Multiobjective Dynamic Constrained Evolutionary Algorithm for Control of a Multi-segment Articulated Manipulator

Krzysztof Michalak[1)], Patryk Filipiak[2)], and Piotr Lipiński[2)]

1) Department of Information Technologies,
Institute of Business Informatics,
Wrocław University of Economics, Wrocław, Poland
krzysztof.michalak@ue.wroc.pl

Computational Intelligence Research Group,
Institute of Computer Science,
University of Wrocław, Wrocław, Poland
{patryk.filipiak,piotr.lipinski}@cs.ii.uni.wroc.pl

Presentation Plan

- The Inverse Kinematics Problem
  - basic IK problem
  - the dynamic version tackled in this paper
- The Evolutionary Algorithm
  - objectives and constraints
  - operators
- Experiments
- Results & Conclusion

The Inverse Kinematics Problem

- Inverse Kinematics (IK)
  - articulated robotic arm with a preset number of segments
  - desired position of the end effector
  - control of the angles between segments

- Problem tackled in this paper
  - constrained (with obstacles)
  - dynamic (the obstacles move)
  - we model a sequence of movements of the manipulator

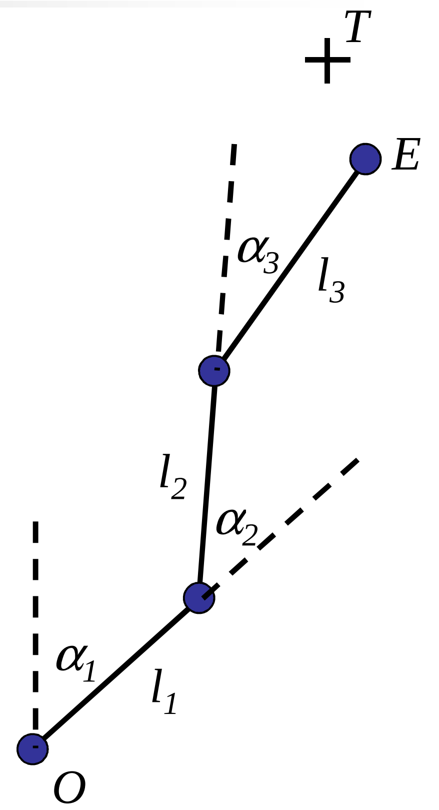
Formal Definitions

- Basic IK problem
  - manipulator attached at point O
  - list of segment lengths $\{ l_1, \ldots, l_{N_S} \}$
  - manipulator position = a list of relative angles $\{ \alpha_1, \ldots, \alpha_{N_S} \}$

    $\alpha_i = 0$ represents a segment pointing in the same direction as the previous one
  - manipulator endpoint E

$$[x_E, y_E] = [x_O, y_O] + \sum_{i=1}^{N} \left[l_j \cos\left(\sum_{j=1}^{i} \alpha_j \right), l_j \sin\left(\sum_{j=1}^{i} \alpha_j \right) \right]$$

  - target point T



Formal Definitions

- Movements of the manipulator
  - movement of the manipulator is defined by values of the angles between manipulator segments at discrete time instants

$$\{\alpha_1(t), \ldots, \alpha_{N_s}(t)\}_{t=0,\ldots,N_t}$$

  - a movement between time instants t and $t + 1$ is performed as a linear change of all the angles

$$\alpha_j(t+\delta) = \alpha_j(t) \cdot (1-\delta) + \alpha_j(t+1) \cdot \delta$$
$$j = 1, \ldots, N_s,$$
$$\delta \in [0, 1].$$

Formal Definitions

- Obstacles
  - a set of N_O obstacles
  - each obstacle O_i is a convex polygon with M_i vertices
  - positions of the vertices change in time

- Optimization problem
  - at each time instant $t = 1, \ldots, N_t - 1$ find a set of angles for time instant t + 1
  - minimize the average distance between the manipulator endpoint E and the target point T for $t = 1, \ldots, N_t$
  - do not intersect any of the obstacles

The Evolutionary Algorithm

- Evolution performed for each time step t with the goal of finding a position at time step $t+1$
- Specimen represents the angles at $t+1$
- Objectives and constrains are calculated based on simulated transition between t and $t+1$
- Contains both the elements typical to evolutionary algorithms used for:
  - Constrained optimization: mechanisms used in the Infeasibility Driven Evolutionary Algorithm (IDEA)
    - violation measure used as one of the objectives
    - a fraction of the population is reserved for infeasible specimens
  - Dynamic optimization:
    - random immigrants added to every generation

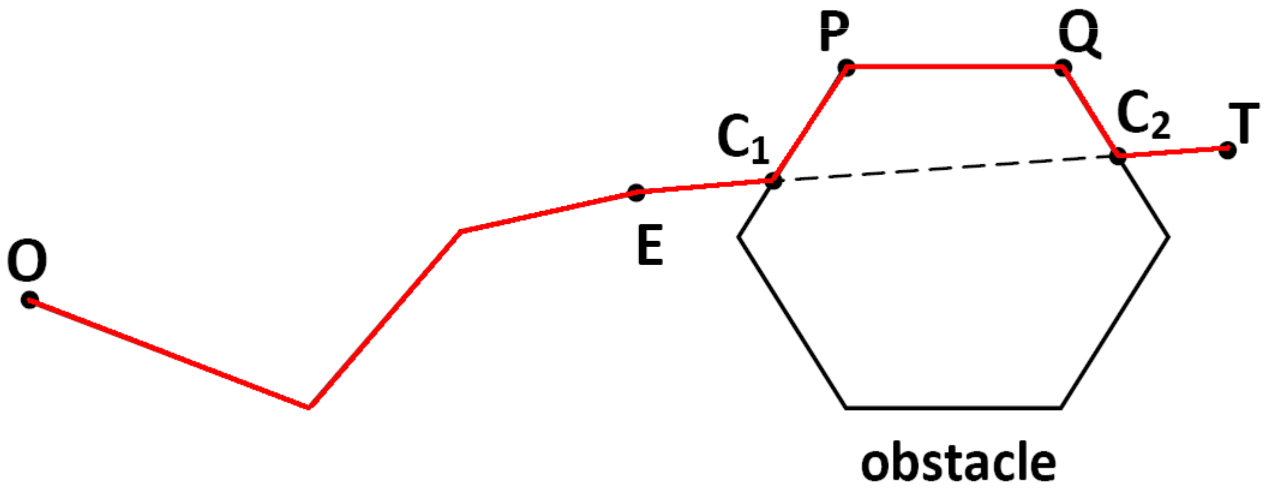
The Evolutionary Algorithm

- Objectives and constraints
  - three objectives: f_1, f_2 and f_3
  - two constraints: g_1 and g_2
    - both g_1 and g_2 have to be 0 for the specimen to be feasible
    - infeasible specimens have $g_1 > 0$ or $g_2 > 0$
  - calculated for N_δ simulation steps from time t *to* $t + 1$
  - n-th step ($n = 1, ..., N_\delta$) corresponds to time:

$$t + \frac{n-1}{N_\delta - 1}$$

The Evolutionary Algorithm

- Objectives and constraints
  - f_1 : the distance between the manipulator endpoint *E* and the target point T
    - no obstacles between *E* and *T* : the Euclidean distance is used
    - there are obstacles between *E* and *T* :



    - values for all simulation steps $n = 1, ..., N_\delta$ are averaged with weights equal to n / N_δ

The Evolutionary Algorithm

- Objectives and constraints
  - f_2: a measure of displacement of the manipulator between time instants t and $t + 1$

$$f_2 = \sum_{k=1}^{N_s} \left[(x_{j_k}(t+1) - x_{j_k}(t))^2 + (y_{j_k}(t+1) - y_{j_k}(t))^2 \right]$$

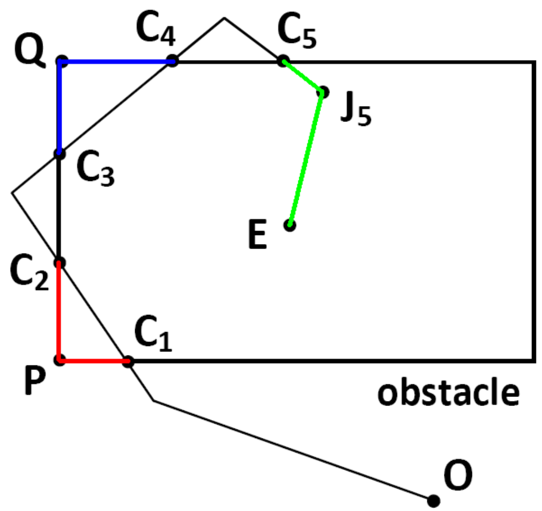
  where:

  $x_{j_k}(t)$, $y_{j_k}(t)$ - the coordinates of the k–th joint of the manipulator calculated for angles at the time instant t

  - f_3: violation measure introduced in the IDEA algorithm

The Evolutionary Algorithm

- Objectives and constraints
  - g_1 : a measure of intersection with obstacles
    - for each pair of intersection points C_1, C_2 the length of the shorter of the paths connecting C_1 and C_2 on the circumference of the obstacle is added to g_1
    - if the endpoint of the manipulator is inside the obstacle the length of the arm inside the obstacle is added.



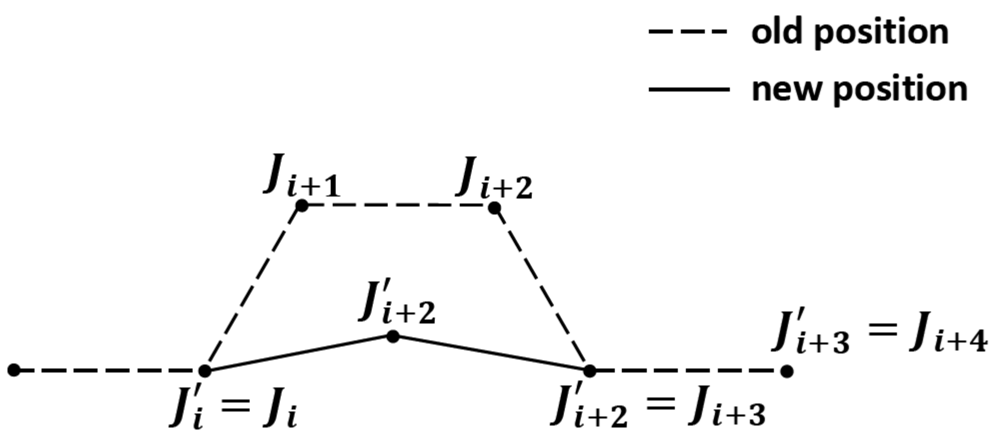
$$g_1 = d(C_1, P) + d(P, C_2) + d(C_3, Q) + d(Q, C_4) + d(C_5, J_5) + d(J_5, E)$$

The Evolutionary Algorithm

- Objectives and constraints
  - g_2: a measure of self-intersections of the manipulator
    - The sum of $1 / (j \cdot k)$ for those j and k for which the manipulator segments J_jJ_{j+1} and J_kJ_{k+1} intersect
  - values of both constraints are summed for all simulations steps $n = 1, \ldots, N_\delta$

The Evolutionary Algorithm

- Operators
  - SBX crossover
  - polynomial mutation
  - three dedicated operators
    - single joint mutation: polynomial mutation at one joint + correction for the next joint
    - the Unfold-3 operator: intended to help the manipulator straighten by replacing any three segments by two if possible



The Evolutionary Algorithm

- Operators
  - three dedicated operators (cont.)
    - the RepairSelfIntersections operator: tries to untangle the manipulator if self-intersections are present

First, a set of intersecting pairs of manipulator segments is identified:

$$I = \left\{ \langle j,k \rangle : \overline{J_j J_{j+1}} \text{ and } \overline{J_k J_{k+1}} \text{ intersect } \right\} \quad . \tag{1}$$

Based on the set I the first and the last joint in the entangled part of the manipulator are identified:

$$j_f = min \{ j : \exists k : \langle j,k \rangle \in I \} + 1 \quad , \tag{2}$$

$$j_l = max \{ k : \exists j : \langle j,k \rangle \in I \} \quad . \tag{3}$$

One index j_{fix} is randomly selected from the range $j_f, \ldots, j_l$ with uniform probability. The angle $\alpha_{j_{fix}}$ is modified by setting $\alpha_{j_{fix}} = \eta \alpha_{j_{fix}}$, where η is a random number drawn from the $U[0,1]$ distribution. This makes the segment $\overline{J_{j_{fix}} J_{j_{fix}+1}}$ closer to pointing straight with respect to the previous segment $\overline{J_{j_{fix}-1} J_{j_{fix}}}$.

The Evolutionary Algorithm

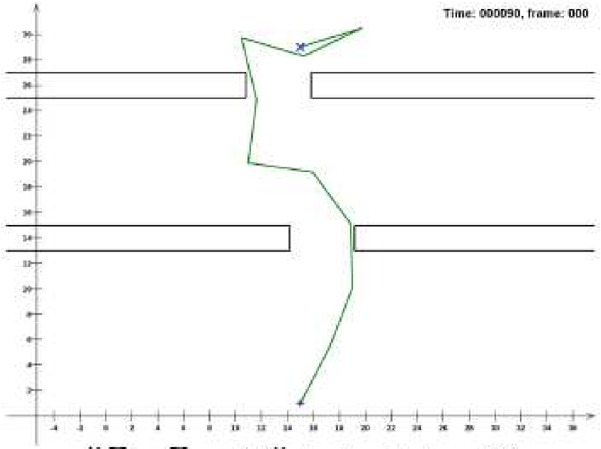
- The main loop

Proceedings, page 203

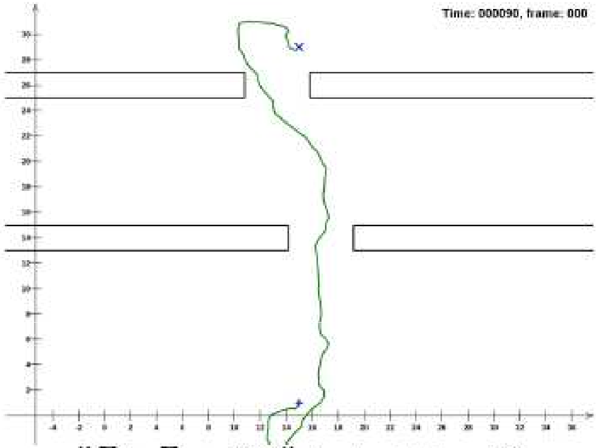
for $g = 1 \to N_{gen}$ **do**
  $P_{offspring} = \emptyset$
  $P_{mate} = \text{SelectMatingPool}(P)$
  for $i = 1 \to N_{pop}/2$ **do**
    $\langle O_1, O_2 \rangle = \text{Crossover}(P_{mate}[2*i-1],\ P_{mate}[2*i])$
    $\text{Mutate}(O_1); \quad \text{Mutate}(O_2)$
    $\text{MutateOneJoint}(O_1); \quad \text{MutateOneJoint}(O_2)$
    $\text{RepairSelfIntersections}(O_1); \quad \text{RepairSelfIntersections}(O_2)$
    $P_{offspring} = P_{offspring} \cup \{O_1, O_2,\}$
  end for
  $P_{rnd} = \text{InitPopulation}(N_{rnd})$
  $\text{RepairSelfIntersections}(P_{rnd})$
  $P_{U3} = \text{Unfold-3}(P_{offspring}) \cup \text{Unfold-3}(P_{rnd})$
  $P_{av} = \text{AvoidObstacles}(P_{offspring}) \cup \text{AvoidObstacles}(P_{rnd}) \cup \text{AvoidObstacles}(P_{U3})$
  $\text{RepairSelfIntersections}(P_{av})$
  $\text{Evaluate}(P_{offspring} \cup P_{rnd} \cup P_{U3} \cup P_{av})$
  $P = P \cup P_{offspring} \cup P_{rnd} \cup P_{U3} \cup P_{av}$
  $\langle P_f, P_{inf} = Split(P)$
  $\text{Rank}(P_f); \quad \text{Rank}(P_{inf})$
  $P = P_{inf}[1 : N_{inf}] \cup P_f[1 : N_f]$
end for

Experiments

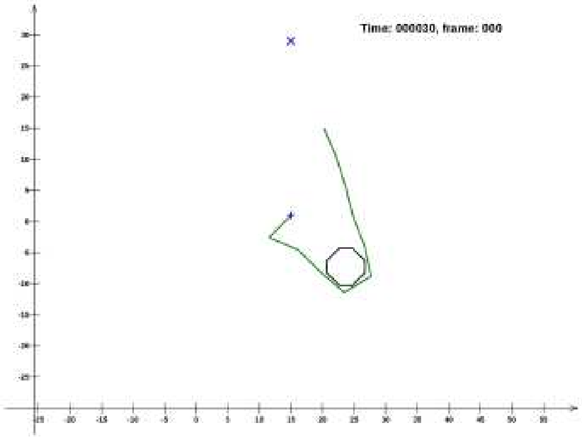
- 4 scenarios based on 3 obstacle courses



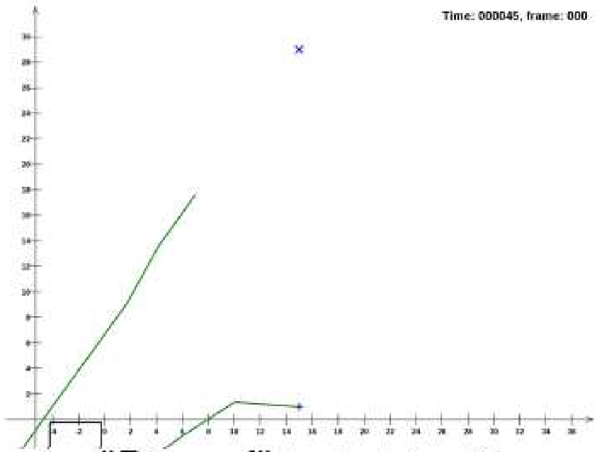
"Zig-Zag 10" test at $t = 90$



"Zig-Zag 100" test at $t = 90$



"Circular" test at $t = 30$



"Diagonal" test at $t = 45$

Experiments

- Manipulators
  - 10 segments of length 5
  - 100 segments of length 0.5
- Parameter settings
  - the number of time instants: $N_t = 100$
  - the number of generations: $N_{gen} = 50, 100, 200$
  - population sizes: $N_{pop} = 50, 100, 200$
  - 10 repetitions of the test for each setting
- Performance evaluation
  - the average Euclidean distance $d(E, T)$
  - the fraction q of time instants during which $d(E, T) < 0.5$

Results & Conclusion

- Numerical evaluation $\Rightarrow$ proceedings, page 205
- Observations from the tests
  - comparison for 10 and 100 segments
    - $d(E, T)$ obtained for the ″Zig-Zag 100″ test is very similar to that obtained for the ″Zig-Zag 10″ test
    - the fraction q of time instants during which $d(E, T) < 0.5$ is higher for the ″Zig-Zag 100″ test
    - increasing the values of the N_{pop} and N_{gen} parameters improves the results significantly

Results & Conclusion

- Observations from the tests (cont.)
  - problems in which it might be beneficial for the algorithm to backtrack and ”go around” the obstacle (”Circular” and ”Diagonal”)
    - increasing the values of the N_{pop} and N_{gen} parameters provides only a moderate improvement in solution quality
    - the algorithm might benefit from planning (e.g. optimizing a sequence of moves for several time instants, not just one movement at a time)
- Video

Thank you!