Design of Powered Floor Systems for Mobile Robots with Differential Evolution

Eric Medvet, Stefano Seriani, Alberto Bartoli, Paolo Gallina

Department of Engineering and Architecture
University of Trieste
Italy

EvoApplications, 25/4/2019, Leipzig (Germany)

http://machinelearning.inginf.units.it
https://erallab.inginf.units.it

Mobile robots

Many key applications, often in bounded environments

- internal logistics
- swarm robotics

Power supply

Battery, has to be recharged ! long lasting operations?

Alternatives:

- ground + aerial (pantograph)
- wireless energy delivery
- **powered oors** and sliding contacts

EVOLUTIONARY ROBOTICS ARTIFICIAL LIFE LAB

Power supply

Battery, has to be recharged ! long lasting operations?

Alternatives:

- ground + aerial (pantograph)
- wireless energy delivery
- **powered oors** and sliding contacts

Designing a powered floor system

Choose:

- conductive/insulating strips width
- sliding contacts on robot: number and position

given some constraints

Designing a powered oor system

Choose:

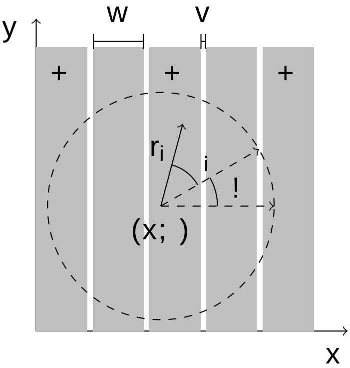
- conductive/insulating strips width
- sliding contacts on robot: number and position

given some constraints

Can we do it automatically?

Table of Contents

Goal



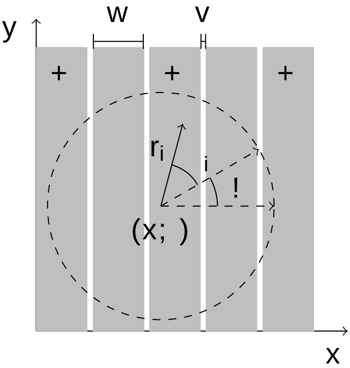
Given:

- conductive (w) and insulating (v) strips width
- feasible region on the robot (as g())
- number n of sliding contacts

nd:

- position $A = f(r_1;\ _1); : : : ; (r_n;\ _n)g$ of the sliding contacts array which ensure power supply in any robot position and rotation

Goal



Given:

- conductive (w) and insulating (v) strips width
- feasible region on the robot (as g())
- number n of sliding contacts

nd:

- position $A = f(r_1; \ _1); : : : ; (r_n; \ _n)g$ of the sliding contacts array which ensure power supply in any robot position and rotation

Powered condition

\ensures power supply in any robot position and rotation"

#

powered condition

How to model?

- one sliding contact in touch for xed position
- at least one positive and one negative in touch for xed position
- at least one positive and one negative in touch for any position

One sliding contact, xed

One sliding contact $r;\ $ touches + with robot in $x;!$:

$$f_c^+(r;\ ;\ x;!) = \begin{cases} 1 & \text{if } (x + r\cos(\ + !)) \bmod (2w + 2v) \quad w \\ 0 & \text{otherwise} \end{cases}$$

Similarly for $f_c\ (r;\ ;\ x;!)$

+ and , xed position

At least one positive for xed position:

$$\sum_{i=1}^{i=n} \overbrace{f_c^{+}(r_i;\ _i; x; !)}^{\text{touch}} \overbrace{g(r_i;\ _i)}^{\text{feasible}} \quad 1$$

+ and , xed position

At least one positive and one negative for xed position:

$$\sum_{i=1}^{i=n} \overbrace{f_c^+(r_i;\ _i;x;!)}^{\text{touch}}\ \overbrace{g(r_i;\ _i)}^{\text{feasible}} \quad 1$$

$$\sum_{i=1}^{i=n} f_c\ (r_i;\ _i;x;!)\, g(r_i;\ _i) \quad 1$$

+ and , xed position

At least one positive and one negative for xed position:

$$f^{+}(A; x; !) = \sum_{i=1}^{i=n} \overbrace{f_c^{+}(r_i; \;_i; x; !)}^{\text{touch}} \overbrace{g(r_i; \;_i)}^{\text{feasible}} \quad 1$$

$$f \quad (A; x; !) = \sum_{i=1}^{i=n} f_c \; (r_i; \;_i; x; !) g(r_i; \;_i) \quad 1$$

Or, simply:

$$f^{+} \quad (A; x; !) = \min \; f^{+}(A; x; !) ; f \quad (A; x; !) \quad 1$$

f^{+} $(A; x; !)$ is the number of contact pairs in touch

+ and , any position

At least one positive and one negative for any position:

$$\min_{\substack{x 2 [0;2(w+v)[\\ ! \; 2 [0;2 \quad [}} f^{+} \quad (A;x;!) \qquad 1$$

+ and , any position

At least one positive and one negative for any position:

$$f\ (A) = \min_{\substack{x \in [0;2(w+v)[\\ ! \in [0;2\ [}} f^{+}\ (A;x;!)\quad 1$$

f (A) is the number of contact pairs in touch in the worst possible condition x; !

1 satis es the powered condition

+ and , any position

At least one positive and one negative for any position:

$$f(A) = \min_{\substack{x \in [0;2(w+v)[\\ ! \in [0;2 [}} f^{+} (A; x; !) \quad 1$$

f (A) is the number of contact pairs in touch in the worst possible condition x; !

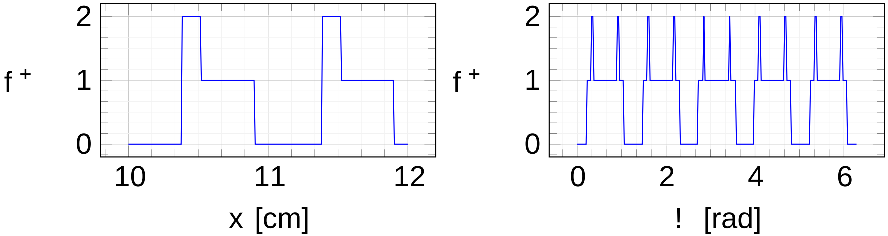
1 satis es the powered condition

and the larger, the better

) good candidate for being a tness function!

What a badf !

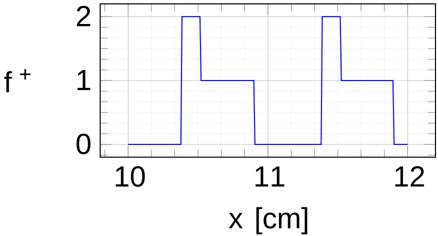
f contains a lot of bad things: min, mod, binary()!

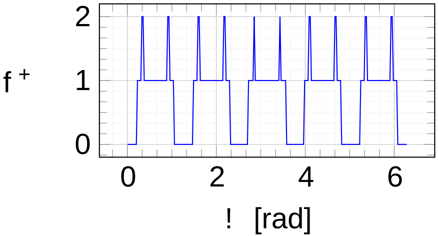


(for a simpleA of 5 equally spaced contacts on a circle)

What a badf !

f contains a lot of bad things: min, mod, binary()!





(for a simpleA of 5 equally spaced contacts on a circle)

#

f might be not able to drive the search

Smootherf

Instead of \n. of contact pairs in touch in the worst condition"

$$f\ (A) = \min_{\substack{x 2 [0;2(w+v)[\\ ! \ 2 [0;2\ \ [}} f^{+}\ \ (A; x; !\ )$$

Smoother f

Instead of "n. of contact pairs in touch in the worst condition"

$$f(A) = \min_{\substack{x \in [0;2(w+v)[\\ \omega \in [0;2\ [}} f^{+}(A;x;\omega)$$

consider "average n. of contact pairs in touch"

$$\hat{f}(A) = \frac{1}{4\ (w+v)} \sum_{\substack{x \in [0;2(w+v)[\\ \omega \in [0;2\ [}} f^{+}(A;x;\omega)\, dx\, d\omega$$

warning! f 1 meets the powered condition, $\hat{f}$ does not!

Practicality of an array

Are the sliding contacts “easily” doable?

Practicality of an array

Are the sliding contacts "easily" doable?

Maximize distance among contacts:

$$d(A) = \frac{1}{n} \sum_{i \in V} \min_{j \in V, j \neq i} \sqrt{r_i^2 + r_j^2 - 2 r_i r_j \cos(\theta_i - \theta_j)}$$

- consider only contacts in the feasible region
- average distance to closest other contact

Recap

Given an array of sliding contacts A, we can measure:

1. number of contact pairs in touch in the worst condition: $f(A)$
2. average number of contact pairs in touch: $\hat{f}(A)$
3. contacts distance $d(A)$

For all: the greater, the better

Table of Contents

EA and solution representation

Individual:

$$\mathbf{x} = e(A) = \quad \frac{r_1}{r}; \frac{\ 1}{2}; : : : ; \frac{r_n}{r}; \frac{\ n}{2}$$

- r is the largest radius in the feasible region
- $\mathbf{x}$ *2* $\mathbb{R}^{2n}$, n number of sliding contacts

Continuous optimization: we chose DE

Fitness

Three variants:

- M: only f
- MD: $f ; d$
- MAD: $f ; \hat{f} ; d$

For MO variants, lexicographical ordering

- f comes first, it is the most important

For actual computation of f, $\hat{f}$, discretization of the robot position $x; !$:

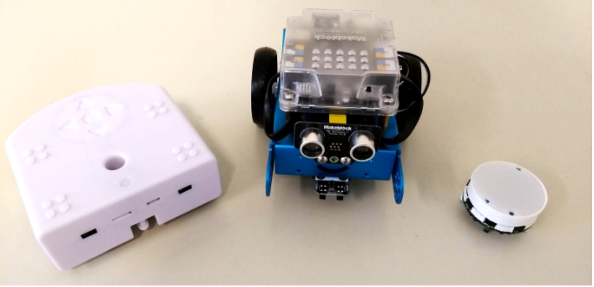
- $x\ 2\ [0; 2(w+v)[$ and $!\ 2\ [0; 2\ \ [$
- $n_{\text{point}} = 100$ each $)$ n_{point}^2

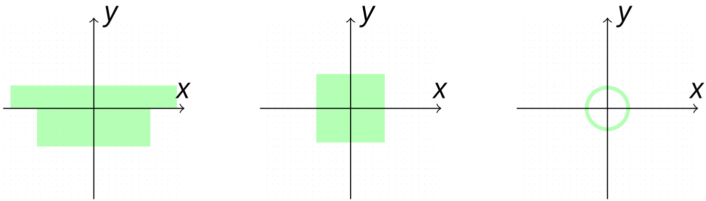
Table of Contents

Questions

1. Can f, $\hat{f}$, and d, possibly combined, drive the search?
2. Impact of parameters n, w?

Robots: Thymio II, mBot, Elisa-3



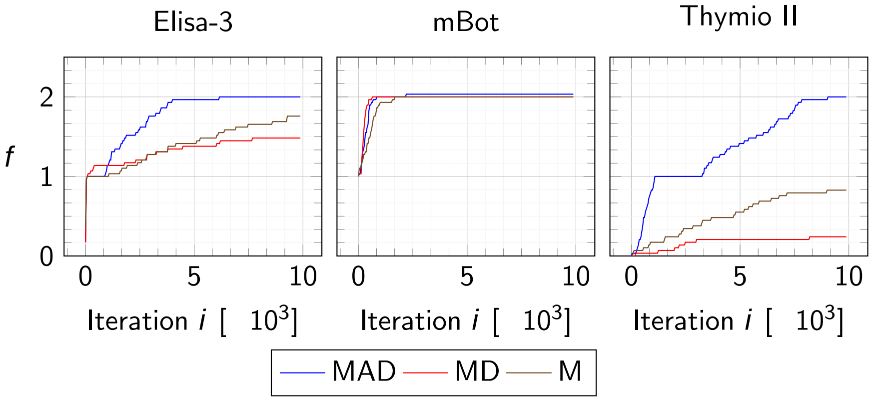


Procedure and parameters

- 30 independent runs for each problem (robot, n, w)
- 10 000 fitness evaluations per run

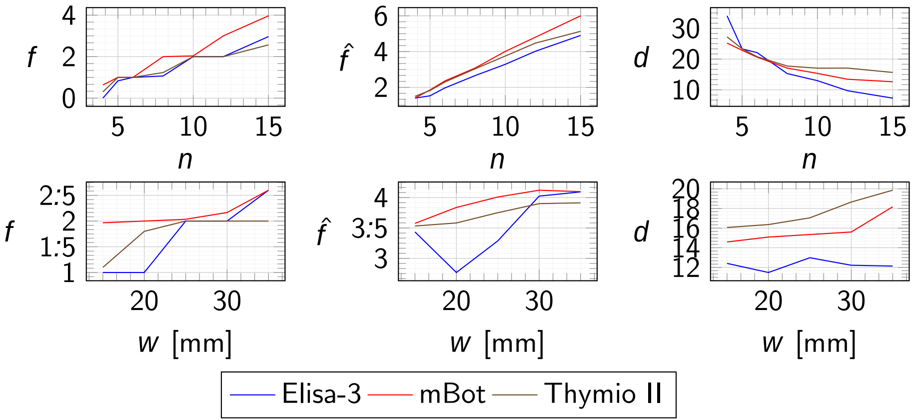
RQ1: fitness variant

$w = 25$ mm, $v = 3$ mm, and $n = 10$



- MAD is always the best
- for Thymio II, complex geometry, only MAD finds a solution!

RQ2: impact of parameters



- for n, as expected (consider cost and practicality!)
- for w, the larger, the better (v is constant)

EVOLUTIONARY ROBOTICS ARTIFICIAL LIFE LAB

Conclusion

Automatic design of sliding contact positions can be done!

- pretty fast: 80 s for one run
- with the proper fitness function

Future work:

- symmetry *!* representation
- promote balancing of + and contacts

Thanks!