Attractor dynamics approach to behavior generation on robots with low-level sensors

Gregor Schöner, INI, RUB

Second order dynamics

- source: Bicho, Schöner, Robotics and Autonomous Systems 21:23-35 (1997)
- idea:
  - dynamics not of heading direction, but of turning rate
  - target acquistion: define desired turning rate toward target (left vs. right)
  - obstacles: turn at desired rate whenever there is impending decision, based on decision left vs. right

dynamical variable

- turning rate omega
- enact by setting new set-point for velocity servo of each motor
- target: information about target being to the left, to the right, or ahead, but no calibrated bearing, psi, to target
- obstacle: turning rate
  - to the right when obstacle close and to the left
  - to the left when obstacle close and to the right
  - zero when obstacle far

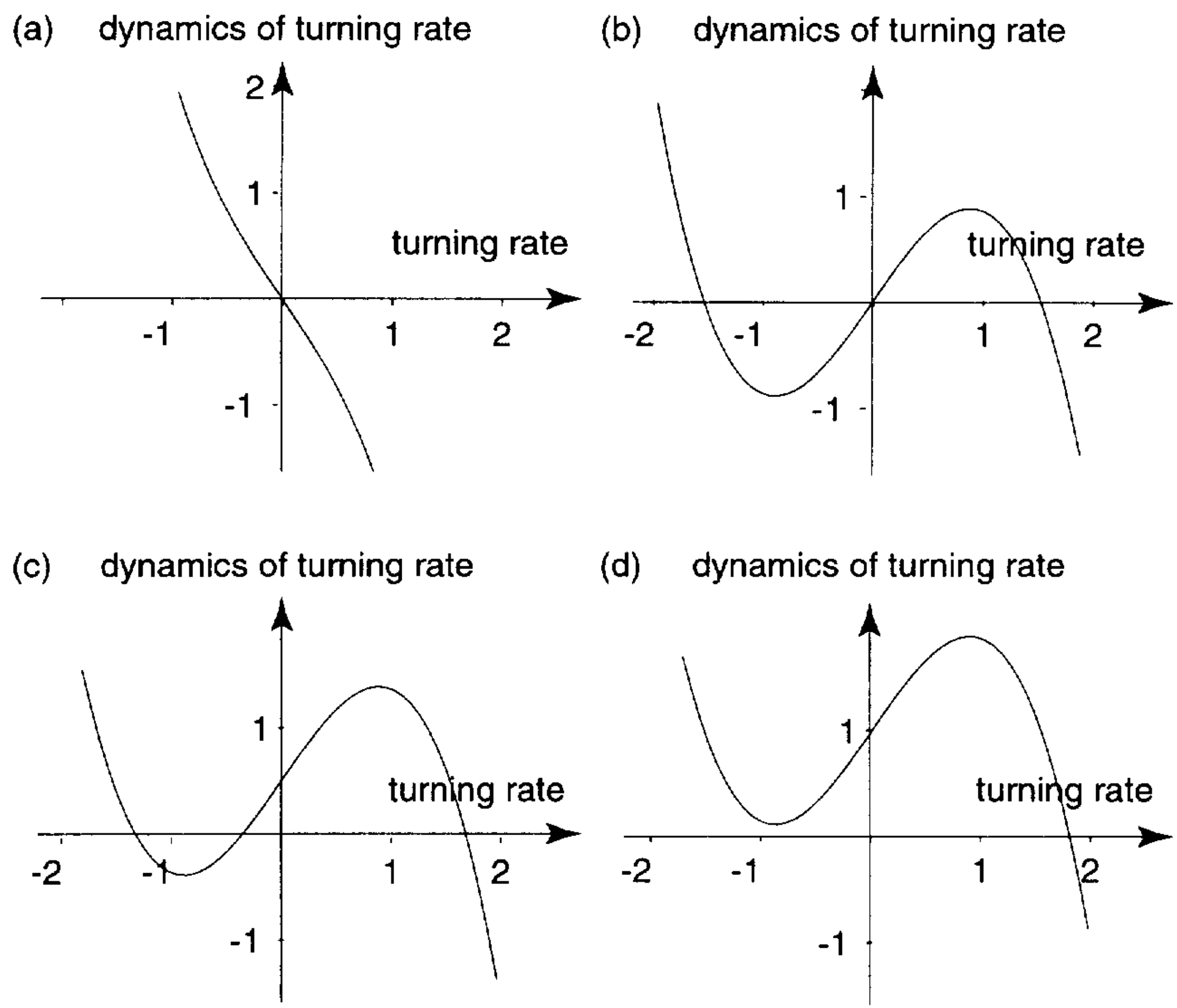
dynamics of turning rate: obstacle avoidance

- pitch-fork normal form (to get left-right symmetry)
- but symmetry potentially broken by additive constant: biases bifurcation toward left or toward right

$$\dot{\omega} = (\alpha + \tfrac{1}{2}\pi) c_{\text{obs}} F_{\text{obs}} + \alpha\omega - \gamma\omega^3$$

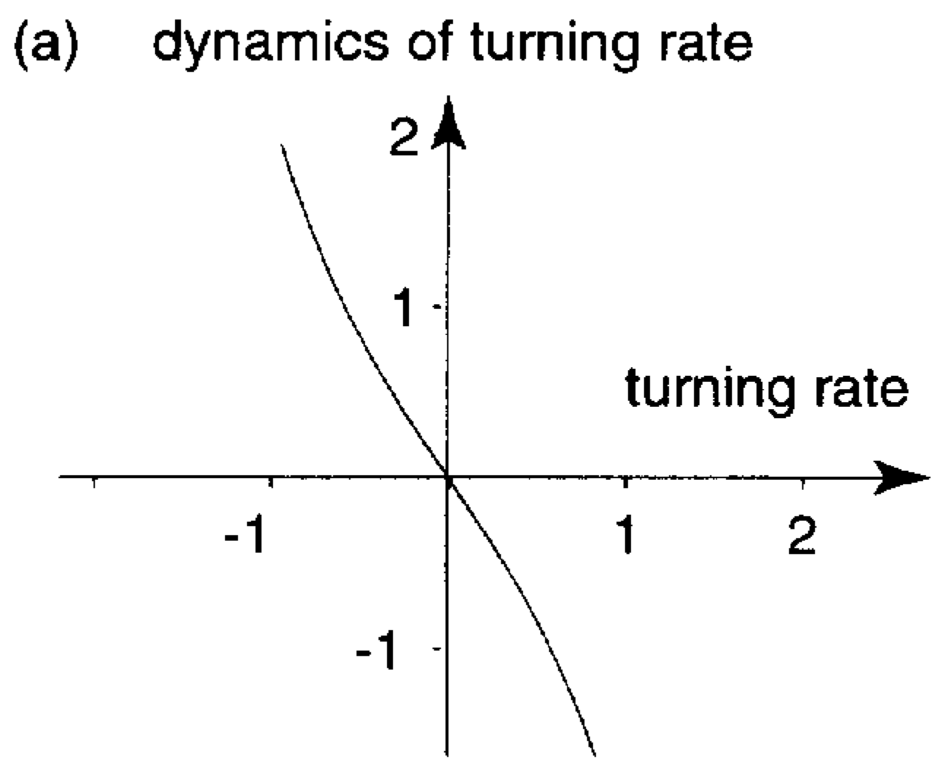
obstacle avoidance

$$\dot{\omega} = (\alpha + \tfrac{1}{2}\pi)c_{\text{obs}}F_{\text{obs}} + \alpha\omega - \gamma\omega^3$$



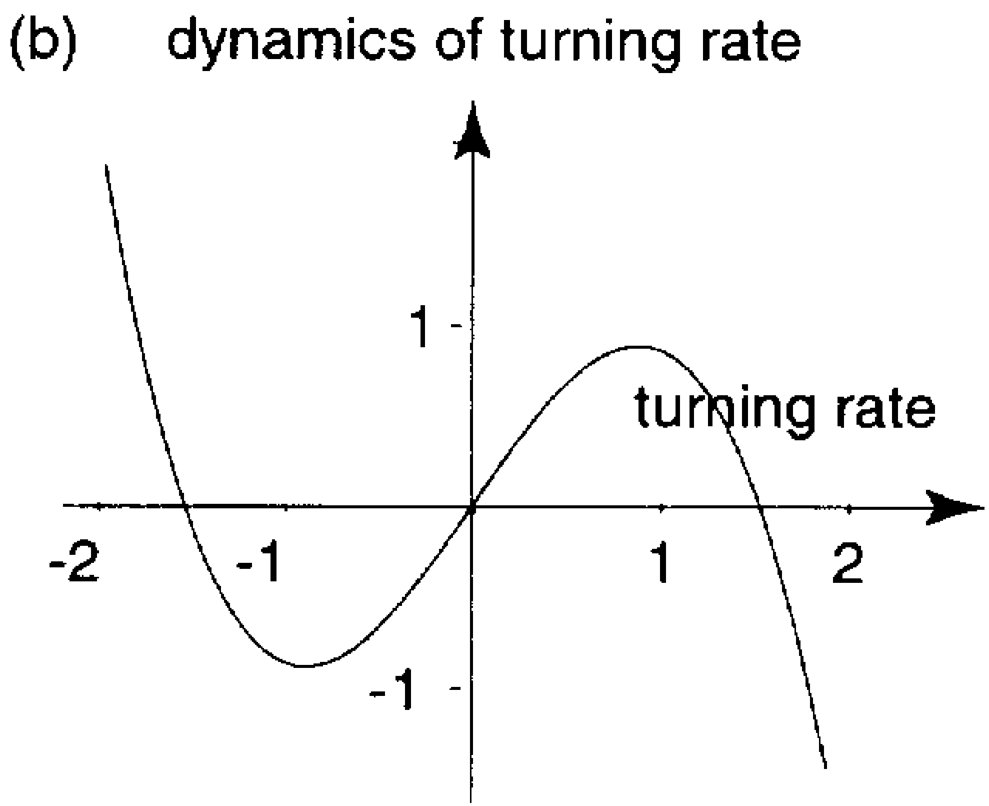
obstacle avoidance

- in absence of obstacle in forward direction (distance large): alpha negative, contant zero



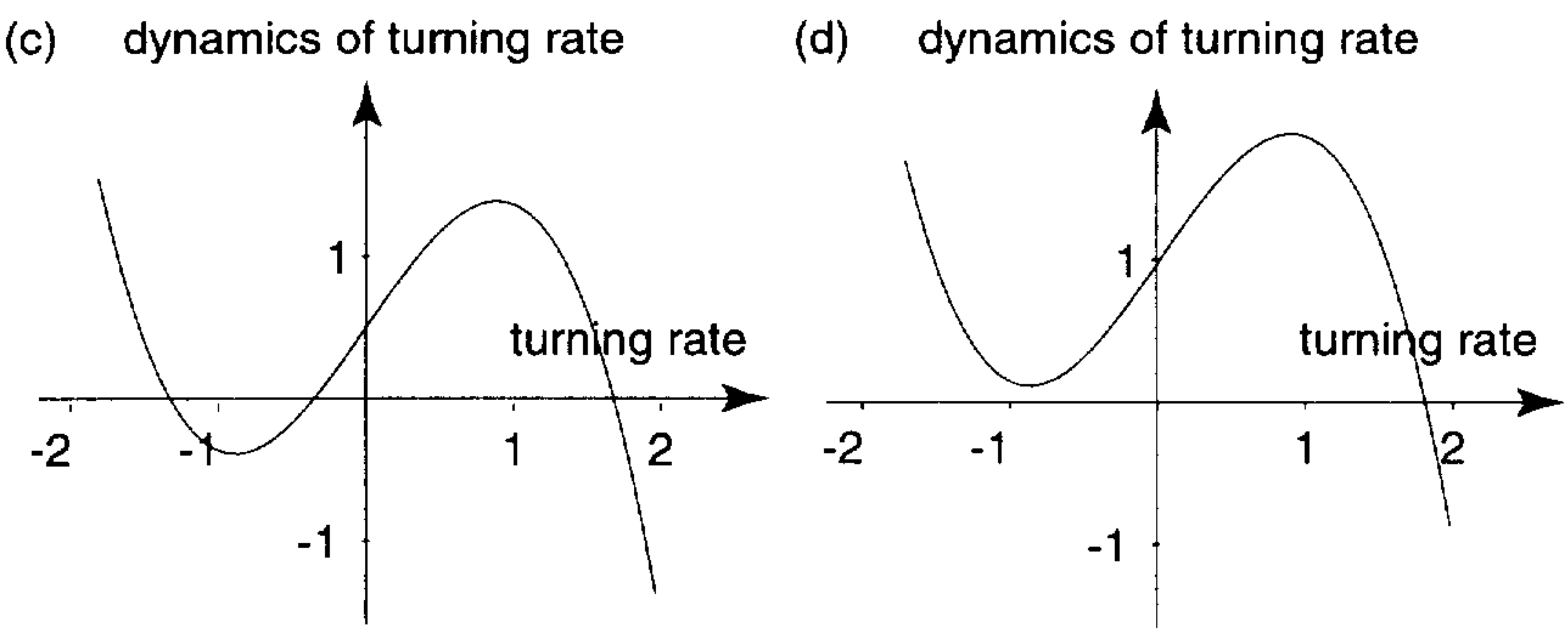
obstacle avoidance

- in presence of obstacle in forward direction, symmetric bifurcation to desired avoidance rotations: alpha positive, constant zero



obstacle avoidance

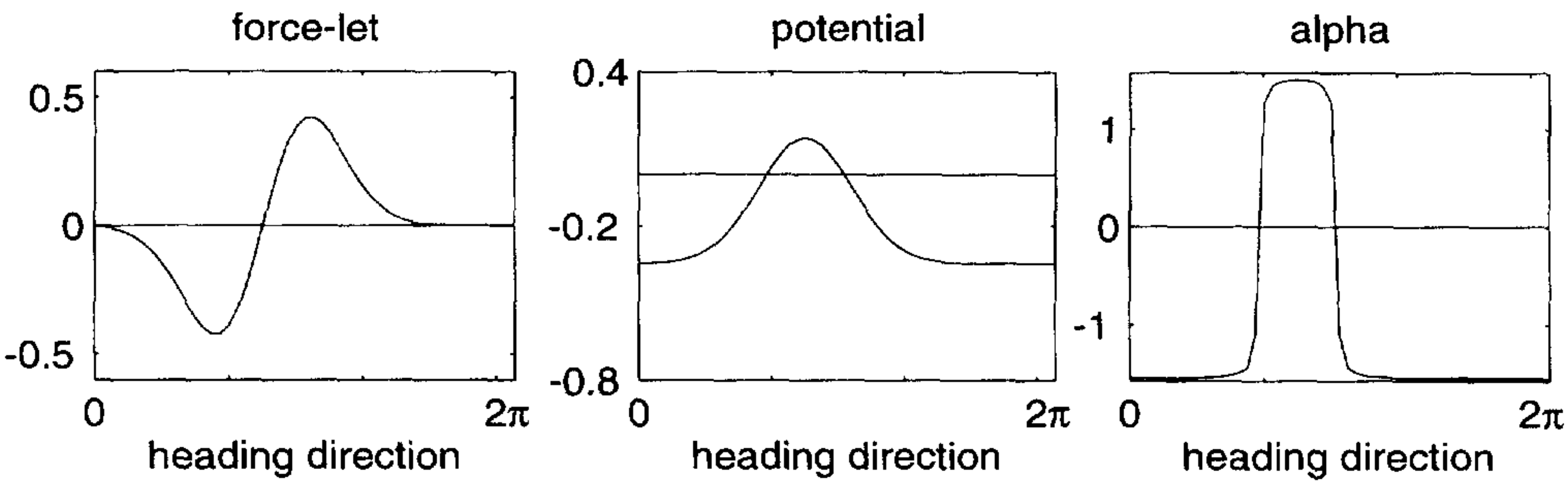
- in presence of obstacle to the right of current heading: tangent bifurcation removes attractor at negative omega, alpha negative, constant negative



mathematical form

- compute constant and alpha from obstacle

$$\dot{\omega} = (\alpha + \tfrac{1}{2}\pi) c_{\text{obs}} F_{\text{obs}} + \alpha\omega - \gamma\omega^3$$



$$F_{\text{obs}} = \sum_i \lambda_i (\phi - \psi_i) \exp\left[-\frac{(\phi - \psi_i)^2}{2\sigma_i^2}\right]$$

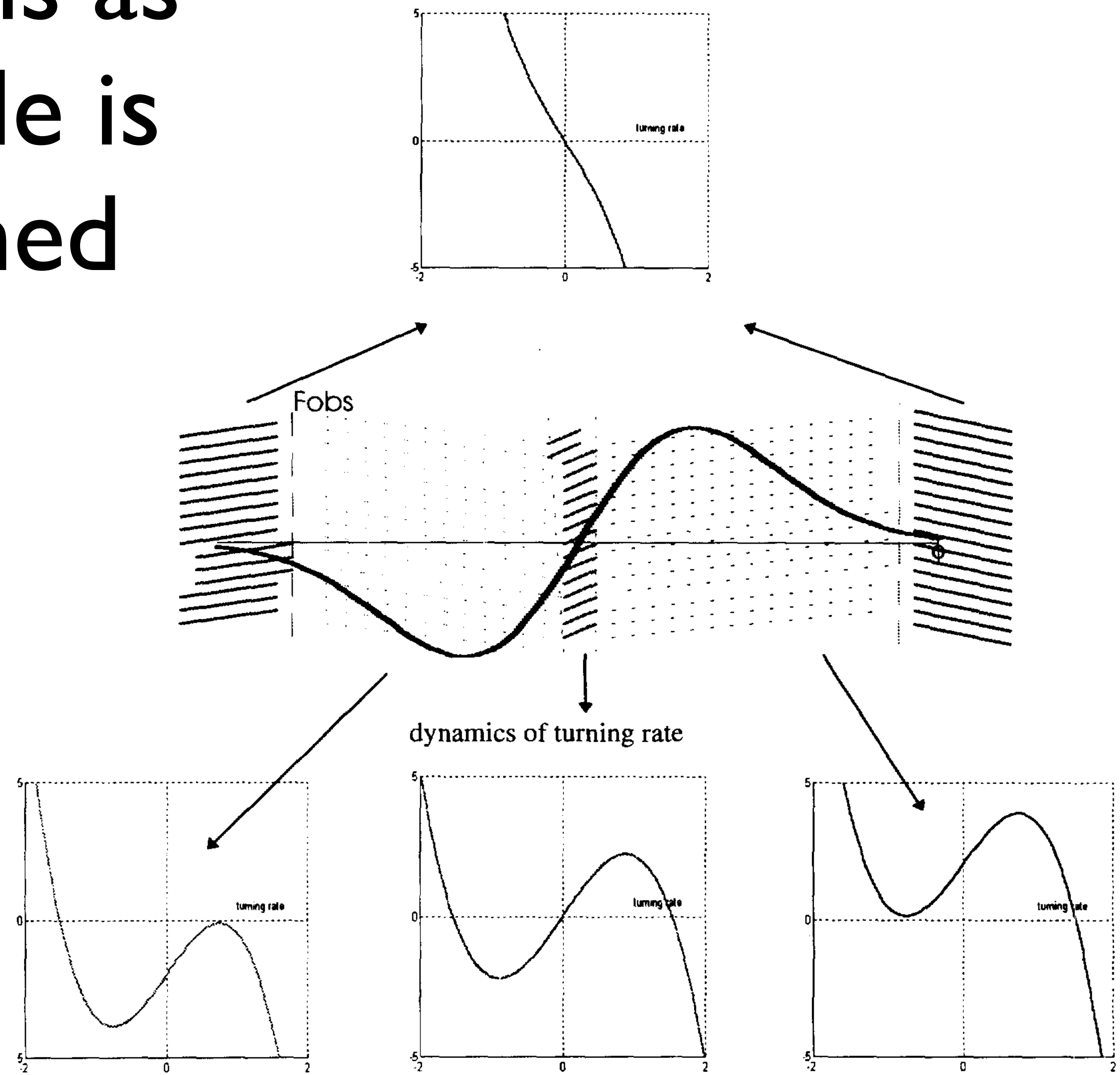
$$\lambda_i = \beta_1 \exp[-d_i / \beta_2]$$

$$\sigma_i = \arctan\left[\tan\left(\frac{\Delta\theta}{2}\right) + \frac{R_{\text{robot}}}{R_{\text{robot}} + d_i}\right]$$

$$V = \sum_i \left(\lambda_i \sigma_i^2 \exp\left[-\frac{\theta_i^2}{2\sigma_i^2}\right] - \frac{\lambda_i \sigma_i^2}{\sqrt{e}}\right)$$

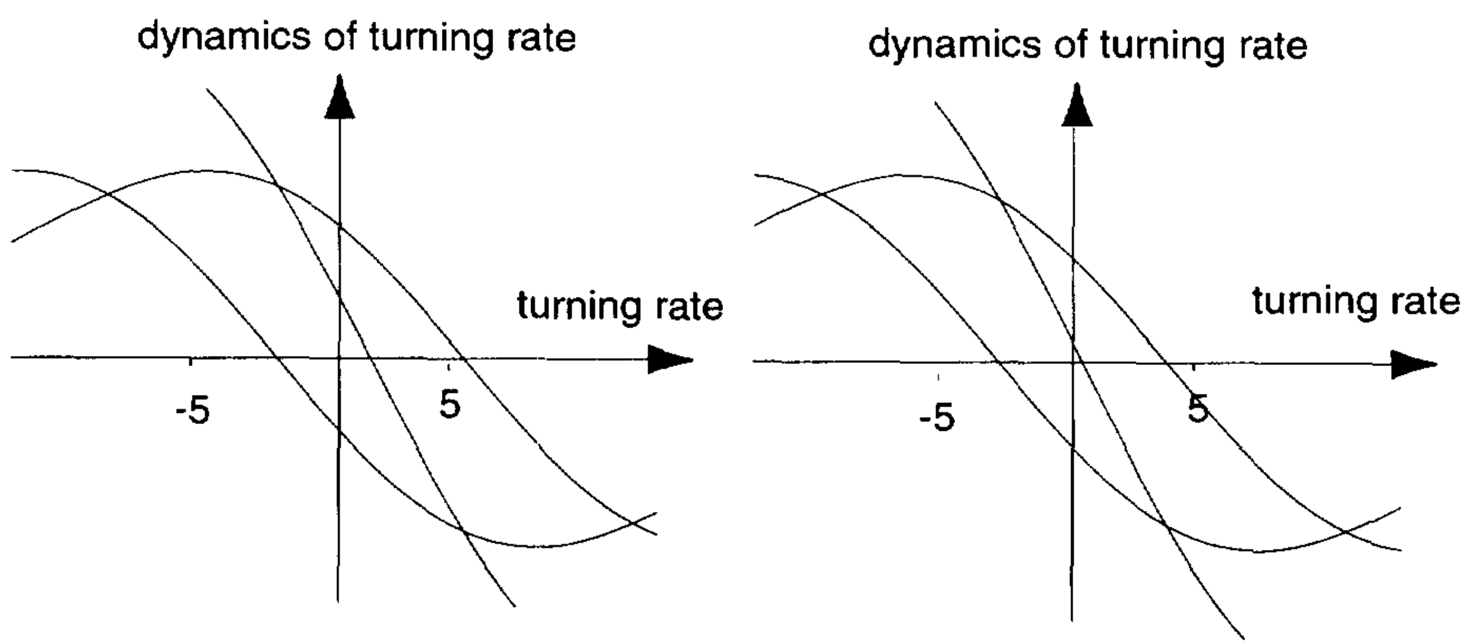
$$\alpha = \arctan[c \;\; V]$$

bifurcations as an obstacle is approached



dynamics: target acquisition

- a sensor for a target on the left sets an attractor at positive turning rate, strength graded with intensity
- a sensor for a target on the right sets an attractor at negative turning rate, strength graded with intensity



mathematical formulation

- force-let of each target sensor

$$g_i(\omega) = -\frac{1}{\tau_\omega}(\omega - \omega_i)\exp\left[-2\frac{(\omega - \omega_i)^2}{\Delta\omega^2}\right].$$

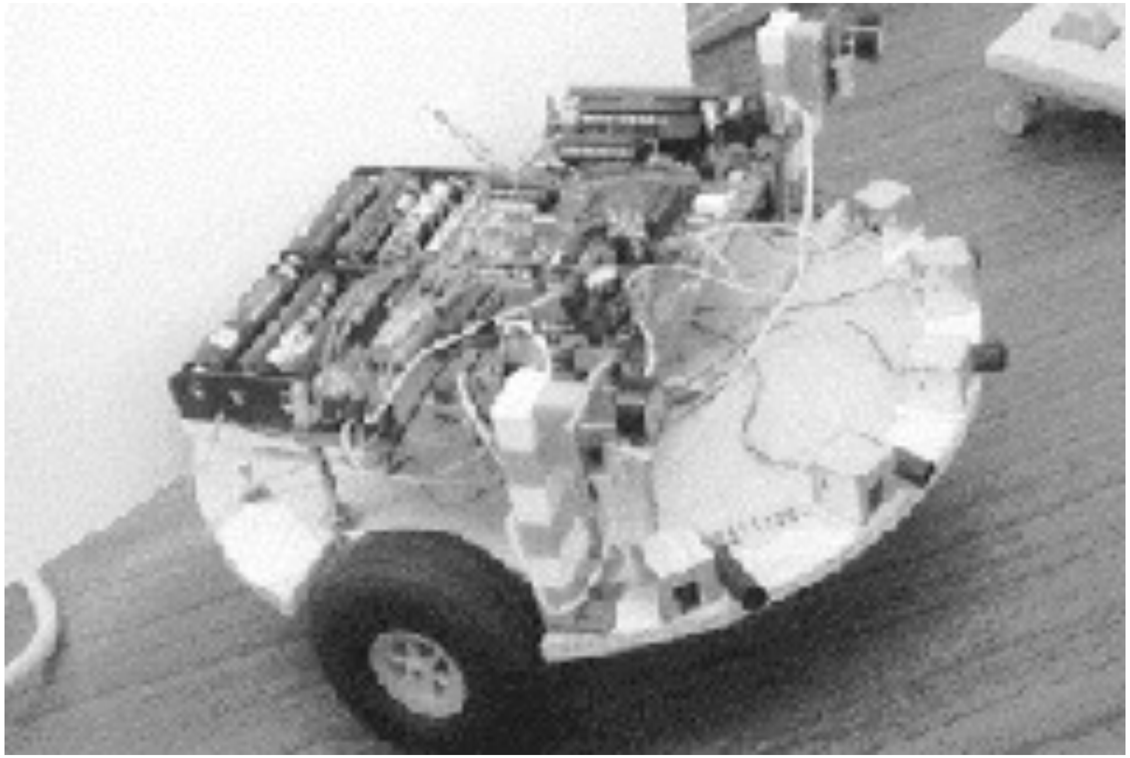
$(i = \text{right or left})$

- summed to total dynamics

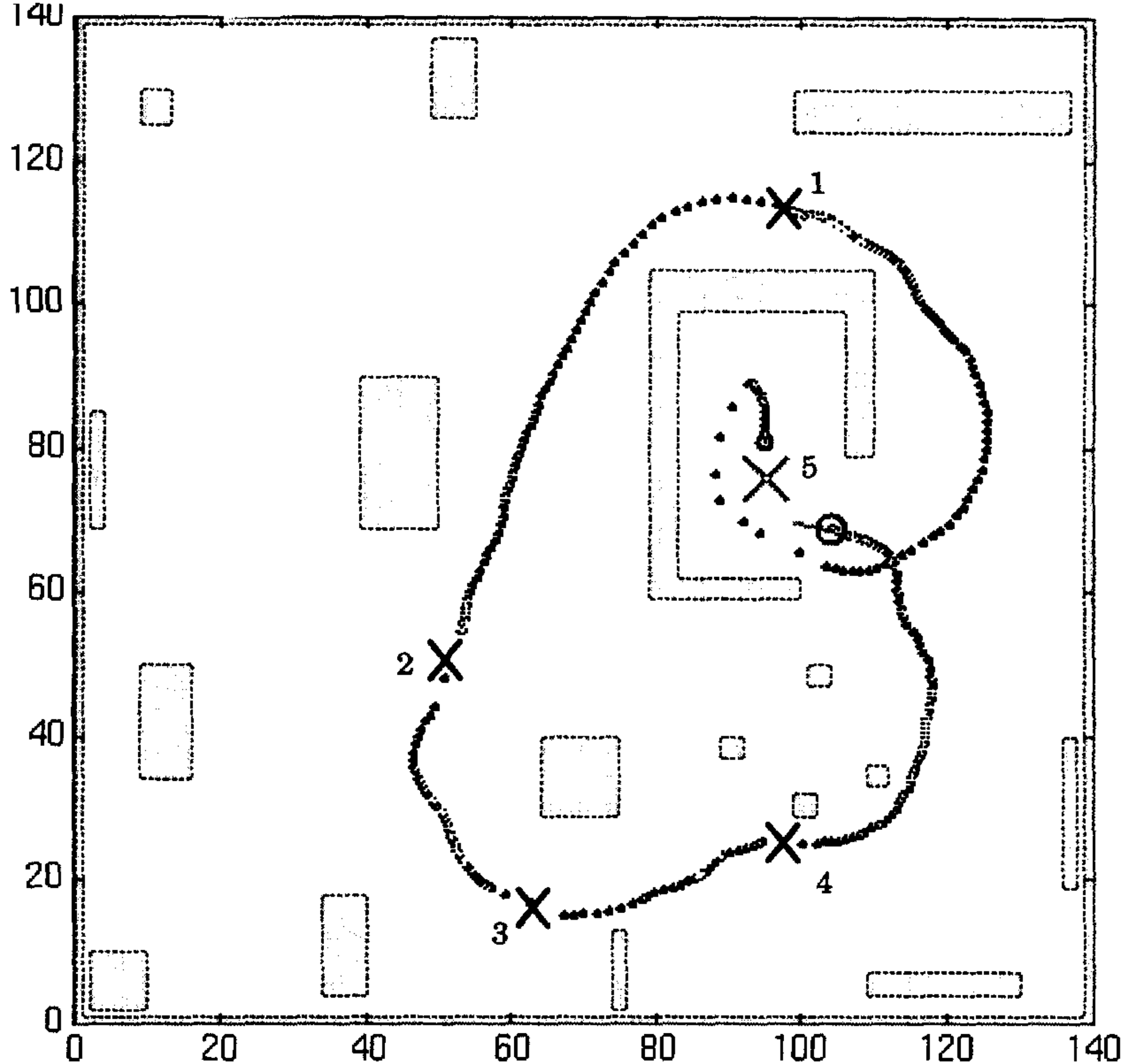
$$g_{\text{left}}(\omega) + g_{\text{right}}(\omega)$$

putting it to work on a simple platform

- Rodinsky!
- circular platform with passive caster wheel
- two (unservoed) motors
- 5 IR sensors
- 2 LDR's
- microcontroller MC68HCA11A0 Motorola (32 K RAM), 8 bit



example trajectories

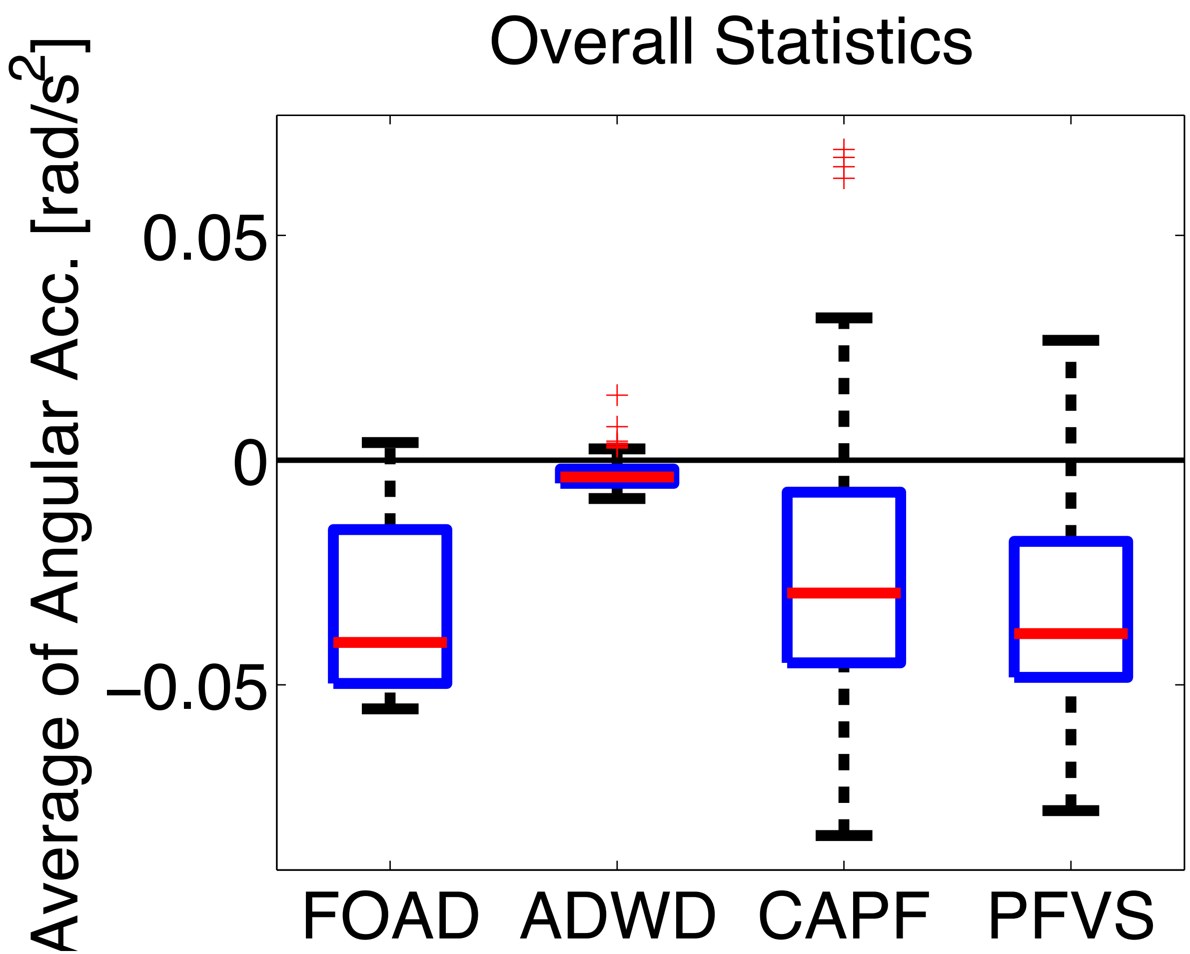


video demonstration

what is the benefit of using second order dynamics?

- ability to integrate constraints which do not specify a particular heading direction, only turning direction
- ability to impose a desired turning rate => enhances agility in turning
- ability to control the second derivative of heading direction=angular acceleration: enables taking into account vehicle dynamics

quantitative comparison



1st order 2nd order

[Hernandes, Becker, Jokeit, Schöner, 2014]

implementations

- larger platform for robot soccer (in Portugal)
- autonomous wheel-chair by Pierre Mallet, Marseille