# Variance or Invariance of Coordinate Systems and Controlled Variables in Motor Control

Massachusetts Institute of Technology
Newman Laboratory for Biomechanics and Human Rehabilitation

**Neville Hogan[a], Hermann Müller[b], Dagmar Sternad[c]**

[a]Mechanical Engineering and Brain & Cognitive Sciences, M.I.T., [b]Movement Science, University of Giessen,
[c]Depts of Biology, Electrical & Computer Science and Physics, Northeastern University

The Action Lab

New York State Center of Research Excellence, contract CO19772,
Toyota Human Partner Robotics Program

NSF PAC-0450218, NIH R01-HD045639, ONR N00014-05.
We would like to thank Se-Woong Park for data analysis.

## Coordinates, Units and Invariance

Analysis of behavioral data aims to identify CNS control priorities. Ideally, analysis should not depend on the coordinates and units of the measured data. At a minimum, awareness of coordinate dependency is necessary.

Invariant features (e.g. hand path straightness) indicate CNS control but coordinate sensitivity can obscure invariance. For example, a logarithmic transformation (e.g. based on Weber's law) changes a quantity as basic as an arithmetic mean.

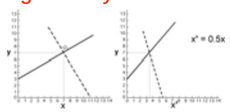
The high dimensionality of the neuromechanical system motivates a search for low-dimensional controlled variables. One pioneering analysis is the Un-Controlled Manifold hypothesis (UCM): variance in "don't care" directions is unrestricted, variance in orthogonal directions is reduced.

While the notion of exploiting redundancy is insightful and attractive, the detailed analysis to identify the "don't care" directions from a behavioral data set faces serious challenges:

**Coordinate dependence and logical circularity.**

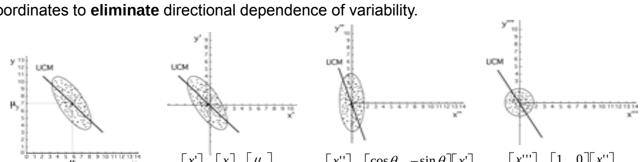
## Coordinate Dependence: Orthogonality

To test a hypothesized controlled variable, UCM analysis identifies variance parallel and orthogonal to the hypothesized UCM.
**Problem: orthogonality depends upon coordinates.**




## Logical Circularity

UCM relies on covariance analysis, calculated within predefined variables, e.g. joint angles. Covariance is real-valued and strictly positive-definite. Consequently, you can **always** change coordinates to **eliminate** directional dependence of variability.



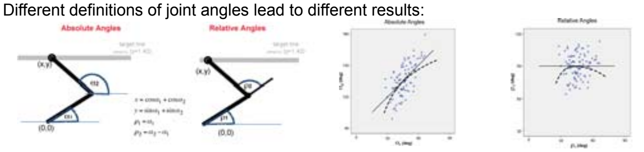

$$\begin{bmatrix}x'\\y'\end{bmatrix}=\begin{bmatrix}x\\y\end{bmatrix}-\begin{bmatrix}\mu_x\\\mu_y\end{bmatrix}\qquad\begin{bmatrix}x''\\y''\end{bmatrix}=\begin{bmatrix}\cos\theta & -\sin\theta\\\sin\theta & \cos\theta\end{bmatrix}\begin{bmatrix}x'\\y'\end{bmatrix}\qquad\begin{bmatrix}x'''\\y'''\end{bmatrix}=\begin{bmatrix}1 & 0\\0 & S\end{bmatrix}\begin{bmatrix}x''\\y''\end{bmatrix}$$

θ: anti-clockwise angle between vertical coordinate axis and ellipse major axis, $S$: scaling factor, $S$ = (minor axis length)/(major axis length)

**Fundamental Circularity**: If you *know* the coordinates in advance, this analysis of covariance may confirm the "don't care" directions, hence the coordinates of control. If you *don't*, this analysis of variability cannot identify the control priority.

## A Mathematical Nicety? – No!

Different definitions of joint angles lead to different results:




## Alternative: Analysis in Result Space

Two "spaces" are identified:

| | |
|---|---|
| **Result space** *R* | defined by variables quantifying performance, e.g. error |
| **Execution space** *X* | defined by variables controlled by the actor |

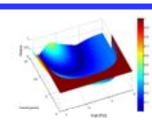
Mechanics of the task determine a map between these two spaces:

$$r = f(x_1,x_2),\ \text{where } r \in R \text{ and } \{x_1, x_2\} \in X$$

If dimension($X$) > dimension(R), redundancy may structure variability

**Key feature:** All analysis of variability is based on comparisons in $R$, a physically defined space that is ***independent of the actor's neural representation.***

## Invariance of the Solution Manifold



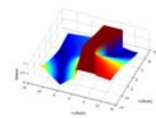
The "solution manifold" *SM* is defined by ideal performance in $R$, e.g., $r = 0$

Consider a change from $\{x_1,x_2\}$ to alternative coordinates $\{a_1,a_2\}$ where $x_1 = b_1(a_1,a_2)$ and $x_2 = b_2(a_1,a_2)$

The map from execution to result space becomes $r = f(b_1(a_1,a_2),b_2(a_1,a_2)) = g(a_1,a_2)$

In original coordinates, *SM* is defined by $0 = f(x_1,x_2)$

In alternative coordinates, *SM* is defined by $0 = g(a_1,a_2)$



Changing execution coordinates *cannot* change ideal performance in $R$.

***The definition of SM is independent of coordinates.***

## Sensitivity ⇔ Curvature

**Puzzle**: If all points on the *SM* are equally satisfactory, why do data typically cluster near a specific location?

**Solution**: Best average result is obtained where performance is least sensitive to variation in execution. Performance is least sensitive where the *SM* has least curvature:

Sensitivity to variation ⇔ Curvature of surface $r = f(x_1,x_2)$

The *SM* has special curvature properties ***independent of execution coordinates.***

$r = f(x_1,x_2)$ is real-valued, hence two principal curvatures, $k_{max}$, $k_{min}$ in *orthogonal* directions

$r$ is *identically* zero along *SM*, hence $k_{min} = 0$ everywhere on *SM*.

The direction of maximum curvature, $k_{max}$, is orthogonal to *SM*.

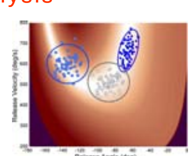
***A "natural" (physically-based) definition of orthogonality in execution space.***

## Theoretical vs. Data-Based Analysis

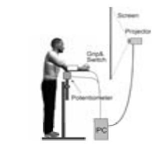
The best location on the *SM* depends on the specific distribution of a varying execution. The distribution of varying executions can take many shapes and is unknown a-priori. Theoretical analysis of *SM* alone is insufficient.

Empirical "data-driven" analysis is advantageous

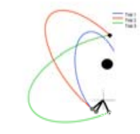
**TNC analysis is a *data-driven* method**



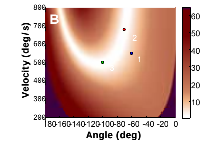

## Tolerance, Noise and Covariation (TNC) Example: Skittles




The Experimental Set-Up

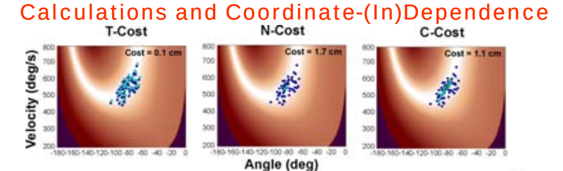


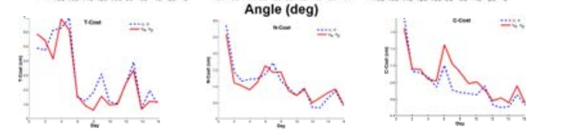
The Virtual Game




Execution and Result Space

## Calculations and Coordinate-(In)Dependence




T-Cost is unaffected by linear coordinate transformations $\mathbf{x} = \mathbf{Ta}$

Observed data point:

$r_o = f(\mathbf{x}_o) = f(\mathbf{Ta}_o) = g(\mathbf{a}_o)$

Shifted data point:

$r_s = f(\mathbf{x}_o+\mathbf{x}_s) = f(\mathbf{Ta}_o+\mathbf{Ta}_s)$

$r_s = f(\mathbf{T}(\mathbf{a}_o+\mathbf{a}_s)) = g(\mathbf{a}_o+\mathbf{a}_s)$

**Coordinate changes that alter UCM analysis do not affect T-Cost.**

N-Cost is unaffected by linear coordinate transformations $\mathbf{x} = \mathbf{Ta}$

Shrink data $\mathbf{x}_o$ to their mean $\mu_x$ by $s = 0.01$

$\Delta\mathbf{x} = s\mathbf{I}(\mathbf{x}_o - \mu_x)$

*Result change:*

$\Delta r \cong \nabla \mathbf{f}^t \Delta \mathbf{x}$

*Alternative coordinates:*

$\Delta r_{alt} \cong \nabla \mathbf{g}^t \Delta \mathbf{a}$

*Transformation:*

$\nabla \mathbf{g}^t = \nabla \mathbf{f}^t \mathbf{T},\ \Delta \mathbf{a} = \mathbf{T}^{-1}\Delta \mathbf{x}$

$\Delta r_{alt} \cong \nabla \mathbf{f}^t \Delta \mathbf{x} = \Delta r$

**Above**: Rank order of T- N- and C-Cost is insensitive to polar or Cartesian coordinates

C-Cost shuffles data without changing marginal distributions.

*Pro*: Unlike UCM analysis, does not require orthogonality.

*Con*: As presently defined is sensitive to coordinate rotations.

## Summary

- UCM analysis requires orthogonality to define parallel and perpendicular variability—hence *fundamentally* coordinate-sensitive
- N-Cost is *insensitive* to smooth coordinate transformations
- Smooth: linear approximation competent over the data set
- T-Cost is *insensitive* to linear coordinate transformations
- C-Cost depends on coordinates

A coordinate-independent formulation is under development