Bernstein Polynomial Blends for Smooth Rule-Based Flocking

Jeyaprakash Chelladurai Haklin Kimm

East Stroudsburg University of Pennsylvania

ACM SIGSIM-PADS 2026, Vienna, Austria
Modeling Methodology I – Wednesday, June 24, 2026
Session Chairs: Martin Bicher and Joachim Denil

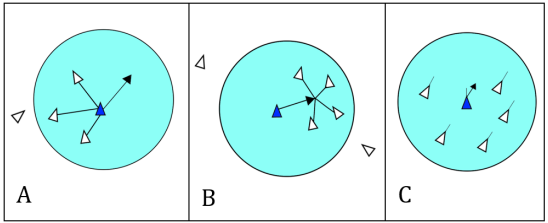
40th ACM SIGSIM Conference on Principles of Advanced Discrete Simulation

20-minute talk map

1. Problem	2. Method	3. Evidence
• Boids use local rules • Fixed weights can cancel • Priorities can flip	• Bernstein shares • Convex, smooth blending • Same simulation for all modes	• Jerk • Heading rate • Curvature and composite score

Main claim: Bernstein blending gives a simple drop-in way to reduce visible jitter while keeping local responsiveness.

Motivation: flocking rules are simple; mixing them is not



Separation, cohesion, and alignment in the Boids model.

Three local rules

- **Separation:** avoid close neighbors.
- **Alignment:** match local heading.
- **Cohesion:** move toward neighbor centroid.

Combination problem

Each rule can be reasonable by itself, but the steering vectors may point in competing directions.

Why fixed linear weights become brittle

Classic weighted sum

$$u_{\mathrm{beh}} = \lambda_{\mathrm{sep}} s_i + \lambda_{\mathrm{ali}} a_i + \lambda_{\mathrm{coh}} c_i$$

- Opposing vectors may partially cancel.
- Small neighborhood changes can cause large heading changes.
- Priority selection avoids cancellation, but introduces hard switches.

Observed artifact

Frame-to-frame **jitter**

and spikes in acceleration, jerk, heading rate, and curvature.

Key idea: use Bernstein shares instead of fixed weights

Bernstein basis properties

- Non-negative shares.
- Shares sum to one.
- Result stays inside the convex hull of behaviors.
- Smooth change as the blend parameter changes.

Behavior blend

$$u_{\mathrm{beh}} = \hat{c}_{\mathrm{sep}}\, w_{\mathrm{sep}} + \hat{c}_{\mathrm{ali}}\, w_{\mathrm{ali}} + \hat{c}_{\mathrm{coh}}\, w_{\mathrm{coh}}$$

$$\hat{c}_{\mathrm{sep}}, \hat{c}_{\mathrm{ali}}, \hat{c}_{\mathrm{coh}} \geq 0, \quad \hat{c}_{\mathrm{sep}} + \hat{c}_{\mathrm{ali}} + \hat{c}_{\mathrm{coh}} \approx 1$$

Interpretation

Alignment becomes a smooth bridge between repulsion and attraction.

Model pipeline

Per time step

1. Find neighbors inside radius and field of view.
2. Compute separation, alignment, cohesion.
3. Blend the three weighted inputs.
4. Add a light boundary nudge.
5. Normalize heading and advance with Forward Euler.

State update

$$x_i^{k+1} = x_i^k + v_i^k \Delta t$$

$$v_i^k = s\,\hat{u}_i^k, \qquad \hat{u}_i^k = \mathrm{normalize}\left(u_{\mathrm{beh},i}^k + u_{\mathrm{bnd},i}^k\right)$$

Fair comparison

Only the steering-combination rule changes across methods.

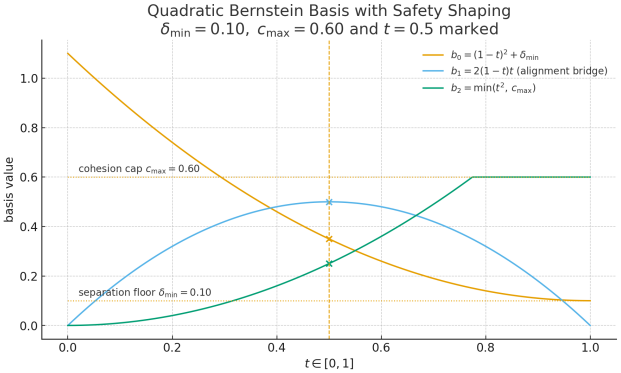
Five steering modes compared

Mode	How it combines rules	Expected risk
Vector addition	Fixed weighted sum	Cancellation jitter
Winner-takes-all	Selects strongest rule each frame	Hard flips
Prioritized budget	Allocates capped steering in priority order	Step pulses
Quadratic Bernstein	One middle alignment bridge	Narrow smoothing band
Cubic Bezier	Two middle lobes for alignment mediation	Smoothest transition

Experimental principle

Same boids, seed, perception, speed, boundary handling, and logging; only the blend rule changes.

Quadratic Bernstein blend: one alignment bridge



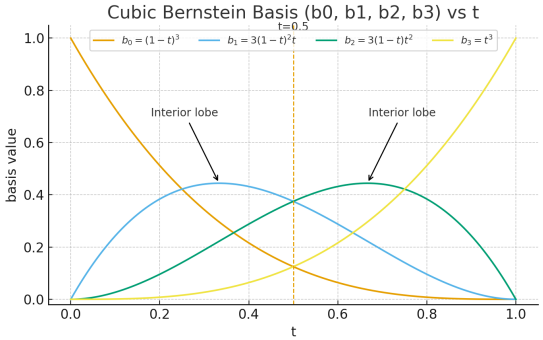
Safety-shaped basis

$$b_0 = (1-t)^2 + \delta_{\min}, \quad b_1 = 2(1-t)t,$$

$$b_2 = \min(t^2, c_{\max})$$

- Separation floor protects safety.
- Cohesion cap avoids over-attraction.
- Middle lobe gives alignment authority near mid-t.

Cubic Bezier blend: two-sided alignment mediation



Cubic basis

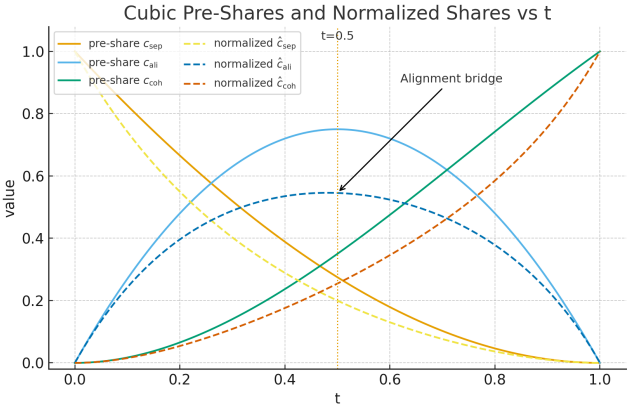
$$b_0 = (1-t)^3, \quad b_1 = 3(1-t)^2 t$$

$$b_2 = 3(1-t)t^2, \quad b_3 = t^3$$

Why it helps

The two interior lobes let alignment mediate both sides of the separation-to-cohesion transition.

Cubic shares after normalization



Pre-shares

$$c_{\text{sep}} = b_0 + g_{\text{sepMid}}\, b_1$$

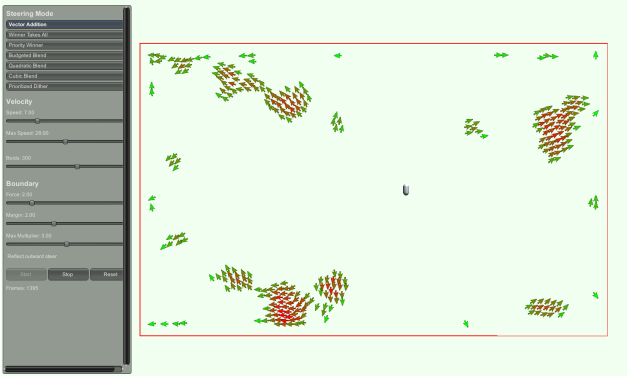
$$c_{\text{ali}} = g_{\text{aliL}}\, b_1 + g_{\text{aliR}}\, b_2$$

$$c_{\text{coh}} = g_{\text{cohMid}}\, b_2 + b_3$$

Normalized shares

$$\hat{c} = \frac{c}{c_{\text{sep}} + c_{\text{ali}} + c_{\text{coh}} + \varepsilon}$$

Simulation setup



Controlled run

- 2D rectangular domain: $[-20, 20] \times [-10, 10]$.
- 300 boids, deterministic seed.
- Perception radius $R = 1.0$; field of view 2π.
- Constant speed, fixed time step.
- 6000 frames per mode.

What we measure

Per-boid kinematics

$$a_i^k = \frac{v_i^k - v_i^{k-1}}{\Delta t}, \qquad j_i^k = \frac{a_i^k - a_i^{k-1}}{\Delta t}$$

$$\omega_i^k = \frac{\theta_i^k - \theta_i^{k-1}}{\Delta t}$$

$$\kappa_i^k = \frac{|v_{ix}^k a_{iy}^k - v_{iy}^k a_{ix}^k|}{\|v_i^k\|^3}$$

Composite smoothness

$$S(t) = z(\mathrm{jerk}) + z(\mathrm{std}(\dot{\theta})) + z(\mathrm{std}(\kappa))$$

- Lower jerk means less abrupt acceleration change.
- Lower heading-rate variation means fewer visible heading jumps.
- Lower curvature variation means steadier paths.

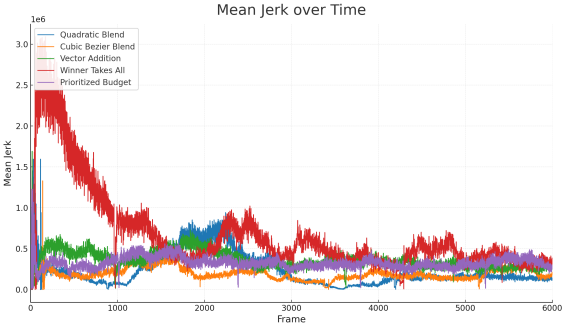
Result summary: smoothness ranking

Rank	Method	Interpretation
#1	Cubic Bezier	Widest alignment bridge; lowest overall roughness.
#2	Quadratic	Lowest curvature; good single-bridge smoothing.
#3	Prioritized budget	Limits large spikes, but creates step-like pulses.
#4	Vector addition	Fast baseline, but cancellation causes jitter.
#5	Winner-takes-all	No cancellation, but abrupt rule switches dominate.

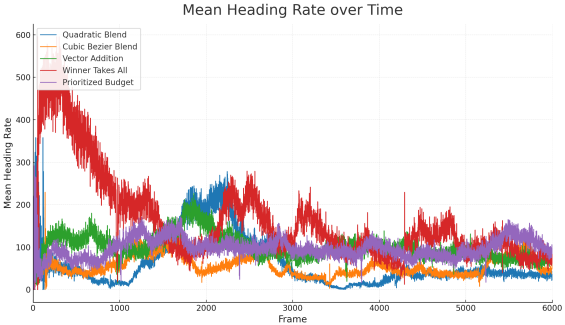
Composite mean ordering

Cubic $<$ Quadratic $<$ Prioritized Budget $<$ Vector Addition $<$ Winner-Takes-All, where lower is smoother.

Results: jerk and heading rate



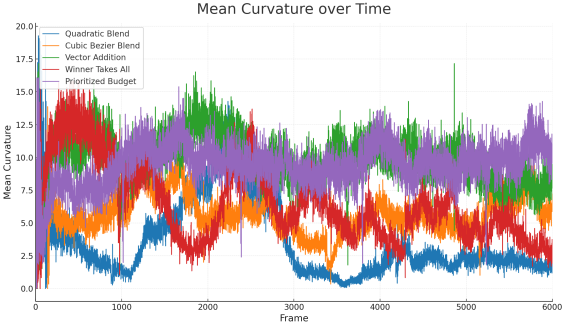
Mean jerk over time.



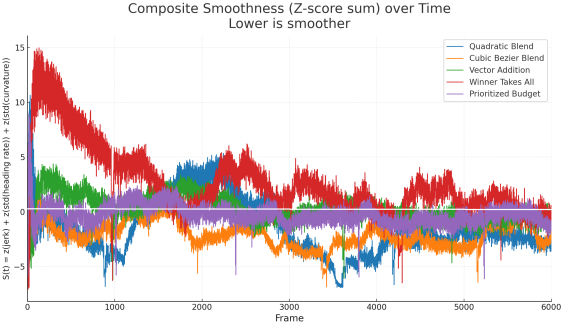
Mean heading rate over time.

- Cubic Bezier gives the calmest steady-state jerk and heading-rate behavior.
- Winner-takes-all shows sharp spikes due to frame-level rule changes.

Results: curvature and composite score



Mean curvature over time.



Composite smoothness over time. Lower is smoother.

- Quadratic has the lowest mean curvature; cubic is close behind.
- Composite score favors cubic because it reduces jerk and heading jumps together.

Why cubic wins in this experiment

What cubic changes

- Does not choose a single winner.
- Does not allow arbitrary cancellation to dominate.
- Gives alignment a broader mediation band.
- Keeps behavior shares bounded and smooth.

Practical consequence

Boids remain responsive to local crowding, but headings evolve continuously as neighborhoods change.

Takeaway

Smooth composition is as important as good local rules.

Limitations and future work

Current scope

- 2D plane.
- Constant speed.
- Fixed sensing parameters.
- Manual blend parameter and share shaping.
- No moving obstacles or disturbances.

Next steps

- Drive t with density, proximity, and occlusion signals.
- Add hysteresis to avoid oscillatory context changes.
- Compare with fuzzy, utility, and learning-based controllers.
- Test pursuit-evasion, coverage, and robotics-inspired tasks.

Conclusion

What we contributed

A Bernstein/Bezier blending framework for combining separation, alignment, and cohesion in rule-based flocking.

What the simulation shows

Cubic Bezier blending produces the smoothest overall motion, with quadratic close behind, under a controlled 300-boid, 6000-frame comparison.

Main message

Replace brittle fixed weights with smooth, non-negative, normalized shares.

Questions?