piotrowo 3A, 60-965 Poznań  
tomasz.gawron@doctorate.put.poznan.pl, maciej.michalek@put.poznan.pl

gdzie  $\mathbf{q} = [\theta \ x \ y]^\top = [\theta \ \bar{\mathbf{q}}]^\top \in Q = \mathbb{R}^3$  stanowi konfigurację robota, a  $\mathbf{u} = [u_1 \ u_2]^\top \in \mathcal{U} \subset \mathbb{R}^2$  jest wektorem sygnałów sterujących, który składa się z prędkości kątowej platformy oznaczonej jako  $u_1$  i prędkości postępowej  $u_2$  punktu prowadzenia  $\bar{\mathbf{q}} = [x \ y]^\top$  widocznego na Rys. 1. Ograniczenie krzywizny ruchu (2) definiuje zbiór dopuszczalnych sterowań  $\mathcal{U} = \{\mathbf{u} \in \mathbb{R}^2 : |u_1| \leq \kappa_b |u_2|\}$ , który zależy od stałej  $\kappa_b$  wynikłej ze specyfikacji konkretnego zadania ruchu robota.

W niniejszym artykule rozważamy problem *globalnego planowania motywowanego sterowaniem* dla systemu (1) przy ograniczeniu (2) na wejście sterujące i ograniczeniach na stan. Aby zdefiniować ograniczenia na stan, wprowadźmy dopuszczalny podzbiór przestrzeni konfiguracyjnej  $Q_a \subseteq Q$ . Przyjmując, że konfiguracja początkowa  $\mathbf{q}_0 = \mathbf{q}(0)$  oraz konfiguracja referencyjna  $\mathbf{q}_d = [\theta_d \ x_d \ y_d]^\top = [\theta_d \ \bar{\mathbf{q}}_d]^\top \in Q_a$  są znane a priori, należy znaleźć sekwencję punktów przejazdowych  $\mathbf{q}_i = [\theta_i \ x_i \ y_i]^\top = [\theta_i \ \bar{\mathbf{q}}_i]^\top \in Q_a, i = 0, 1, \dots, N$ , której realizacja poprzez zastosowanie sterowania VFO zdefiniowanego w Rozdziale 2 do systemu (1) przy ograniczeniu (2) zapewni, że

$$\forall t \geq 0 \ \mathbf{q}(t) \in Q_a \wedge \mathbf{q}(t) \xrightarrow{t \rightarrow T} \mathbf{q}_d, \quad (3)$$

gdzie  $T \in (0, \infty)$  stanowi horyzont czasowy sterowania. Oznacza to, że robot musi zostać sprowadzony do konfiguracji referencyjnej  $\mathbf{q}_d$  przy jednoczesnym utrzymaniu jego konfiguracji w zbiorze  $Q_a$ . Zakłada się, że zbiór  $Q_a$  jest znany a priori. Podobnie do propozycji [1], zdefiniowane zadanie obejmuje planowanie sekwencji podzadania sterowania ze sprzężeniem zwrotnym, w przeciwieństwie do klasycznego zadania *planowania ruchu* zakładającego wyznaczenie sterowań w pętli otwartej.

Problemy planowania są często rozwiązywane przy użyciu algorytmów cyklicznie wykonujących pseudolosowe (np. [3]) lub deterministyczne (np. [9]) próbkowanie zbioru dopuszczalnych konfiguracji  $Q_a$  w celu konstrukcji grafu dopuszczalnych ruchów pomiędzy konfiguracjami pośrednimi nazywanymi dalej punktami przejazdowymi. Graf ten jest systematycznie rozbudowywany dopóki wystąpi w nim połączenie pomiędzy konfiguracją początkową  $\mathbf{q}_0$  i konfiguracją zadaną  $\mathbf{q}_d$ . Procedura, która sprawdza możliwość dodania nowego punktu przejazdowego do grafu i oblicza koszt ruchu pomiędzy poszczególnymi punktami przejazdowymi nazywana jest *procedurą predykującą* (z ang. *extend procedure*) dalej nazywaną skrótowo PP. W ogólności PP dla próbkującego algorytmu planowania jest odwzorowaniem:

$$c = \text{extend}(\mathbf{q}_c, \mathbf{q}_s) : Q_a \times Q_a \rightarrow \mathbb{R}_{\geq 0}, \quad (4)$$

gdzie  $\mathbf{q}_c = [\theta_c \ x_c \ y_c]^\top = [\theta_c \ \bar{\mathbf{q}}_c]^\top \in Q_a$  to wybrany przez algorytm planowania punkt przejazdowy należący do grafu (zwykle tzw. najbliższy sąsiad  $\mathbf{q}_s$ ), a  $\mathbf{q}_s = [\theta_s \ x_s \ y_s]^\top = [\theta_s \ \bar{\mathbf{q}}_s]^\top \in Q_a$  to próbka z przestrzeni konfiguracyjnej wygenerowana przez algorytm planowania. PP oblicza koszt ruchu  $c \in [0, \infty)$  poniesiony przy przeprowadzeniu robota z  $\mathbf{q}_c$  to  $\mathbf{q}_s$ . Koszt  $c$  osiąga nieskończoność, jeśli przeprowadzenie robota z  $\mathbf{q}_c$  to  $\mathbf{q}_s$  wiąże się z naruszeniem ograniczeń wynikających z definicji zadania (3) i ograniczenia (2). W tym wypadku algorytm odrzuca próbkę  $\mathbf{q}_s$  i przechodzi do kolejnej iteracji.

Klasyczne procedury predykujące implementowane są poprzez całkowanie numeryczne odpowiedzi systemu na sterowania odcinkami stałe lub poprzez wybór z predefiniowanego skończonego zbioru prymitywów ruchu (np. [9]). Zastosowanie

w.w. podejść wiąże się z problemami takimi jak utrudnione projektowanie sterowania ze sprzężeniem zwrotnym, konieczność odpowiedniej dyskretyzacji przestrzeni problemu (kluczowa jest topologiczna struktura i wielkość zbioru prymitywów), a także brak zbieżności do planu optymalnego w sensie przyjętego kosztu ruchu ze względu na dyskretyzację w przestrzeni dopuszczalnych sterowań co pokazano w [5] poprzez analizę ilościową. W literaturze można znaleźć również inne dedykowane przykłady procedur predykujących takie jak ścieżki Dubinsa i krzywe parametryczne [10]. W pracach [5] i [7] zaproponowano natomiast ciekawe PP motywowane sterowaniem.

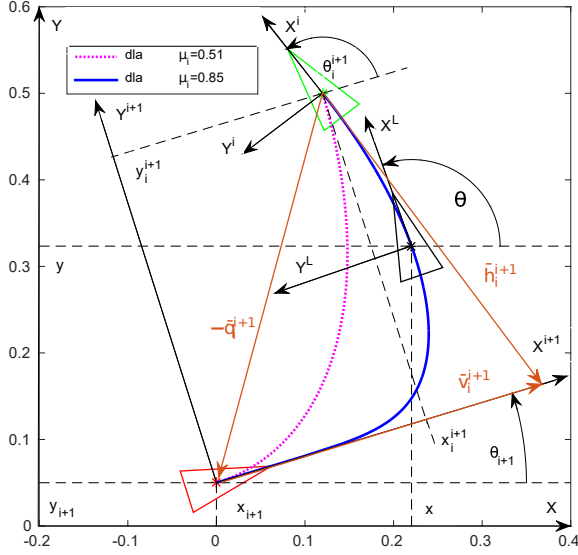
W poniższej pracy proponujemy nową PP generującą punkty przejazdowe, które mogą być zrealizowane przez sterownik VFO opisany w [4]. Charakteryzuje się ona szeregiem pożądanych własności, które wyróżniają ją spośród rozwiązań dostępnych w literaturze. Mianowicie procedura ta:

- generuje plan ruchu biorąc pod uwagę dynamikę systemu sterowania w pętli zamkniętej podczas realizacji ruchu w przeciwieństwie do np. [10] i [8],
- eliminuje konieczność projektowania sterowania realizującego plan ruchu - wystarczy zastosowanie gotowego sterownika VFO,
- gwarantuje spełnienie ograniczenia krzywizny (2) i nie wymaga zastosowania metod numerycznych do sprawdzenia spełnienia tego ograniczenia,
- w przeciwieństwie do [10] i innych pozwala na wydajne sprawdzenie ograniczeń na stan w przypadku, gdy podzbiór  $Q_a$  definiowany jest zbiorem wielokątów wypukłych; procedura sprawdzająca jest dokładna (tj. analityczna) i ma złożoność  $O(P)$ , gdzie  $P$  to maksymalna ilość boków wielokąta,
- nie wykorzystuje dyskretyzacji w przestrzeni stanu lub sterowań na potrzeby predykcji ruchu robota, co oznacza że proces planowania jest globalnie asymptotycznie optymalny,
- dzięki swojej wydajności jest efektywną miarą odległości wykorzystywaną do poszukiwania najbliższych sąsiadów w grafie punktów przejazdowych, co znacząco wpływa na proces planowania (patrz [6]).

## 2. Sterowanie VFO

Do wyjaśnienia proponowanej procedury predykującej niezbędne jest wprowadzenie sterowania VFO dla zadania przejazdu przez zbiór punktów z [4]. Plan ruchu składa się z sekwencji punktów przejazdowych generowanych przez procedurę predykującą, stąd gdy sterownik VFO prowadzi robota do punktu przejazdowego  $\mathbf{q}_{i+1} = [\theta_{i+1} \ x_{i+1} \ y_{i+1}]^\top = [\theta_{i+1} \ \bar{\mathbf{q}}_{i+1}^\top]^\top \in Q_a$ , mówimy że robot znajduje się w  $i$ -tym segmencie ruchu, którego wykonanie rozpoczęto w chwili czasu  $t_i$ . Przyjmuje się, że punkt  $\mathbf{q}_0$  jest równy konfiguracji początkowej  $\mathbf{q}(0)$  a ostatni punkt przejazdowy  $\mathbf{q}_N$  równy jest konfiguracji referencyjnej  $\mathbf{q}_d$ . Dla uproszczenia późniejszej analizy sterowanie VFO zapisano w układzie współrzędnych  $(i+1)$ -szego punktu przejazdowego i wprowadzono następującą notację. Zmienne wyrażone w układzie  $\{i+1\}$  oznaczono indeksem górnym " $i+1$ ". Pozostałe zmienne wyrażono w układzie globalnym. Na przykład  $\mathbf{q}_i^{i+1} = [\theta_i^{i+1} \ x_i^{i+1} \ y_i^{i+1}]^\top = [\theta_i^{i+1} \ (\bar{\mathbf{q}}_i^{i+1})^\top]^\top \in Q_a$  oznacza  $i$ -ty punkt przejazdowy w układzie  $(i+1)$ -szego punktu przejazdowego co pokazano na Rys. 1.

Zgodnie z powyższą notacją sterowanie VFO dla  $i$ -tego segmentu ruchu moż-



Rys. 1: Ścieżki kreślone przez monocykl przy zastosowaniu sterowania VFO w  $i$ -tym segmencie ruchu dla 2 wartości parametru  $\mu_i$  i  $k_p = 1$

na zapisać następująco:

$$\mathbf{u} \triangleq [h_{ai}^{i+1} \quad \rho_i \sigma_i \cos(e_{ai}^{i+1})]^\top, \quad (5)$$

$$h_{ai}^{i+1} \triangleq k_a e_{ai}^{i+1} + \dot{\theta}_{ai}^{i+1}, \quad e_{ai}^{i+1} \triangleq \theta_{ai}^{i+1} - \theta^{i+1}, \quad \theta_{ai}^{i+1} \triangleq \text{Atan2c}(\sigma_i h_{yi}^{i+1}, \sigma_i h_{xi}^{i+1}) \in \mathbb{R}, \quad (6)$$

$$\bar{\mathbf{h}}_i^{i+1} = [h_{xi}^{i+1} \quad h_{yi}^{i+1}]^\top \triangleq -k_p \bar{\mathbf{q}}^{i+1} + \bar{\mathbf{v}}_i^{i+1}, \quad \bar{\mathbf{v}}_i^{i+1} \triangleq -k_p \mu_i \sigma_i \|\bar{\mathbf{q}}^{i+1}\| [1 \ 0]^\top, \quad \mu_i = \eta_i k_p^{-1},$$

$$\rho_i \triangleq \begin{cases} U_2 & \text{dla } i < N-1, \\ U_2 \frac{\|\bar{\mathbf{q}}^{i+1}\|^\gamma}{\|\bar{\mathbf{q}}^{i+1}(t_i)\|} & \text{dla } i = N-1, \end{cases} \quad (7)$$

gdzie  $k_a, k_p, U_2 > 0$  i  $\gamma \in (0, 1]$  są parametrami projektowymi, a  $\bar{\mathbf{q}}^{i+1} = [x^{i+1} \ y^{i+1}]^\top$  oznacza pozycję robota w układzie  $\{i+1\}$ . Operator całkowy  $\text{Atan2c}(\cdot, \cdot)$  stanowi uciągloną wersję funkcji  $\text{Atan2}(\cdot, \cdot)$ . Sterowanie orientujące  $u_1$  gwarantuje zbieżność konfiguracji robota do pola wektorowego zbieżności  $\mathbf{h}_i^{i+1} \triangleq [h_{ai}^{i+1} \ (\bar{\mathbf{h}}_i^{i+1})^\top]^\top$ , którego krzywe całkowe prowadzą do  $(i+1)$ -szego punktu przejazdowego. Zbieżność osiągnięto przez śledzenie orientacji pomocniczej  $\theta_{ai}^{i+1}$ , tj. minimalizację błędu śledzenia orientacji pomocniczej  $e_{ai}^{i+1}$ . Kiedy  $e_{ai}^{i+1} = 0$ , mówimy o *nominalnych warunkach ruchu*. Pole zbieżności  $\mathbf{h}_i^{i+1}$  jest kształtowane poprzez względny współczynnik naprowadzania  $\mu_i \in (0, 1)$ . Jak widać na Rys. 1, względny współczynnik naprowadzania  $\mu_i$  wpływa na szybkość zbieżności orientacji  $\theta^{i+1}$  do zera. Strategia ruchu w  $i$ -tym segmencie zależy od zmiennej decyzyjnej  $\sigma_i \in \{-1, 1\}$ , przy czym  $\sigma_i = -1$  wymusza jazdę tyłem podczas gdy  $\sigma_i = 1$  wymusza jazdę przodem. Prędkość postępową

robota kształtowana jest przez profil  $p_i$ , którego definicja (patrz (7)) zakłada, że robot powinien podróżować przez punkty przejazdowe ze stałą prędkością postępową  $U_2$  i zatrzymać się w ostatnim punkcie przejazdowym.

### 3. Procedura predykująca

**Opis ogólny** Dla jasności wyводу najpierw przedstawiony zostanie ogólny schemat postępowania PP VFO. Szczegóły kolejnych jej kroków zostaną wyjaśnione w następnych akapitach. Dla uproszczenia zakłada się, że graf punktów przejazdowych budowany przez algorytm planowania ma pierwszy wierzchołek tożsamy z ostatnim punktem  $q_N = q_d$  i jest on rozwijany w kierunku punktu początkowego  $q_0$ . Proponowana PP może być bez przeszkód przystosowana do algorytmów rozwijających graf w dowolnym kierunku (np. bidirectional RRT). W celu objaśnienia proponowanej procedury, rozważmy jej działanie na przykładzie algorytmu RRT. Przyjmuje się, że  $q_s$  jest konfiguracją spróbkowaną (tj. wylosowaną przez RRT), a  $q_c$  stanowi punkt zawarty w grafie punktów przejazdowych, który jest najbliższy punktowi  $q_s$  względem przyjętej miary odległości. W odróżnieniu od innych PP, przyjmujemy że wspomniana miara odległości między dwoma punktami to koszt ruchu  $c$  wynikły z zastosowania proponowanej PP przy założeniu upraszczającym  $Q_a = Q$ . Oznacza to, że proponowana miara odległości jest optymalna w środowisku bezkolizyjnym. PP przebiega następująco:

- K1. Zakładamy, że  $q_c$  i  $q_s$  tworzą segment ruchu, tj.  $q_i := q_s$  i  $q_{i+1} := q_c$ .
- K2. Obliczamy parametry VFO ( $\mu_i, \sigma_i$ ) gwarantujące nominalne warunki ruchu.
- K3. Jeśli parametry obliczone w K2 są niedopuszczalne, obliczamy dodatkowy pośredni punkt przejazdowy  $w_i = [\phi_i \ x_{wi} \ y_{wi}]^T$  pomiędzy  $q_i$  i  $q_{i+1}$  oraz parametry VFO ( $\mu_i, \sigma_i$ ) dla dwóch powstałych segmentów: pierwszego łączącego  $q_i$  z  $w_i$  i drugiego łączącego  $w_i$  z  $q_{i+1}$ . Kolejne kroki wykonujemy osobno dla obu powstałych segmentów ruchu.
- K4. Wyznaczamy krzywą  $x^{i+1} = f(y^{i+1})$  opisującą ruch robota.
- K5. Wyznaczamy maksymalną krzywiznę  $|\hat{\kappa}_i|$  ruchu robota w segmencie ruchu. Sprawdzamy spełnienie ograniczenia (2). Jeśli  $|\hat{\kappa}_i| > \kappa_b$ , przyjmujemy koszt ruchu  $c = \infty$  i pomijamy K6, w przeciwnym razie przechodzimy do K6.
- K6. Obliczamy koszt  $c$  wykorzystując krzywą całkową  $x^{i+1} = f(y^{i+1})$ .

**Wyznaczanie parametrów VFO (krok K2)** W celu zapewnienia nominalnych warunków ruchu dla punktów przejazdowych  $q_{i+1}$  i  $q_i$ , orientacja pomocnicza musi spełniać zależność  $\theta_{ai}^{i+1}(t_i) = \theta_i^{i+1}$  (por. (6)). Można więc założyć  $\theta_{ai}^{i+1}(t_i) = \theta_i^{i+1}$  w równaniu (6) i wyznaczyć  $\mu_i, \sigma_i$ , aby obliczyć proponowaną strategię ruchu  $\hat{\sigma}_i \in \{-1, 1\}$  i względny współczynnik naprowadzania  $\hat{\mu}_i$  zgodnie z równaniami:

$$\hat{\mu}_i = |k_i|, \quad \hat{\sigma}_i = \text{sign}(k_i), \quad k_i = \left( \frac{y_i^{i+1}}{\text{tg}(\theta_i^{i+1})} - x_i^{i+1} \right) / \|\bar{q}_i^{i+1}\|. \quad (8)$$

Jak pokazano poprzednio w [2], jeśli  $\theta_{ai}^{i+1}(t_i)$  obliczona z (6) przy  $\mu_i = \hat{\mu}_i$  i  $\sigma_i = \hat{\sigma}_i$  nie jest równa zadanej orientacji  $\theta_i^{i+1}$ , (8) nie jest dobrze zdefiniowane, lub  $\mu_i \notin (0.5, 1)$

(przedział wyjaśniono przy opisie K5), to wnioskujemy że nominalne warunki ruchu nie mogą zostać zapewnione dla zadanej pary punktów przejazdowych. Ponadto, w celu uproszczenia procedur sprawdzania kolizji oraz wyznaczania punktu pośredniego (patrz K3), zakłada się że  $|\theta_i^{i+1}| \leq \frac{\pi}{2}$  oraz  $\sigma_i = -\text{sign}(x_i^{i+1})$ . Jeśli nominalne warunki ruchu mogą zostać zapewnione przy powyższych dodatkowych ograniczeniach, przyjmuje się  $\mu_i := \hat{\mu}_i$ ,  $\sigma_i := \hat{\sigma}_i$ . W przeciwnym wypadku należy wyznaczyć dodatkowy punkt przejazdowy  $w_i$  (por. krok K3).

**Analityczna predykcja ruchu robota (krok K4)** W pracy [2] wyprowadzono krzywą całkową  $x^{i+1} = f(y^{i+1})$  opisującą ruch robota w warunkach nominalnych zapewnionych przez odpowiednie parametry  $(\mu_i, \sigma_i)$  wyznaczone w kroku K2. Krzywą tą widać na Rys. 1. Można ją zapisać zależnością:

$$x^{i+1} = \frac{\sigma_i |y^{i+1}|}{2} \left[ \left( \frac{y^{i+1}}{\psi_i} \right)^{\mu_i} - \left( \frac{y^{i+1}}{\psi_i} \right)^{-\mu_i} \right], \quad \psi_i = y_i^{i+1} \exp \left( \frac{-\text{arsinh}(x_i^{i+1}/y_i^{i+1})}{\mu_i \sigma_i \text{sign}(y_i^{i+1})} \right), \quad (9)$$

z niezerowym parametrem  $\psi_i$ . Krzywa ta jest niezależna od wyboru profilu prędkości  $\rho_i$ . Szczegółowe wyprowadzenie zależności (9) można znaleźć w [2].

Przydatna w projektowaniu postaci kosztu ruchu  $c$  dokładna długość krzywej (9) nie może być wyznaczona analitycznie. Można jednak podać jej górne ograniczenie  $L_i$  i następujący wywiedziony z niego koszt ruchu  $c_i$ :

$$c_i \triangleq (1 - \mu_i) \|\mathbf{q}_i^{i+1}\| + \mu_i L_i, \quad L_i = \|\bar{\mathbf{h}}_i(t_i)\|/k_p + \|\bar{\mathbf{v}}_i(t_i)\|/k_p.$$

Ograniczenie  $L_i$  ma łatwą interpretację geometryczną, widoczną na Rys. 1. Koszt ruchu  $c_i$  zdefiniowano poprzez heurystyczne skalowanie ograniczenia  $L_i$ , które bierze pod uwagę fakt, że zwiększanie współczynnika  $\mu_i$  skutkuje dłuższymi ścieżkami robota. Taka propozycja pozwala na mniej konserwatywne szacowanie kosztu ruchu, jednak nie jest ona kluczowa dla proponowanej metody - można zaproponować dowolny koszt ruchu korzystając z krzywej (9).

**Spełnienie ograniczenia krzywizny ruchu (krok K5)** Zgodnie z analizą z [2], przy założeniu  $\mu_i \in (0.5, 1)$  przyjętym w kroku K2, krzywizna ruchu robota wzdłuż krzywej (9) ma globalne maksimum  $\hat{\kappa}_i$  zależne liniowo od parametru  $\psi_i$ . Implikuje to istnienie granicznej wartości  $\hat{\psi}_i$  odpowiadającej granicznej krzywej całkowej takiej, że  $|\psi_i| > |\hat{\psi}_i| \Rightarrow |\hat{\kappa}_i| < \kappa_b$ . Stąd sprawdzenie spełnienia ograniczenia (2) wymaga jedynie sprawdzenia wartości parametru  $\psi_i$ . Wzór na konkretną wartość maksimum  $\hat{\kappa}_i$  dla  $\psi_i = 1$  potrzebną do wyznaczenia  $|\hat{\psi}_i|$  podano w [2].

**Wyznaczenie pośredniego punktu przejazdowego (krok K3)** Pośredni punkt przejazdowy  $w_i = [\phi_i \ x_{wi} \ y_{wi}]^\top$  wyznaczany jest poprzez wykorzystanie faktu, że orientacja pomocnicza  $\theta_{ai}^{i+1}$  jest stała wzdłuż prostej  $y^{i+1} = a_i^{i+1} x^{i+1}$ . Zakładając  $\sigma_i = -\text{sign}(x_i^{i+1})$ , można podstawić  $y^{i+1} = a_i^{i+1} x^{i+1}$  do (6), aby uzyskać:

$$a_i^{i+1} = \left( \mu_i \sqrt{1 + (1 - \mu_i^2) \text{tg}(\theta_{ai}^{i+1})^2} - 1 \right) / (\text{tg}(\theta_{ai}^{i+1}) \mu_i^2 - 1 / \text{tg}(\theta_{ai}^{i+1})). \quad (10)$$

Założmy teraz chwilowo, że orientacja  $\phi_i$  i parametr  $\mu_i$  zostały wybrane. Wtedy można wykorzystać zależność (10) dla segmentu ruchu łączącego  $\mathbf{w}_i$  z  $\mathbf{q}_{i+1}$  poprzez podstawienie  $\theta_{ai}^{i+1} := \phi_i^{i+1}$ , aby wyznaczyć współczynnik kierunkowy  $a_i^{i+1}$  prostej, na której musi leżeć punkt  $\mathbf{w}_i$ . Takie same obliczenia należy wykonać dla segmentu łączącego  $\mathbf{q}_i$  z  $\mathbf{w}_i$  podstawiając  $\theta_{ai}^{i+1} := \theta_i^{i+1} - \phi_i^{i+1}$  w (10) aby określić drugą linię prostą, na której musi leżeć punkt  $\mathbf{w}_i$  i tym samym jego dokładną pozycję. Orientację  $\phi_i$  należy wybrać tak, żeby  $\phi_i \neq \theta_{i+1}$  lub  $\phi_i \neq \theta_i$  aby powstałe dwa segmenty ruchu były dobrze określone w przypadku  $\theta_{i+1} = \theta_i$ . Dla zapewnienia równomiernego rozłożenia kątów w punktach przejazdowych przyjęto heurystyczną metodę wyboru parametrów:  $\phi_i^{i+1} = \theta_i^{i+1}/2$  dla  $\theta_i^{i+1} \neq 0$ ,  $\phi_i^{i+1} = \pi/4$  dla  $\theta_i^{i+1} = 0$ , i  $\mu_i = 0.65$  w każdym przypadku.

**Spełnienie ograniczeń na stan** W przypadku definicji  $Q_a$  poprzez zbiór wielokątów wypukłych można analitycznie sprawdzić czy krzywa (9) mieści się w danym wielokącie wypukłym. Wykorzystując zależność (10) wyznacza się na krzywej (9) punkt  $\mathbf{s}_i$  o stycznej z współczynnikiem nachylenia równym współczynnikowi nachylenia prostej, na której leży dany bok wielokąta. Punkt najbliższy danemu bokowi wielokąta to punkt  $\mathbf{s}_i$  lub jeden z końców segmentu ruchu. Można pokazać, że jeśli te 3 punkty są zawarte w badanym wielokącie to cała krzywa (9) łącząca początek i koniec segmentu ruchu jest również w nim zawarta. Dzięki temu sprawdzanie spełnienia ograniczeń na stan jest dokładne (tj. brak dyskretyzacji) i znacząco szybsze od naiwnego algorytmu badającego wiele punktów na krzywej (9).

#### 4. Wyniki badań symulacyjnych

Na potrzeby symulacji oraz eksperymentów PP VFO została zintegrowana z algorytmami planowania z biblioteki OMPL. Zbiór  $Q_a$  zdefiniowano przez siatkę zajętości o rastrze 0.005 m. Rozważono dwa scenariusze ruchu pokazane na Rys. 2 i Rys. 3. Tabela 1 przedstawia podsumowanie wyników 100 symulacji (50 symulacji S1 i 50 symulacji S2). W każdej symulacji sprawdzano jednocześnie PP VFO i procedurę opartą o  $\eta^3$ -krzywe z [8]. W przypadku  $\eta^3$ -krzywych jako koszt ruchu przyjęto długość ścieżki uzyskaną przez całkowanie numeryczne. Sprawdzono następujące wskaźniki jakości planów: dokładna długość planowanej ścieżki  $l_p$ , wskaźnik average speed arc length  $n_{spal}$  stanowiący miarę gładkości ruchu (0 jest najlepszą wartością, uzasadnienie i definicja w [5]), ilość połączeń w wygenerowanym drzewie  $N_v$ , i ilość segmentów ruchu w wynikowym planie  $N_s$ . Na podstawie wartości wskaźnika  $n_{spal}$  wnioskuje się, że proponowana PP generuje plany o większej gładkości ruchu. Z drugiej strony, wykorzystanie  $\eta^3$ -krzywych powoduje generowanie krótszych ścieżek robota. Ta tendencja do generowania krótszych ścieżek utrzymuje się w przypadku algorytmu RRT\*, jednak różnica w ich długości znacząco spada w przeciwieństwie do różnicy wskaźnika  $n_{spal}$ . Wydaje się więc, że podejście VFO korzysta znacząco z optymalizacji kosztu  $c$ . Warto również wspomnieć, że PP VFO powoduje tworzenie znacznie większej ilości krawędzi w drzewach rozwiązań przy podobnej długości rozwiązań i dokładnie takim samym czasie planowania jak w przypadku  $\eta^3$ -krzywych. Oznacza to, że w tym samym czasie sprawdzana jest wyraźnie większa ilość potencjalnych rozwiązań co wskazuje na wydajność PP VFO wynikającą z jej analitycznego charakteru. Tenden-

Tab. 1: Średnie wyniki 100 symulacji planowania przy  $\kappa_b = 3 \text{ m}^{-1}$

algorytm planowania	$l_p[m]$	$n_{spal}$	$N_v$	$N_s$
RRT/VFO (5 s)	$10.1 \pm 1.8$	$-1.3 \pm 0.4$	$1261 \pm 308$	$9.3 \pm 1.9$
RRT/ $\eta^3$ (5 s)	$8.6 \pm 0.6$	$-6.8 \pm 1.5$	$920 \pm 216$	$10.1 \pm 2.3$
RRT*/VFO (60 s)	$8.1 \pm 0.7$	$-0.5 \pm 0.2$	$2756 \pm 402$	$10.2 \pm 2.1$
RRT*/ $\eta^3$ (60 s)	$7.67 \pm 0.6$	$-7.2 \pm 1.2$	$2025 \pm 100$	$8.9 \pm 3.0$

cja ta powinna się nasilić w przypadku definicji zbioru  $Q_a$  przez wielokąty wypukłe.

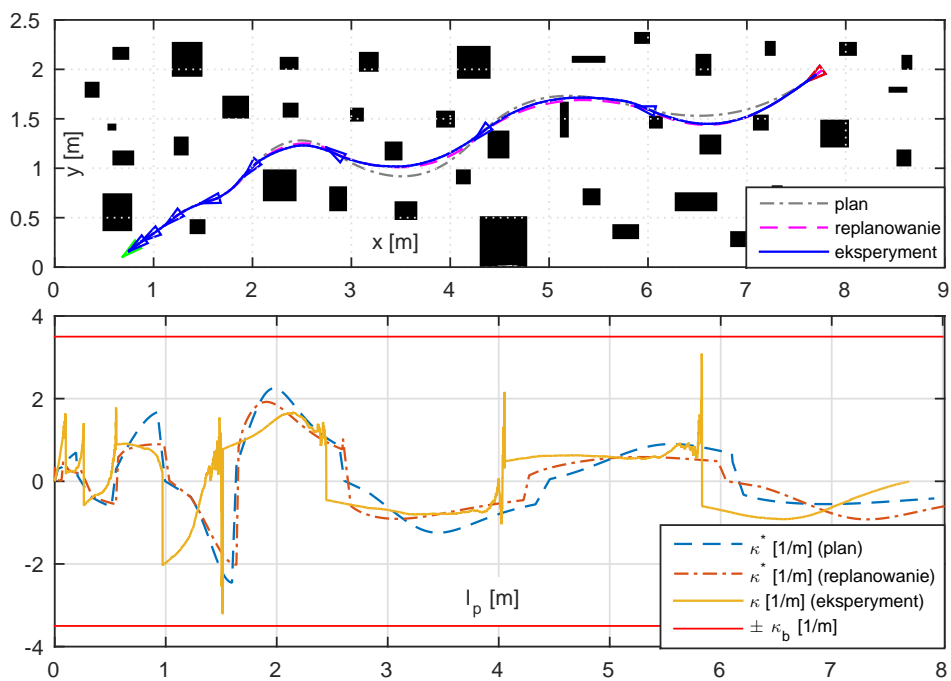
## 5. Wyniki eksperymentów

Eksperymenty przeprowadzono na laboratoryjnej platformie mobilnej MTracker o kinematyce klasy (2,0) z kołami o promieniu 0.025 m i odległością między nimi wynoszącą 0.145 m. Plan obliczony a priori zrealizowano na bieżąco na pokładzie robota poprzez zastosowanie sterowania VFO. Stan robota estymowany był prostym systemem fuzji sensorycznej składającym się z predyktora ruchu i systemu lokalizacji bezwzględnej detekującego pasywnie znaczniki Aruco z wykorzystaniem pokładowej kamery robota.  $(i+1)$ -szy segment ruchu rozpoczynano gdy  $\|\mathbf{q}^{i+1}\| < 0.05 \text{ m}$ . Parametry projektowe sterownika VFO wybrano następująco:  $\gamma = 0.5$ ,  $U_2 = 0.2 \text{ m/s}$ ,  $k_a = 2 U_2$ ,  $k_p = U_2$ . Krzywiznę ruchu robota ograniczono programowo do wartości  $\kappa_b = 3.5 \text{ m}^{-1}$ . Aby zapewnić możliwość kompensacji zakłóceń podczas sterowania, planowanie wykonano z ograniczeniem  $\kappa_b = 3 \text{ m}^{-1}$ . Przy rozpoczęciu każdego segmentu ruchu wykonywano prostą procedurę replanowania polegającą na wyznaczeniu nowych parametrów  $(\sigma_i, \mu_i)$  zgodnie z krokiem K2 i sprawdzeniu przynależności do zbioru  $Q_a$ . Wyniki eksperymentów widoczne są na Rys. 2 - 3. Konfigurację początkową  $\mathbf{q}_0$ , konfigurację końcową  $\mathbf{q}_d$  i zaplanowane punkty przejazdowe oznaczono trójkątami. W przypadku scenariusza S1 przyjęto  $\mathbf{q}_0 = [-0.75\pi \ 0.785 \ 0.2]^\top$  i  $\mathbf{q}_d = [-0.75\pi \ 7.77 \ 1.95]^\top$ , natomiast dla scenariusza S2 przyjęto  $\mathbf{q}_0 = [-0.75\pi \ 1.17 \ 0.75]^\top$  i  $\mathbf{q}_d = [-0.55\pi \ 6.7 \ 2.35]^\top$ . Krzywiznę ruchu przedstawiono w funkcji długości ścieżki robota, przez co niektóre jej przebiegi są przesunięte w fazie na skutek różnych długości ścieżek zaplanowanych i faktycznych. W obu proponowanych scenariuszach zadanie ruchu zostało wykonane zgodnie z definicją. Wyniki eksperymentów pokazują odporność proponowanego rozwiązania na rozmaite zakłócenia pojawiające się podczas realizacji zaplanowanego ruchu.

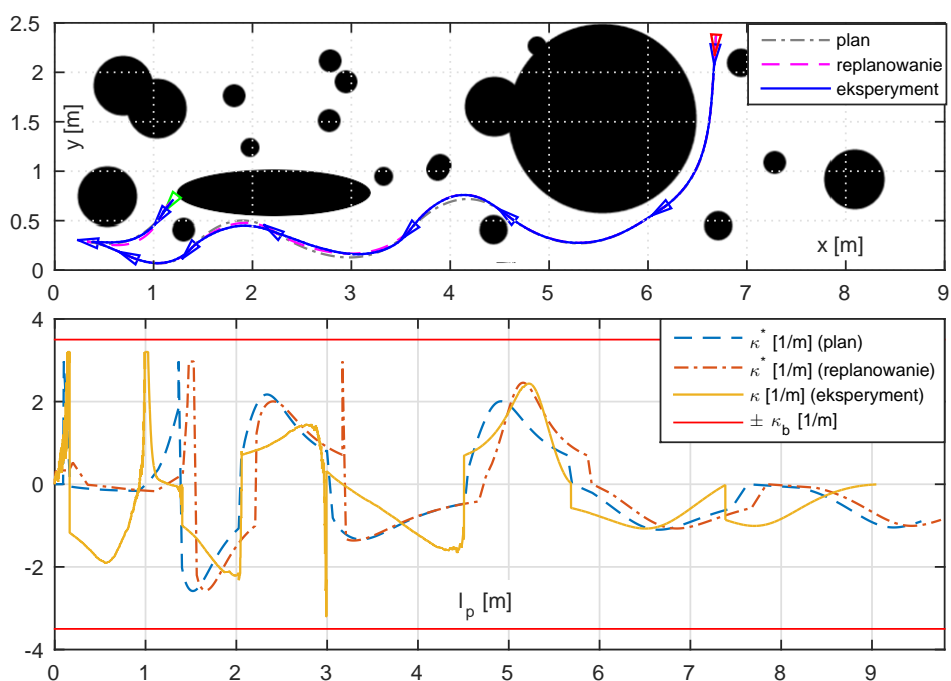
## 6. Podsumowanie

Proponowana PP VFO wydaje się być użytecznym komponentem próbkujących algorytmów planowania, który zapewnia wydajne generowanie i realizację planów ruchu w trudnych praktycznych scenariuszach. Dzięki wykorzystaniu analizy sterowania VFO wygenerowany plan nie musi być dodatkowo przetwarzany przed wykonaniem go poprzez sterowanie ze sprzężeniem zwrotnym. Proponowane rozwiązanie można rozszerzyć na inne klasy robotów mobilnych takie jak roboty o kinematyce samocho-





Rys. 2: S1: Planowanie algorytmem RRT\* i realizacja przez sterowanie VFO



Rys. 3: S2: Planowanie algorytmem RRT\* i realizacja przez sterowanie VFO

dowej, czy też roboty o kinematyce N-przyczepowej wykorzystując podejście kaskadowe. Problemem w praktycznej implementacji wspomnianego rozszerzenia może być nieciągłość sygnałów sterujących występująca w chwilach przełączenia sterownika VFO pomiędzy realizacją kolejnych punktów przejazdowych. Autorzy aktualnie pracują nad usunięciem tej nieciągłości.

## LITERATURA

- [1] D.C. Conner, H. Choset, A.A. Rizzi. Integrating planning and control for single-bodied wheeled mobile robots. *Auton. Robots*, 2011, wolumen 30, numer 3, s. 243–264.
- [2] T. Gawron, M. M. Michałek. VFO stabilization of a unicycle robot with bounded curvature of motion. In: 2015 Int. Workshop on Robot Motion and Control. *Proceedings*, Poznan, Poland, July, 2015, s. 263–268.
- [3] S. Karaman, E. Frazzoli. Sampling-based optimal motion planning for non-holonomic dynamical systems. In: 2013 IEEE Int. Conf. Robotics and Automation. *Proceedings*, Karlsruhe, Germany, May, 2013, s. 5041–5047.
- [4] M. Michałek, K. Kozłowski. Motion planning and feedback control for a unicycle in a way point following task: the VFO approach. *Int. J. Appl. Math. Comput. Sci.*, 2009, wolumen 19, numer 4, s. 533–545.
- [5] L. Palmieri, K.O. Arras. A novel RRT extend function for efficient and smooth mobile robot motion planning. In: 2014 IEEE/RSJ Int. Conf. Intelligent Robots and Systems. *Proceedings*, Chicago, IL, Sept, 2014, s. 205–211.
- [6] L. Palmieri, K.O. Arras. Distance metric learning for RRT-based motion planning with constant-time inference. In: ICRA 2015, Seattle, Washington, USA. *Proceedings*, May, 2015, s. 637–643.
- [7] A. Perez et al. LQR-RRT\*: Optimal sampling-based motion planning with automatically derived extension heuristics. In: 2012 IEEE Int. Conf. Robotics and Automation. *Proceedings*, Saint Paul, MN, May, 2012, s. 2537–2542.
- [8] A. Piazzzi, C.G.L. Bianco, M. Romano.  $\eta^3$  splines for the smooth path generation of wheeled mobile robots. *IEEE Trans. on Robotics*, Oct, 2007, wolumen 23, numer 5, s. 1089–1095.
- [9] M. Pivtoraiko, R. A. Knepper, A. Kelly. Differentially constrained mobile robot motion planning in state lattices. *Journal of Field Robotics*, 2009, wolumen 26, numer 3, s. 308–333.
- [10] K. Yang et al. Spline-based RRT path planner for non-holonomic robots. *J. Intell. Robot. Syst.*, 2014, wolumen 73, numer 1-4, s. 763–782.

### VFO EXTEND PROCEDURE FOR SAMPLING-BASED MOTION PLANNING OF MOBILE ROBOTS

In this paper we develop an extend procedure for sampling based motion planners, which is driven by the VFO (Vector-Field-Orientation) controller for a unicycle with bounded curvature of motion. The key feature of the proposed extend procedure is utilization of specific properties of VFO control strategy, which enabled us to develop the efficient analytic collision-checking and constraint-satisfaction-checking procedures. We present experimental results of motion planning and its practical execution with a laboratory-scale mobile robot.