Specification of Brunelleschi's hoist: A device that alternatively raises and lowers payloads without inversion of motive force.

G.Cordier

August 4, 2023

1 From Brunelleschi's hoist to TLA+ specification

As an introduction, let us quote[1]:

> The problem is based on a real machine, developed by the Renaissance goldsmith/sculptor/architect/engineer/inventor 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 unharnessed 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 $\det(\mathrm{x}, \mathrm{y}, \mathrm{z}) \in \{\text{-}1, 1\}$ of the block matrix $[\mathrm{x}\,\mathrm{y}\,\mathrm{z}]$. 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) $\Box$ 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 explicitely choose a rotation direction for the vertical shaft;

(b) It does not force us to explicitely 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!

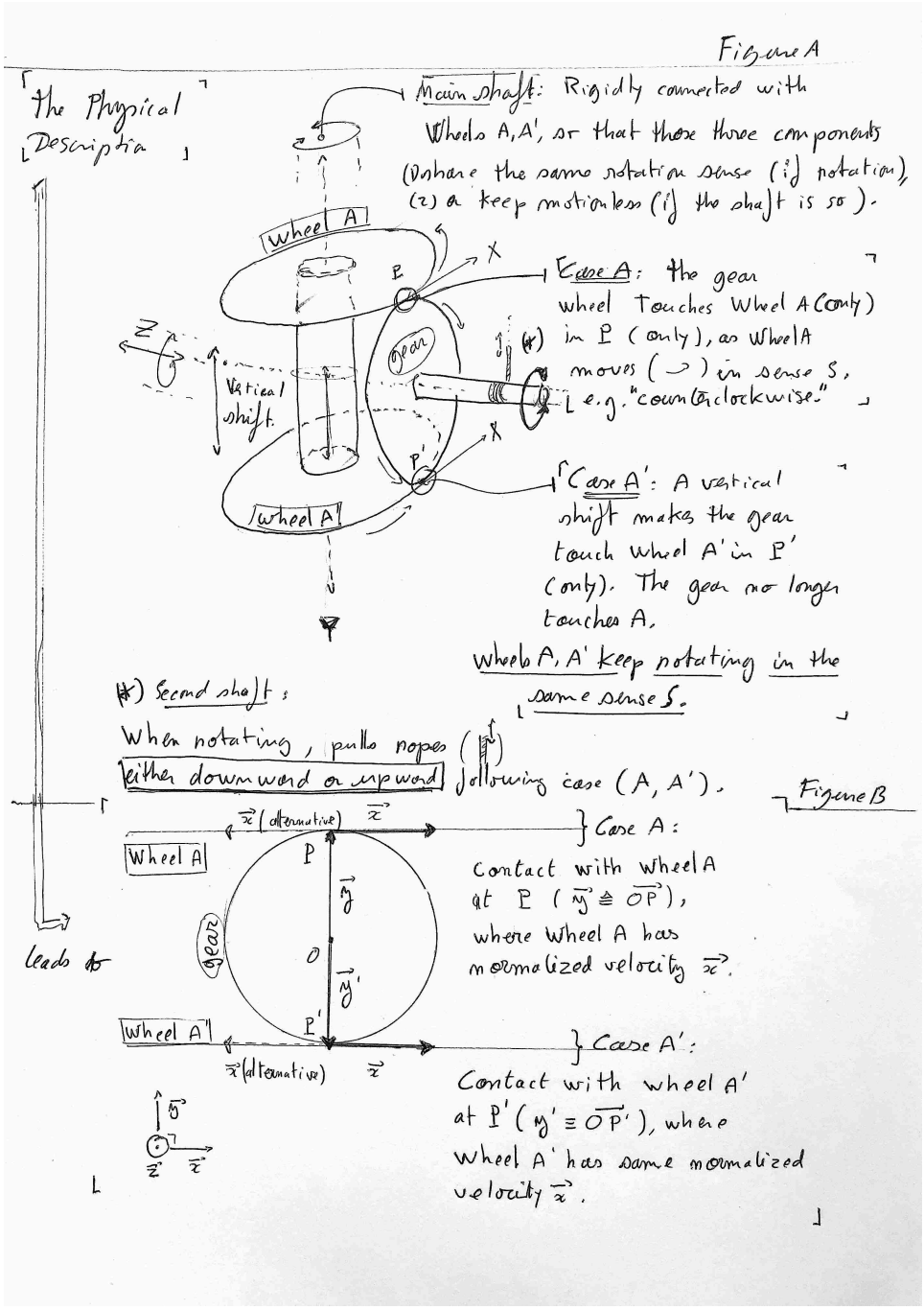

Figure 1: Physical and geometrical descriptions of the system (the gear) and its environment (the vertical shaft)

2 The TLA+ source code

MODULE gear

EXTENDS Integers, 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.

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

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

$\text{abs}[t \in \text{Int}] \triangleq$ IF $t > -t$ THEN t ELSE $-t$

Circle is a subset of Int × Int × {0}; which is infinite... This works (slowly!)IF you add additional constraints in InvX, InvY (See the "Unecessay if . . " comment). IF NOT, then it fails: TLAPS will not solve any polynomial equation for you.

Note that TLC will not like it either and will raise the usual "non-enumerable set" exception.

$$\begin{array}{l}\text{Circle} \triangleq \{\\ \quad w \in \text{Int} \times \text{Int} \times \{0\} :\\ \qquad \text{Manhattan norm:}\\ \qquad \land\ \text{abs}[w[1]] + \text{abs}[w[2]] = 1\\ \qquad \text{Euclidian norm:}\\ \qquad \land\ \text{abs}[w[1]] * \text{abs}[w[1]] + \text{abs}[w[2]] * \text{abs}[w[2]] = 1\\ \}\end{array}$$

Better use this Circle if you want TLC be nice to your spec. You can drop the additional constaints in InvX, InvY (see above), since CircleTLC is now a subset of a FINITE set.

$$\begin{array}{l}\text{CircleTLC} \triangleq \{\\ \quad w \in \{-1, 0, 1\} \times \{-1, 0, 1\} \times \{0\} : \text{abs}[w[1]] + \text{abs}[w[2]] = 1\\ \}\end{array}$$

Simpler is better: Straighforward definition of Circle:

$$\begin{array}{l}\text{CircleEnum} \triangleq \{\\ \quad \langle 1, 0, 0\rangle, \langle -1, 0, 0\rangle,\\ \quad \langle 0, 1, 0\rangle, \langle 0, -1, 0\rangle\\ \}\end{array}$$

You may drop "BY DEF ... abs" whether you use it.

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

$\text{InnerProd} \triangleq x[1] * y[1] + x[2] * y[2]$

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

$$\text{Det} \triangleq \text{IF } (x \in \text{Circle} \land y \in \text{Circle}) \text{ THEN } x[1] * y[2] - x[2] * y[1] \text{ ELSE } 0$$

InvX: x is picked in Circle $*$; now x is fixed and MUST NOT change.

$$\begin{aligned}
\text{InvX} \triangleq\ & \land\ x \in \text{Circle} \\
& \text{Unnecessary if Circle is explicitely defined as a finite set:} \\
& \land\ x[1] \in \{-1, 1\} \\
& \land\ x[2] = 0
\end{aligned}$$

InvY y is picked in Circle as well.

$$\begin{aligned}
\text{InvY} \triangleq\ & \land\ y \in \text{Circle} \\
& \land\ y[1] = 0 \\
& \text{Unnecessary if Circle is explicitely defined as a finite set:} \\
& \land\ y[2] \in \{-1, 1\}
\end{aligned}$$

InvXY: x and y MUST BE nontrivially orthogonal.

$$\begin{aligned}
\text{InvXY} \triangleq\ & \land\ x \in \text{Circle} \\
& \land\ y \in \text{Circle} \\
& \land\ \text{InnerProd} = 0
\end{aligned}$$

Our invariant $\Box$Inv is now "controlled" by the follwing Inv.

$$\begin{aligned}
\text{Inv} \triangleq\ & \land\ \text{InvX} \\
& \land\ \text{InvY} \\
& \land\ \text{InvXY}
\end{aligned}$$

The Next action:

$$\begin{aligned}
\text{Next} \triangleq\ & \text{Boundaries} \\
& \land\ y \in \text{Circle} \\
& \land\ x \in \text{Circle} \\
& \text{So, x no action will ever change x:} \\
& \land\ \text{UNCHANGED } x \\
& \land\ y' \in \text{Circle} \quad \text{You want to request that.} \\
& \text{y is flipped in the sense that } y' := -y : \\
& \land\ y' = [y \ \text{ EXCEPT } ![1] = -y[1],\ ![2] = -y[2]]
\end{aligned}$$

Our spec Spec, then. Remark that Inv is also the initial condition.

$\text{Spec} \triangleq \text{Inv} \wedge \Box[\text{Next}]_{\langle x, y\rangle}$

Typing variables: Relevant type is Circle.

$\text{Typing}(v) \triangleq v \in \text{Circle}$

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
$\langle 1\rangle 1$ $\text{Inv} \wedge \text{Next} \implies \text{InvX}'$
BY DEF InvX, InvXY, Inv, Next
$\langle 1\rangle 2$ $\text{Inv} \wedge \text{Next} \implies \text{InvY}'$
BY DEF InvY, InvXY, Inv, Next, Circle
$\langle 1\rangle 3$ $\text{Inv} \wedge \text{Next} \implies \text{InvXY}'$
BY DEF InvXY, Inv, Next, Circle, InnerProd
$\langle 1\rangle 4$ QED BY $\langle 1\rangle 1$, $\langle 1\rangle 2$, $\langle 1\rangle 3$ DEF Inv

Equivalently, the invariant $\Box$Inv is true under specs Spec.

THEOREM ThInv $\triangleq$ $\text{Spec} \implies \Box\text{Inv}$
$\langle 1\rangle 1$ $\text{Inv} \wedge \text{UNCHANGED } \langle x, y\rangle \implies \text{Inv}'$
BY DEF Circle, InnerProd, InvX, InvY, InvXY, Inv
$\langle 1\rangle 2$ $\text{Inv} \wedge \Box[\text{Next}]_{\langle x, y\rangle} \implies \Box\text{Inv}$
BY PTL, LemInv, $\langle 1\rangle 1$
$\langle 1\rangle$ QED BY PTL, $\langle 1\rangle 1$, $\langle 1\rangle 2$ DEF Spec

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

ThType: Stephan Merz′ version.

THEOREM ThTypeCompactVersion $\triangleq$ $\text{Spec} \implies \Box\text{Typing}(x) \wedge \Box\text{Typing}(y)$
$\langle 1\rangle$.Inv $\implies \text{Typing}(x) \wedge \text{Typing}(y)$
BY DEF Inv, InvX, InvY, Typing
$\langle 1\rangle$.QED
BY ThInv, PTL

ThType: Stephan Merz′ version, with a SUFFICES ASSUME − PROVE .

THEOREM ThType $\triangleq$ $\text{Spec} \implies \Box\text{Typing}(x) \wedge \Box\text{Typing}(y)$
$\langle 1\rangle$ SUFFICES ASSUME Spec
PROVE
$\wedge\, \text{Spec} \implies \Box\text{Inv}$

$\land$ Inv $\implies$ Typing(x) $\land$ Typing(y)
BY PTL DEF Inv, InvX, InvY, Typing
$\langle 1 \rangle$ Inv $\implies$ Typing(x) $\land$ Typing(y)
BY DEF Inv, InvX, InvY, Typing
$\langle 1 \rangle$ QED BY ThInv, PTL

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

1. det(x, y, z) = det(x′, y′, z) $\land$ UNCHANGED $\langle$x, y$\rangle$
2. det(x, y, z) = 1 $\land$ det(x′, y′, z) = $-$1 $\land$ Next
3. det(x, y, z) = $-$1 $\land$ det(x′, y′, z) = 1 $\land$ Next;

which establishes that our spec is correct. QED

THEOREM ThOscillatingDet $\triangleq$ Inv $\land$ Next $\implies$
$\land$ Det $\in \{-1, 1\}$
$\land$ Det′ $\in \{-1, 1\}$
$\land$ Det′ = $-$Det
$\langle 1 \rangle$ SUFFICES ASSUME Inv $\land$ Next
PROVE
$\land$ Det $\in \{-1, 1\}$
$\land$ Det′ $\in \{-1, 1\}$
$\land$ Det′ = $-$Det
OBVIOUS
$\langle 1 \rangle$1 Inv $\land$ Next $\implies$ Det $\in \{-1, 1\}$
BY DEF InvX, InvY, Inv, Next, Det, Circle, abs
$\langle 1 \rangle$2 Inv $\land$ Next $\implies$ Det′ = $-$Det
BY DEF InvX, InvY, Inv, Next, Det, Circle
$\langle 1 \rangle$3 Inv $\land$ Next $\implies$
$\land$ Det $\in \{-1, 1\}$
$\land$ Det′ = $-$Det
BY $\langle 1 \rangle$1, $\langle 1 \rangle$2
$\langle 1 \rangle$4 Det $\in \{-1, 1\} \land$ Det′ = $-$Det $\implies$ Det′ $\in \{-1, 1\}$
OBVIOUS
$\langle 1 \rangle$5 Inv $\land$ Next $\implies$ Det′ $\in \{-1, 1\}$
BY $\langle 1 \rangle$3, $\langle 1 \rangle$4
$\langle 1 \rangle$6 QED BY $\langle 1 \rangle$1, $\langle 1 \rangle$2, $\langle 1 \rangle$3, $\langle 1 \rangle$5

\ ∗ Modification History
\ ∗ Last modified Fri Aug 04 16:13:30 CEST 2023 by gcordier
\ ∗ Created Wed Aug 01 13:07:52 CEST 2023 by gcordier

3 Aknowledgements

I'm extremely grateful to Stephan Merz, of INRIA Nancy (VeriDis team, LORIA laboratory), who read the specification and suggested improvements that made the specification simultaneously more concise and more expressive.

References

[1] Biomimetics & Dexterous Manipulation Laboratory. BrunelleschiNotes. http://bdml.stanford.edu/Main/BrunelleschiNotes. Accessed: August 4, 2023.

[2] Leslie Lamport. TLA+. http://lamport.azurewebsites.net/tla/tla.html. Accessed: August 4, 2023.

[3] Markus Kuppe et al. TLA+ repository. https://github.com/tlaplus. Accessed: August 4, 2023.