

Specification of a hoisting system that is able
to alternatively raise and lower payloads with
no inversion of motive force.

G.Cordier

August 2, 2023

As an introduction, let us quote[1]:

The problem is based on a real machine, developed by the Renaissance artist/engineer Filippo Brunelleschi during the period 1420-1425. The machine was used to hoist large sandstone and marble blocks for completion of the cupola of the cathedral in Florence [Since] it bothered him [...] that when the block reached the top of the cupola, the oxen had to be un-harnessed and re-harnessed to change the direction of the hoist and lower the lifting tackle back to the ground. Brunelleschi's solution was a three-speed, reversible hoist. This was a novel concept and drew much admiration from other Renaissance artists and engineers [...].

So, Brunelleschi's design may be described as follow: The very hoisting system (*the gear*: A wheel around an horizontal shaft) gets motion from a main (vertical) shaft. Such motion is captured from a wheel (A) that is rigidly fixed to the main shaft. In other words, A is the driving wheel. Contact between gear and wheel A (only) occurs at point P; see [Figure 1].

Moreover, a second wheel A' , which can be seen as a (distinct) copy of A is symmetrically set. More specifically, the symmetry is performed with respect to the "mirror"plan (O, X, Z); see [Figure 1].

The key ingredient is that a vertical shift can make the gear switch contact P to its A' counterpart P' . The gear is then equipped with (only) two degrees of freedom (off rotation) : Either contact in P or in P'

It can now be shown that the gear can either lift or pull payloads, on demand. To do so, it is enough to switch from one state/configuration to the other one. In other words, raising and lowering payloads can now be done (1) with a single device, (2) **without reversing** the driving force, *i.e.* without reversing the vertical shaft rotation. To sum it up, changing the rotation direction is no longer done at the "engine level" but, instead, by a simple, effortless shift of the wheels.

The following specification "encodes" the system (the gear) and its environment (wheels A, A') at a geometrical level of astraction; see [Figure 1.B]. Provided celerity vector x , radius vector y and normal vector z (all being normalized, then nonzero), the rotation direction is identified with the determinant $\det(x, y, z) \in \{-1, 1\}$ of the block matrix $[xyz]$. The spec shall then be proved once established that such determinant "oscillates" between

-1 (given one contact, say P) and 1 (provided the other one, say P').

The following specification is written in TLA+; see [2], [3]. TLA+ is based on TLA, an extension of the first logic order (in ZFC). Noticeably,

- (a) ' denotes a variable assignement at the next following step;
- (b) \square asserts that a formula is always valid, *i.e.* true for every reachable step.

Such specification is actually pretty general:

- (a) It does not force us to explicitly choose a rotation direction for the vertical shaft;
- (b) It does not force us to explicitly choose an initial driving wheel (A or A');
- (c) In theory, it is easy to drop gear and wheels, so that the system keeps working as long as there exists at any time a single contact point P, P'. For instance, we may replace wheels by arbitray convex shapes. Another approach is to see that the fundamental properties of the whole system are stable under "conservative" diffeomorphism φ (more rigorously: $\det \varphi' > 0$). This would mean a of useless trouble in practice ...but that is not a part of the specification!

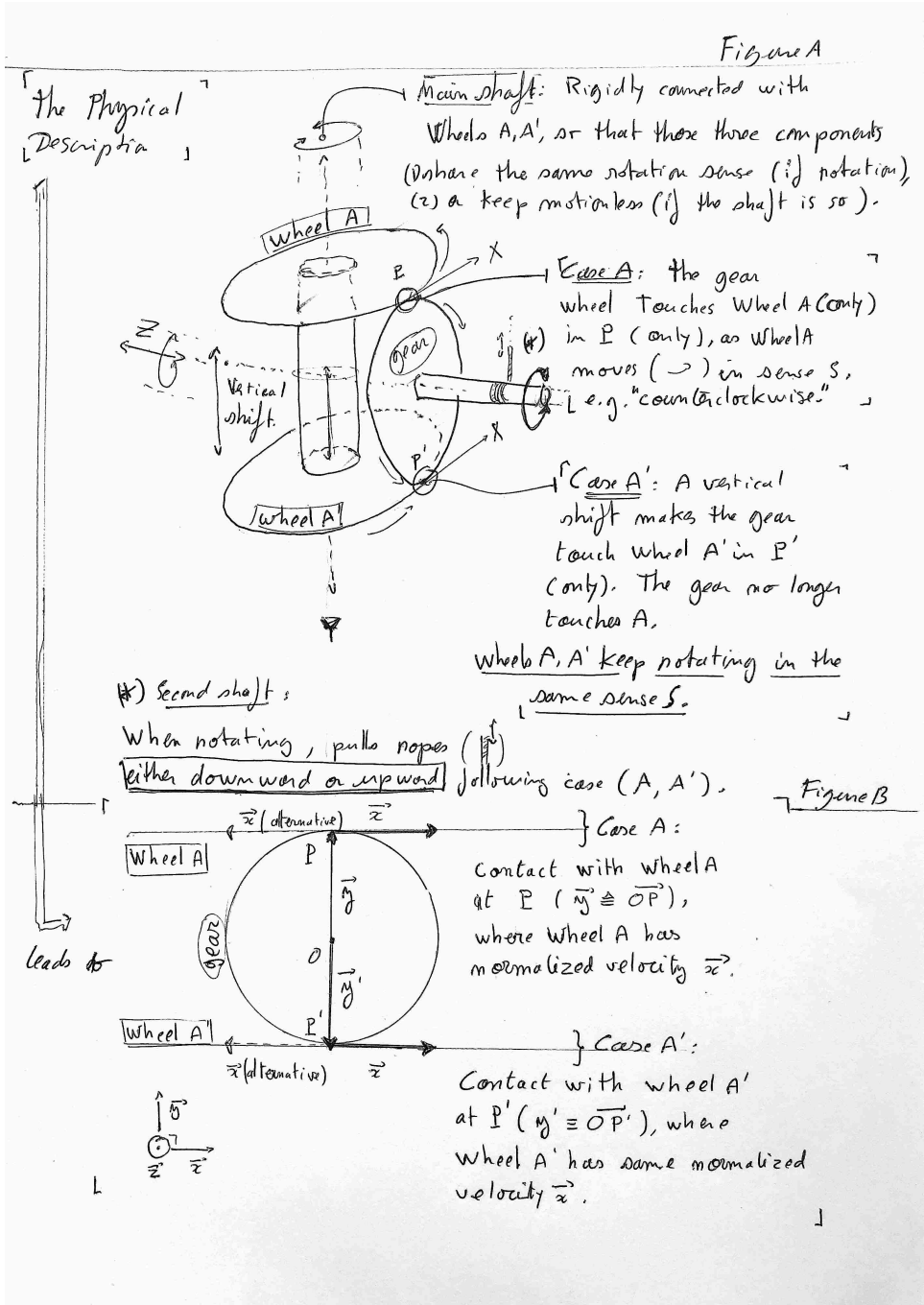


Figure 1: Physical and geometrical descriptions of the system (the gear) and its environment (the vertical shaft)

EXTENDS Reals, TLAPS

Without loss of generality, see our figures, the point of view vector Z is $\langle 0, 0, 1 \rangle$, as the gear wheel lies in the plane (O, X, Y) ; where O is the center of the gear wheel. Still without loss of generality, such wheel has constant radius 1.

CONSTANT Z Z MUST BE the constant vector $\langle 0, 0, 1 \rangle$

VARIABLES x, y 3D vectors of the physical space.

Without loss of generality, we only pick x, y in the following Vect; see above assumptions.

Vect \triangleq {
 $\langle 1, 0, 0 \rangle, \langle -1, 0, 0 \rangle,$
 $\langle 0, 1, 0 \rangle, \langle 0, -1, 0 \rangle$
 }

InnerProd is the inner product $x \cdot y$. Hence

InnerProd \triangleq
 $x[1] * y[1] +$
 $x[2] * y[2] +$
 $x[3] * y[3]$

Our assumption $Z = \langle 0, 0, 1 \rangle$ implies that the matrix $[x \ y \ z]$ has determinant $\text{Det} := x[1] * y[2] - x[2] * y[1]$. Hence the following operator Det

Det $\triangleq x[1] * y[2] - x[2] * y[1]$

InvX: x is a picked in Vect *: now x is fixed here and MUST NOT change.

InvX \triangleq
 $\wedge x \in \text{Vect}$
 $\wedge x[1] \in \{ -1, 1 \}$
 $\wedge x[2] = 0$
 $\wedge x[3] = 0$

InvY y is a picked in Vect as well.

InvY \triangleq
 $\wedge y \in \text{Vect}$
 $\wedge y[1] = 0$
 $\wedge y[2] \in \{ -1, 1 \}$
 $\wedge y[3] = 0$

InvXY: x and y MUST BE nontrivially orthogonal

InvXY \triangleq
 $\wedge x \neq \langle 0, 0, 0 \rangle$ Redundant whether InvX is enabled.

$\wedge y \neq \langle 0, 0, 0 \rangle$ Redundant whether InvY is enabled.
 $\wedge \text{InnerProd} = 0$

Let us repeat that $Z = \langle 0, 0, 1 \rangle$
 $\text{InvZ} \triangleq Z = \langle 0, 0, 1 \rangle$

Our invariant $\square \text{Inv}$ is now “controlled” by the following Inv
 $\text{Inv} \triangleq$
 $\wedge \text{InvX}$
 $\wedge \text{InvY}$
 $\wedge \text{InvXY}$
 $\wedge \text{InvZ}$

The Next action
 $\text{Next} \triangleq$

So, x no action will ever change x:
 $\wedge \text{UNCHANGED } x$
 y is flipped in the sense that $y' := -y$:
 $\wedge y' = [y \text{ EXCEPT } ![1] = -y[1], ![2] = -y[2]]$

Our spec Spec, then. Remark that Inv is also the initial condition
 $\text{Spec} \triangleq \text{Inv} \wedge \square [\text{Next}]_{\langle x, y \rangle}$

Typing variables: Relevant type is Vect.
 $\text{Typing}(v) \triangleq v \in \text{Vect}$

The following lemma states that Next preserves Inv.

LEMMA LemInv $\triangleq \text{Inv} \wedge \text{Next} \implies \text{Inv}'$

$\langle 1 \rangle$ SUFFICES ASSUME $\text{Inv} \wedge \text{Next}$

PROVE Inv'

OBVIOUS

$\langle 1 \rangle$ USE DEF Vect, InnerProd, InvX, InvY, InvXY, InvZ, Inv, Next

$\langle 1 \rangle 1$ $\text{Inv} \wedge \text{Next} \implies \text{Inv}'$ OBVIOUS

$\langle 1 \rangle 4$ QED BY $\langle 1 \rangle 1$

Equivalently, the invariant $\square \text{Inv}$ is true under specs Spec.

THEOREM ThInv $\triangleq \text{Spec} \implies \square \text{Inv}$

$\langle 1 \rangle$ USE DEF Vect, InnerProd, InvX, InvY, InvXY, InvZ, Inv

$\langle 1 \rangle 1$ $\text{Inv} \wedge \text{UNCHANGED } \langle x, y \rangle \implies \text{Inv}'$ BY DEF Inv

$\langle 1 \rangle 2$ $\text{Inv} \wedge \square [\text{Next}]_{\langle x, y \rangle} \implies \square \text{Inv}$ BY PTL, LemInv, $\langle 1 \rangle 1$

$\langle 1 \rangle$ QED BY $\langle 1 \rangle 1$, $\langle 1 \rangle 2$ DEF Spec

ThInv straightforwardly establishes that, under specification Spec, x, y ,
have always type Vect.

THEOREM ThType \triangleq Spec $\implies \Box$ Typing(x) \wedge \Box Typing(y)

$\langle 1 \rangle$ SUFFICES ASSUME Spec

PROVE \Box Typing(x) \wedge \Box Typing(y)

OBVIOUS

$\langle 1 \rangle$ USE DEF

Vect, InnerProd, Det, InvX, InvY, InvXY, InvZ, Inv, Next, Spec, Typing

$\langle 1 \rangle 1$ Spec $\implies \Box$ InvBY ThInv

$\langle 1 \rangle 2$ Inv \implies Typing(x) \wedge Typing(y) OBVIOUS

$\langle 1 \rangle 3$ QED BY PTL, $\langle 1 \rangle 1$, $\langle 1 \rangle 2$ DEF Inv, Typing

The following theorem asserts that there are only three options for
the step $\det(x, y, z) \rightarrow \det(x', y', z')$:

1. $\det(x, y, Z) = \det(x', y', Z) \wedge$ UNCHANGED $\langle x, y \rangle$

2. $\det(x, y, Z) = 1 \wedge \det(x', y', Z) = -1 \wedge$ Next

3. $\det(x, y, Z) = -1 \wedge \det(x', y', Z) = 1 \wedge$ Next

which establishes that our spec is correct. QED

THEOREM ThOscillatingDet \triangleq Inv \wedge Next \implies

\wedge Det $\in \{-1, 1\}$

\wedge Det' $\in \{-1, 1\}$

\wedge Det' = - Det

$\langle 1 \rangle$ SUFFICES ASSUME Inv \wedge Next

PROVE

\wedge Det $\in \{-1, 1\}$

\wedge Det' $\in \{-1, 1\}$

\wedge Det' = - Det

OBVIOUS

$\langle 1 \rangle$ USE DEF

Vect, InnerProd, Det, InvX, InvY, InvXY, InvZ, Inv, Next, Spec

$\langle 1 \rangle 1$ Inv \wedge Next \implies Det $\in \{-1, 1\}$ OBVIOUS

$\langle 1 \rangle 2$ Inv \wedge Next \implies Det' $\in \{-1, 1\}$ OBVIOUS

$\langle 1 \rangle 3$ Inv \wedge Next \implies Det' = - Det OBVIOUS

$\langle 1 \rangle 4$ QED BY $\langle 1 \rangle 1$, $\langle 1 \rangle 2$, $\langle 1 \rangle 3$

References

- [1] Biomimetics & Dexterous Manipulation Laboratory. BrunelleschiNotes. <http://bdml.stanford.edu/Main/BrunelleschiNotes>. Accessed: August 2, 2023.
- [2] Leslie Lamport. TLA+. <http://lamport.azurewebsites.net/tla/tla.html>. Accessed: August 2, 2023.
- [3] Markus Kuppe et al. TLA+ repository. <https://github.com/tlaplus>. Accessed: August 2, 2023.