

Geometric and numerical aspects of redundancy

Pierre-Brice Wieber, Adrien Escande, Dimitar Dimitrov and Alexander Sherikov

Abstract If some resources of a robot are redundant with respect to a given objective, they can be used to address other, additional objectives. Since the amount of resources required to realize a given objective can vary, depending on the situation, this gives rise to a limited form of decision making, when assigning resources to different objectives according to the situation. Such decision making emerges in case of conflicts between objectives, and these conflicts appear to be situations of linear dependency and, ultimately, singularity of the solutions. Using an elementary model of a mobile manipulator robot with two degrees of freedom, we show how standard resolution schemes behave unexpectedly and inefficiently in such situations. We propose then as a remedy to introduce carefully tuned artificial conflicts, in the form of a trust region.

1 Preamble on Redundancy in Robotics

According to the Oxford Dictionary of English, redundancy is the state of being not or no longer needed or useful. If some resources of a robot appear to be not needed or useful to realize a given objective, a common idea is to make use of them to address another, additional objective. This gives rise to so-called *redundancy resolution schemes* [1]. In the typical, iterative procedure, the robot is assigned to:

Pierre-Brice Wieber

INRIA Grenoble Rhône-Alpes e-mail: Pierre-Brice.Wieber@inria.fr

Adrien Escande

CNRS-AIST Joint Robotics Laboratory UMI3218/RL e-mail: adrien.escande@gmail.com

Dimitar Dimitrov

INRIA Grenoble Rhône-Alpes e-mail: mail@drdv.net

Alexander Sherikov

INRIA Grenoble Rhône-Alpes e-mail: Alexander.Sherikov@inria.fr

1. Realize a first objective. If some resources appear to be redundant, they can be used to additionally...
2. Realize a second objective. If some resources still appear to be redundant, they can be used to additionally...
3. Realize a third objective, *etc.*

1.1 Kinematic Redundancy

Historically, the primary target of redundancy resolution schemes has been the kinematics of robots, where the resources considered are basically degrees of freedom, and the objectives are standard kinematic tasks (reaching, pointing) [1]. Recent developments in this field include the capacity to consider tasks expressed either as equality or inequality constraints [2]. This allows handling kinematic tasks such as avoiding, or staying within a region. As a typical example, proposed in [2], a humanoid robot is assigned with the following objectives:

1. Maintain balance. If possible, additionally...
2. Avoid collisions. If possible, additionally...
3. Reach an object. If possible, additionally...
4. Keep this object within sight.



Fig. 1 As a typical example, proposed in [2], a humanoid robot is assigned with the following objectives: 1. Maintain balance; 2. Avoid collisions; 3. Reach an object; 4. Keep this object within sight.

A key property of inequality constraints is that they can be active or not, depending on the situation. As a result, the amount of resources required to realize an objective can vary. In the example given above, depending on the position of obstacles, the robot may have enough remaining resources to reach the target object or

not, and keep it within sight or not (see Figure 1). This gives rise to a limited form of decision making.

1.2 Force Redundancy

Forces are another resource than can classically become redundant in a robot. This can be actuator forces, or contact forces with the environment. As an example, proposed in [3], a humanoid robot is assigned with the following objectives:

1. Maintain balance. If possible, additionally...
2. Reach an object. If possible, additionally...
3. Use only specified contact forces.

In this case, the redundancy resolution scheme decides whether to use only the specified contact forces or not, depending on the position of the target object. This is used to avoid putting weight on a potentially risky surface, unless absolutely necessary to reach a target object with the right hand (see Figure 2).

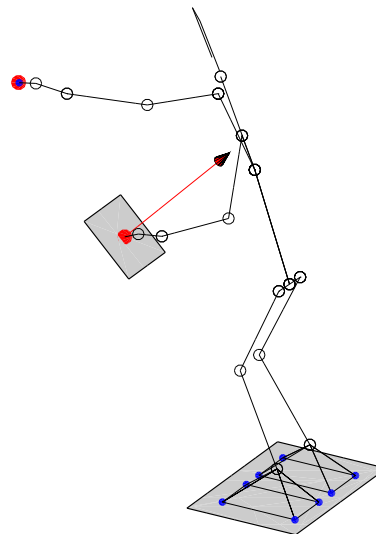


Fig. 2 In this example, a humanoid robot avoids putting weight with its left hand on a potentially risky surface, unless absolutely necessary to reach a target object with the right hand.

1.3 Time Redundancy

Another resource which is less often discussed is time. Typical circumstances are that some objectives should be realized either:

- as soon as possible (they should be realized at some point in the future, and if possible, they should be realized earlier as well), as used in the example below to generate time optimal motions,
- as long as possible (they should be realized now, and if possible, they should be realized later as well), as investigated in [4] to generate safe motions in uncontrolled environments,
- as much as possible (it is equally important to realize the objectives now and later), as actually investigated in [3] to make decisions on balanced contact phases.

As an example, proposed in [5], two industrial manipulator robots (Figure 3) are assigned with the following objectives:

1. Avoid collisions. If possible, additionally. . .
2. Do not accelerate too rapidly. If possible, additionally. . .
3. Reach their targets as soon as possible.

This results in generating time-optimal reaching motions online, under acceleration constraints and, more importantly, collision avoidance.

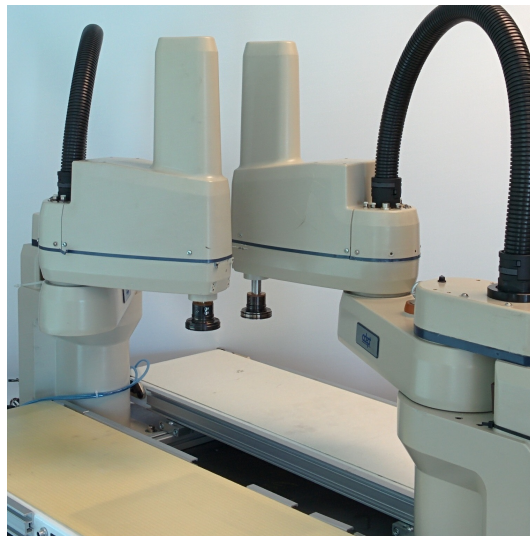


Fig. 3 In this example, two industrial manipulator robots are assigned with the following objectives: 1. Avoid collisions; 2. Do not accelerate too rapidly; 3. Reach their targets as soon as possible.

1.4 A general and fundamental issue

Beyond kinematics, forces, and time, the idea of making use of redundant resources to address additional objectives is very general. Even Isaac Asimov's famous Three

Laws of Robotics can be reformulated to emphasize an underlying aspect of redundancy of resources:

1. A robot may not injure a human being or, through inaction, allow a human being to come to harm. If possible, additionally...
2. A robot must obey the orders given to it by human beings. If possible, additionally...
3. A robot must protect its own existence.

These three laws exemplify our final goal as roboticists, which is to have robots able to tackle complex situations, where different objectives have to be addressed concurrently. These objectives may conflict, and the robot has to make decisions accordingly. These decisions may involve important safety issues such as ‘maintain balance’, ‘avoid collisions’, ‘do not injure a human being’. Our observation here is that redundancy resolution schemes are all about that: resolving conflicts between objectives, making decisions accordingly, and enforcing in the end the safety of the robot and of its environment. This is undeniably a fundamental issue in robotics, and the main motivation for the following analysis.

2 The Case with Two Objectives

Suppose that a robot is assigned with two objectives, that can be expressed as two functions $f(r)$ and $g(r)$ that should be equal to 0. If there is a conflict and these functions can't be equal to 0, they should at least be as close as possible to 0. Traditional approaches are to weight the (squared) norm of these objective functions, or to prioritize them.

2.1 Weighting

Weighting would lead to solving the following, unconstrained nonlinear program:

$$\underset{r}{\text{minimize}} \quad \frac{1}{2} \|f(r)\|^2 + \frac{\omega}{2} \|g(r)\|^2, \quad (1)$$

with some given positive weight ω . In this case, the first order necessary condition for optimality is that

$$f(r)^T \frac{\partial f(r)}{\partial r} + \omega g(r)^T \frac{\partial g(r)}{\partial r} = 0. \quad (2)$$

If there is no conflict between these objectives, a solution with both $f(r) = 0$ and $g(r) = 0$ can be realized, and the necessary optimality condition above is trivially satisfied. However, if there is a conflict, the solutions are with $f(r) \neq 0$ or $g(r) \neq 0$

or both, and the necessary optimality condition above reveals that the rows of the Jacobian matrices are linearly dependent.

2.2 Prioritizing

Prioritizing would lead to imposing that $g(r) = 0$ (if possible), and solving the following, constrained nonlinear program:

$$\underset{r}{\text{minimize}} \quad \frac{1}{2} \|f(r)\|^2 \quad (3)$$

$$\text{such that } g(r) = 0. \quad (4)$$

In this case, the first order necessary condition for optimality is that

$$f(r)^T \frac{\partial f(r)}{\partial r} + \lambda^T \frac{\partial g(r)}{\partial r} = 0 \quad (5)$$

with some Lagrange multiplier λ to be determined.

Once again, if there is no conflict between these objectives, a solution with $f(r) = 0$ can be realized, and the necessary optimality condition above is trivially satisfied (with $\lambda = 0$). However, if there is a conflict, the solutions are with $f(r) \neq 0$ (and potentially $\lambda \neq 0$), and the necessary optimality condition above reveals that the rows of the Jacobian matrices are again linearly dependent.

2.3 Conflicts and Linear Dependency

It appears that situations of conflict are actually situations of linear dependency. This shouldn't come as a surprise, as this linear dependency is nothing more than the mathematical expression of the fact that objectives would like to draw from the same set of resources of the robot (expressed here in the variable r). We're going to see that this linear dependency can be the source of significant problems.

3 An Elementary Mobile Manipulator Robot

Consider an elementary mobile manipulator robot with two degrees of freedom: able to translate along the x axis, and equipped with an arm of unit length attached to a rotary joint with an angle θ , as depicted on Figure 4, so $r = (x, \theta)$. Suppose that it is assigned with the following two objectives:

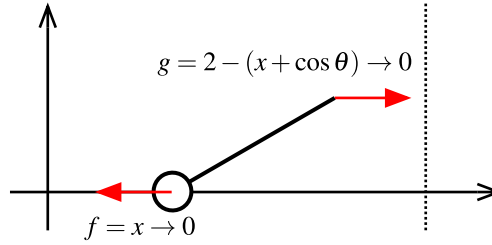
1. Reach a target with its end effector, at a coordinate equal to 2,

$$g = 2 - (x + \cos \theta) \rightarrow 0. \quad (6)$$

2. If possible, additionally, stay at the origin,

$$f = x \rightarrow 0. \quad (7)$$

Fig. 4 An elementary mobile manipulator robot, able to translate along the x axis, and equipped with one rotary joint of angle θ , is assigned with the following objectives:
1. Reach an object with its end effector, $g = 2 - (x + \cos \theta) \rightarrow 0$; 2. Stay at the origin, $f = x \rightarrow 0$.



3.1 Unreachability in the Objectives Space

In the space defined by the two functions (f, g) , the common objective is to be as close as possible to the origin $(0, 0)$ (see Figure 5). We can observe however that this point is not reachable since, following the definitions (6) and (7) of these functions, we obviously always have

$$f + g \geq 1. \quad (8)$$

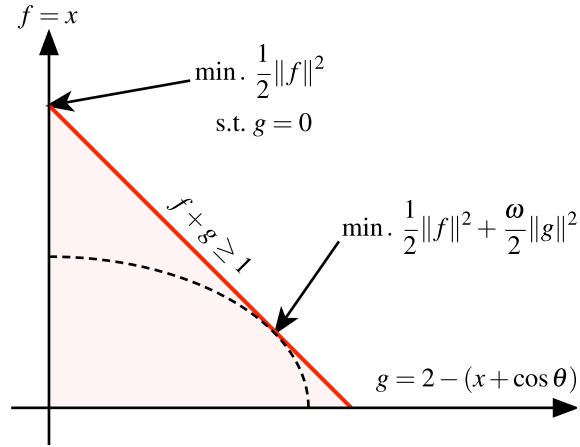
This unreachability means that these two functions can't be equal to 0 at the same time: there is a conflict between these objectives.

3.2 Solutions lie at Singularities

We have seen that situations of conflict are situations of linear dependency. In this example, depending on the choice of resolution scheme, weighted (1) or prioritized (3)-(4), solutions will lie at different points on the boundary of the reachable space, where $f + g = 1$ (see Figure 5). We can see from the definitions (6) and (7) that on this boundary, $\theta = 2k\pi$. Our mobile manipulator looks then as in Figure 6, and the Jacobian matrix of the two objective functions

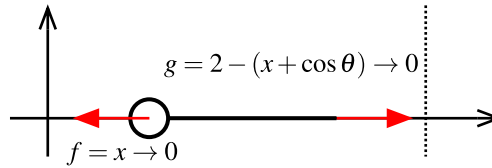
$$\begin{bmatrix} df \\ dg \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -1 & \sin(\theta) \end{bmatrix} \begin{bmatrix} dx \\ d\theta \end{bmatrix} \quad (9)$$

Fig. 5 In the space defined by the two functions (f, g) , the common objective is to be as close as possible to the origin $(0, 0)$. However, this point is not reachable since, following the definitions of these functions, we obviously always have $f + g \geq 1$. Depending on the choice of resolution scheme, solutions will lie at different points on the boundary, where $f + g = 1$.



becomes singular.

Fig. 6 When $f + g = 1$, $\theta = 2k\pi$, and the Jacobian matrix of the two objective functions becomes singular.



3.3 A Common Outcome of Redundancy Resolution Schemes

Most examples provided in the literature on redundancy resolution schemes typically consider conflicts with objective functions $\|f(r)\|^2$ of the form $\|r - r_p\|^2$, where r_p is some preferred value for the resources r [6, 7, 8]. In those cases, the Jacobian matrix $\frac{\partial f(r)}{\partial r}$ is the identity matrix, which is always trivially linearly dependent with any Jacobian matrix $\frac{\partial g(r)}{\partial r}$. It follows that in those cases, linear dependency does not materialize only at the optima: there are no particular losses of matrix rank at the solutions, they do not particularly lie at singularities.

But such objective functions involve by construction all the resources of the robot, so there are no redundant resources left afterwards for the redundancy resolution scheme to proceed with. Objective functions which do not involve all the resources of the robot are naturally much more frequent. However, few examples of conflicts in such situations have appeared in the literature, and mostly in the case

of humanoid robots [2, 9, 10, 11], probably because such robots can be particularly redundant. In these cases, linear dependency may materialize only at the optima, so there is a loss of matrix rank there: solutions lie at singularities, as above. This has already been acknowledged in [12] to be a common occurrence in redundancy resolution schemes. This singularity at the solutions is what will make standard approaches behave inefficiently.

4 Resolution Process

The usual resolution method for Nonlinear Least Squares Programs is to perform a sequence of Gauss-Newton steps, where nonlinear functions are replaced with their first-order linear approximations [13]. In the case of the nonlinear programs introduced earlier for the weighted and prioritized approaches, this leads to Quadratic Programs (QPs)

$$\underset{dr}{\text{minimize}} \frac{1}{2} \left\| f(r) + \frac{\partial f(r)}{\partial r} dr \right\|^2 + \frac{\omega}{2} \left\| g(r) + \frac{\partial g(r)}{\partial r} dr \right\|^2 \quad (10)$$

and

$$\underset{dr}{\text{minimize}} \frac{1}{2} \left\| f(r) + \frac{\partial f(r)}{\partial r} dr \right\|^2 \quad (11)$$

$$\text{such that } g(r) + \frac{\partial g(r)}{\partial r} dr = 0. \quad (12)$$

In robotics, this would correspond to a Closed-Loop Inverse Kinematics (CLIK) method, or to a task-space velocity control (but similar results are actually observed for task-space acceleration or torque control).

4.1 On the Boundary of the Reachable Space

When the state of our robot is on the boundary of the reachable space, when $f + g = 1$ and $\theta = 2k\pi$, we have seen that the rows of the Jacobian matrices $\frac{\partial f(r)}{\partial r}$ and $\frac{\partial g(r)}{\partial r}$ are linearly dependent, and the robot is in a singularity (as in Figure 6). In this case, the solutions to the QPs above are such that

$$dx = \frac{\omega}{1 + \omega} - x \quad (13)$$

for the weighted approach, and

$$dx = 1 - x \quad (14)$$

for the prioritized approach. Considering a least-norm solution, as usual in this case, we would also have $d\theta = 0$. The two numerical schemes appear therefore to converge to their respective solutions (on the boundary of the reachable space): $x = \frac{\omega}{1+\omega}$ and $\theta = 2k\pi$ for the weighted approach, $x = 1$ and $\theta = 2k\pi$ for the prioritized approach.

4.2 Away from the Boundary

When the state of our robot is away from the boundary of the reachable space, when $f + g > 1$, rows of the Jacobian matrices are linearly *independent*, so the linearized objectives

$$f(r) + \frac{\partial f(r)}{\partial r} dr = 0 \quad (15)$$

and

$$g(r) + \frac{\partial g(r)}{\partial r} dr = 0 \quad (16)$$

do not conflict and can be realized altogether. In this case, both numerical schemes lead the robot in the same direction dr , such that

$$df = \frac{\partial f(r)}{\partial r} dr = -f \quad (17)$$

and

$$dg = \frac{\partial g(r)}{\partial r} dr = -g. \quad (18)$$

This direction points towards the origin $(0,0)$ of the objectives space, towards the boundary of the reachable space (Figure 5).

4.3 Not Going in the Right Direction

We have seen that both the weighted and the prioritized numerical schemes converge to their respective solutions when on the boundary of the reachable space, and when away from this boundary, they first move towards it. It appears however that they don't go really in the right direction in that case. Indeed, following definition (7), equation (17) means

$$dx = -x, \quad (19)$$

so the mobile base of the robot (with coordinate x) is actually moving towards the origin $x = 0$, instead of any of the desired solutions $x = \frac{\omega}{1+\omega}$ or $x = 1$.

This is because the two linearized objectives do not conflict when away from the boundary, so the resolution schemes can temporarily aim at satisfying both, and that means having x go towards 0 in the first place. This behavior can be grasped

visually on Figure 4, where the two linearized objectives can be represented as two red arrows: the mobile base will move in the direction of the origin until the robot looks as on Figure 6, when the state of the robot arrives on the boundary of the reachable space, and the two linearized objectives finally conflict. This conflict is then resolved satisfactorily, and the robot is eventually led to the desired solution.

The robot must therefore reach first a situation of conflict between the linearized objectives, *i.e.*, a singularity, before going in the right direction. And in case the numerical scheme is not made robust to singularities, it might get stuck oscillating around this singularity, never reach it, and never start moving in the right direction. This exact behavior was clearly observed in [14]. The usual approach to singularity robustness is to introduce damping (Tikhonov regularization) [15], which has been observed to work properly in the case of conflicting objectives [10], although it has been acknowledged to be difficult to tune by hand, as the problem is actually very sensitive [2]. This is one of the issues addressed in the method proposed next, a key feature of which is automatic tuning of parameters.

5 Introducing Artificial Conflicts

Having the robot go in a wrong direction, and waiting until it reaches a singularity before it starts going in the right direction, introduces an unpredictable delay before convergence begins, what is inefficient and undesirable. This happens because the desired solutions lie at singularities, and to provide a meaningful approximation, the first-order linear models have to be singular in a similar way. Otherwise, these models basically point in a wrong direction. The idea then is to introduce carefully tuned artificial conflicts, to interfere with these models and have them point in a better direction.

5.1 A Trust-Region Method

A standard remedy to rank-deficiency issues in Nonlinear Least Squares Programs is to resort to the Levenberg-Marquardt method, which can be seen as combining Gauss-Newton steps with a trust-region method [13]. For the prioritized approach, the QP (11)-(12) can be modified in the following way:

$$\underset{dr}{\text{minimize}} \quad \frac{1}{2} \left\| f(r) + \frac{\partial f(r)}{\partial r} dr \right\|^2 \quad (20)$$

$$\text{such that } g(r) + \frac{\partial g(r)}{\partial r} dr = 0, \quad (21)$$

$$\|dr\|_\infty \leq \Delta, \quad (22)$$

introducing a bound Δ on the norm of the step dr , using here an L^∞ -norm to obtain a standard QP formulation.

The crucial consequence of introducing this bound is that it doesn't only affect the size of dr , it also affects its direction, by interfering with the minimization of the objective in (20). This is unlike a line search method, which keeps the direction constant and adapts only the size of dr (that would correspond in robotics to simply varying control gains, such as in task scaling [16]).

If the bound Δ is chosen large enough, it will not interfere, and the solution of the QP will be unaffected. But if chosen small enough, it will conflict with the minimization of the objective in (20), producing a direction dr that fully reflects the objective in the constraint (21), much less the objective to minimize in (20), as if these two objectives were themselves conflicting.

Introducing such an artificial conflict is appropriate only if the two objectives do conflict in the end. The problem is, there is usually no way to know in advance if this is the case or not. The bound Δ must therefore be carefully adapted online, based on some heuristics. It has been proposed in [14] to adapt this bound, depending on the occurrence of oscillations, due to singularity. In that case however, it is only when approaching singularity that the artificial conflict would be introduced, so the problem of going in a wrong direction in the first place would remain.

We propose here to follow a standard trust-region method, where this bound is adapted, depending on how much the nonlinear objective functions differ from their first-order linear approximations throughout the Gauss-Newton iterations [13]. As a result, artificial conflict can be introduced early in the process, taking care automatically that it does not interfere with convergence.

5.2 Adaptive Damping

Following a theorem due to Moré and Sorenson [13], trust-region methods can also be considered as a form of adaptive damping. For the weighted approach, the QP (10) can be modified in the following way:

$$\underset{dr}{\text{minimize}} \quad \frac{1}{2} \left\| f(r) + \frac{\partial f(r)}{\partial r} dr \right\|^2 + \frac{\omega}{2} \left\| g(r) + \frac{\partial g(r)}{\partial r} dr \right\|^2 + \frac{\Lambda}{2} \|dr\|^2 \quad (23)$$

with a regularization coefficient Λ that has to be carefully chosen and adapted.

If $\Lambda = 0$, the QP and its solution are unaffected. But if $\Lambda > 0$, the regularization term $\|dr\|^2$ naturally conflicts with both objectives. And if Λ is large enough with respect to $\min.\{1, \omega\}$, this regularization term will interfere significantly with the objective with lower weight, producing a direction dr that reflects relatively much more the objective with higher weight, as if these two objectives were themselves conflicting.

As before, introducing such an artificial conflict is appropriate only if the two objectives do conflict in the end, and there is usually no way to know in advance if

this is the case or not. The regularization weight Λ must therefore be adapted online as well, based on some heuristics. As before, we propose to follow a standard trust-region method, where this weight is adapted, depending on how much the nonlinear objective functions differ from their first-order linear approximations throughout the Gauss-Newton iterations [13].

Adjusting the damping coefficient is usually proposed in robotics with the goal to interfere as little as possible with the linearized objectives, only when they become close to singular [1]. The situation here is opposite, since the goal is to interfere significantly with the linearized objectives, and especially in situations where they are far from singular.

5.3 Implementation Aspects

Concerning the weighted approach, the QP (10) could also be modified to include a bound (22) instead of a regularization, but that would transform the original, unconstrained Least Squares problem into a slightly more complex, constrained one. This is the reason why the regularized formulation (23) is usually favored.

Concerning the prioritized approach, the QP (11)-(12) could also be modified to include a regularization instead of a bound, as in [2, 10]. But solutions to the non-regularized prioritized problem can be obtained very efficiently, with the help of specific matrix factorizations, and lexicographic active set methods when handling inequality constraints [9, 17], which do not apply to the regularized case. The difference in computation time can be more than 30-fold – a good reason to favor the bounded formulation (20)-(22). An advantageous aspect of regularization however is that it precludes ill-conditioning when approaching singularities, and ill-conditioning can prevent the proper termination of the active set methods used to handle inequality constraints, by inducing a cycling in their iterations. Which option should be favored in the end is still an open question.

Finally, traditional trust-region methods are designed for unconstrained, single-objective optimization problems [13]. They apply naturally to the regularized Least Squares problem (23) that appears in the weighted approach. But for the constrained, multi-objective problems that appear in the prioritized approach, we have to resort to more recent and experimental, multidimensional filter methods [18].

6 Numerical Results

We are going to observe now more precisely how the numerical schemes discussed above behave with the problem (6)-(7). A set of initial guesses for the solutions is chosen randomly, each assigned a unique color, and subsequent iterations are plotted in the objectives space, defined by the two functions (f, g) , using the same layout

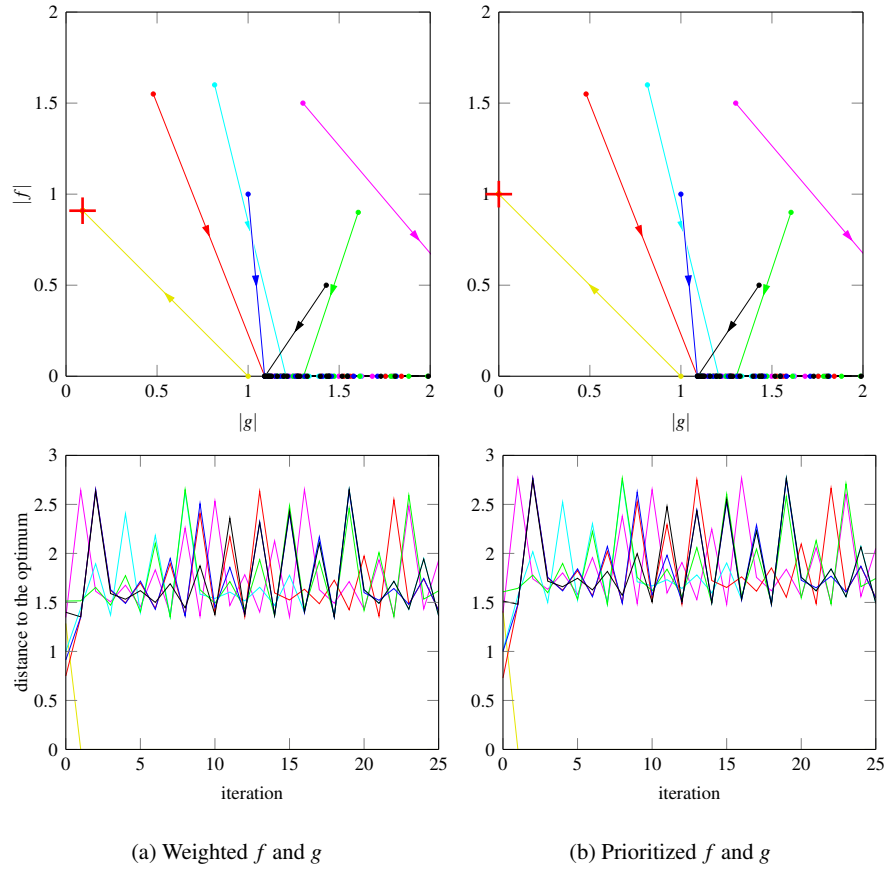


Fig. 7: Sequences of Gauss-Newton steps, as introduced in the QP (10) for the weighted approach (on the left, with a weight $\omega = 10$), and in the QP (11)-(12) for the prioritized approach (on the right), do not converge to the corresponding optima $(\frac{1}{11}, \frac{10}{11})$ and $(0, 1)$, represented by red crosses on the (f, g) plots above, except when initialized exactly on the boundary $f + g = 1$. The curves below show the corresponding evolutions of the distance to the optima with each iteration, which does not decrease at all in general.

as in Figure 5. The distance to the solutions is also provided as a function of the number of iterations in order to visualize convergence speed.

We can see in Figure 7 that sequences of Gauss-Newton steps, as introduced in the QP (10) for the weighted approach (on the left, with a weight $\omega = 10$), and in the QP (11)-(12) for the prioritized approach (on the right), do not converge to the corresponding optima $(\frac{1}{11}, \frac{10}{11})$ and $(0, 1)$, represented by red crosses, except when initialized exactly on the boundary $f + g = 1$. Note how iterations are identical

between the weighted approach (on the left) and the prioritized approach (on the right) when $f + g > 1$, as discussed earlier in Section 4.2.

We can see in Figure 8 that simply dividing the length of the steps by 10, as would occur with a gain of 0.1 in a Closed Loop Inverse Kinematics scheme, naturally slows down each and every iteration, but does not lead to any improvement in convergence.

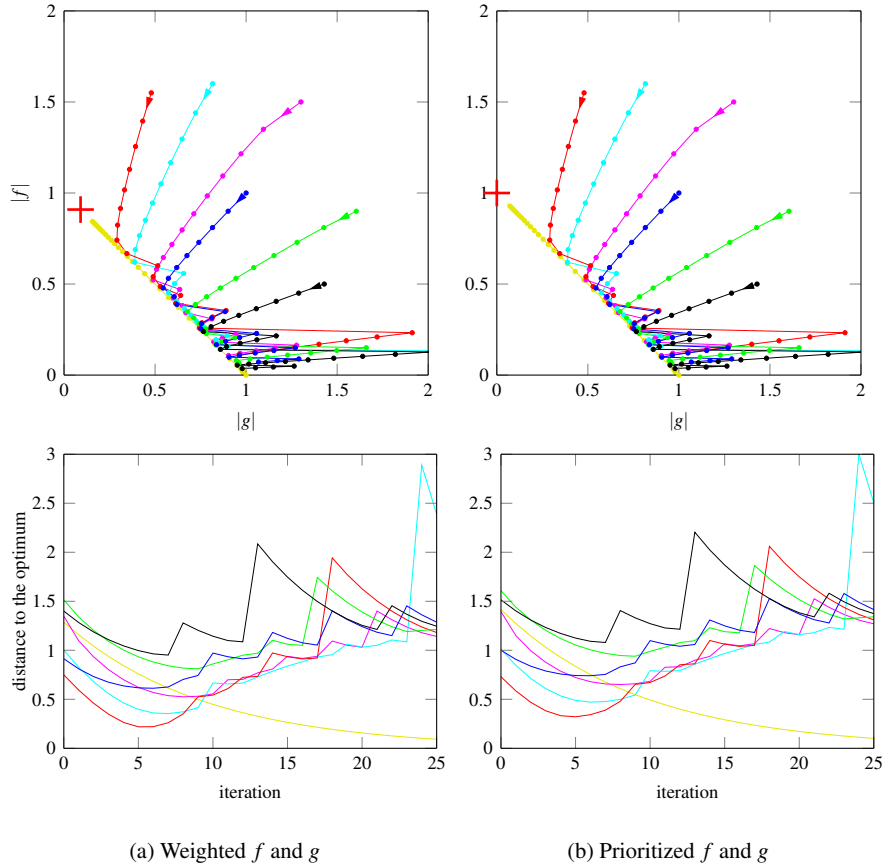


Fig. 8: Simply dividing the length of the steps by 10, as would occur with a gain of 0.1 in a Closed Loop Inverse Kinematics scheme, naturally slows down each and every iteration, but does not lead to any improvement in convergence.

We can see in Figure 9 that for the weighted approach, introducing a regularization as in the QP (23), with a coefficient as high as $\Lambda = 0.4$, still does not lead to any better convergence. For the prioritized approach, introducing a bound as in the QP (20)-(21), with a constant $\Delta = 0.5$, leads to convergence in a few occasions,

but not systematically. We can see in Figure 10 that it is only with a regularization coefficient as high as $\Lambda = 0.55$ and a bound as small as $\Delta = 0.1$ that convergence begins to appear more reliably.

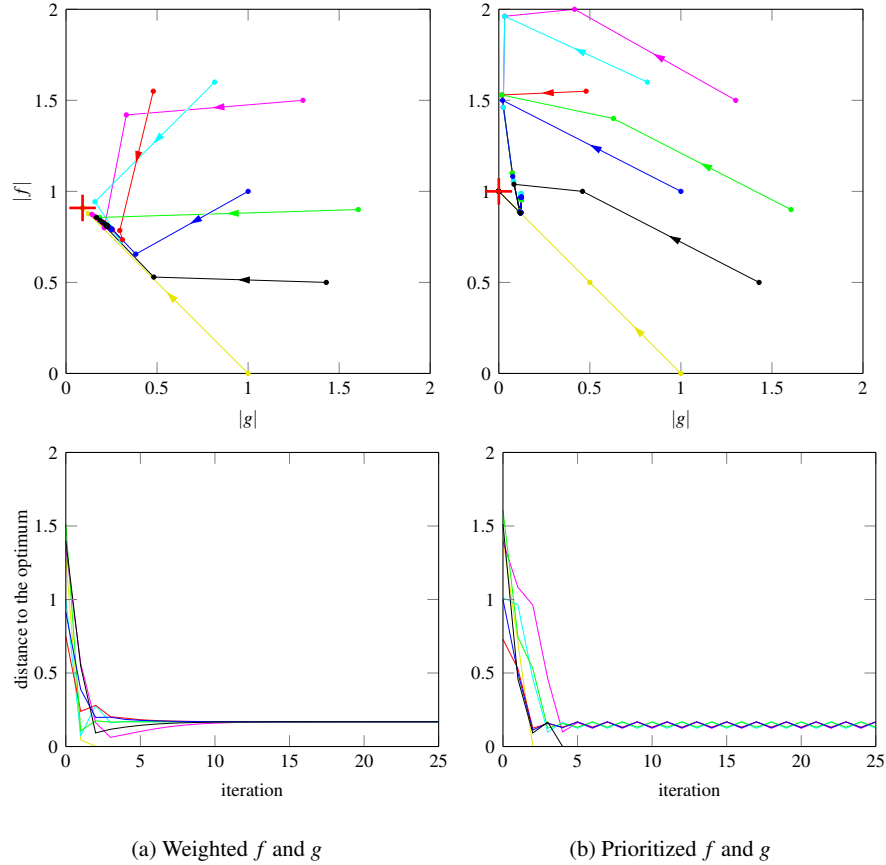


Fig. 9: For the weighted approach (on the left), introducing a regularization as in the QP (23), with a coefficient as high as $\Lambda = 0.4$, still does not lead to any better convergence. For the prioritized approach (on the right), introducing a bound as in the QP (20)-(21), with a constant $\Delta = 0.5$, leads to convergence in a few occasions, but not systematically.

It appears that the regularization coefficient Λ and the bound Δ must be tuned carefully in order to reach convergence. This can be done automatically and efficiently with a simple trust-region method, as discussed in [13]. The corresponding behavior can be observed in Figure 11. Of course, convergence speed may not always be as good as with a finely hand tuned coefficient or bound, but this does not

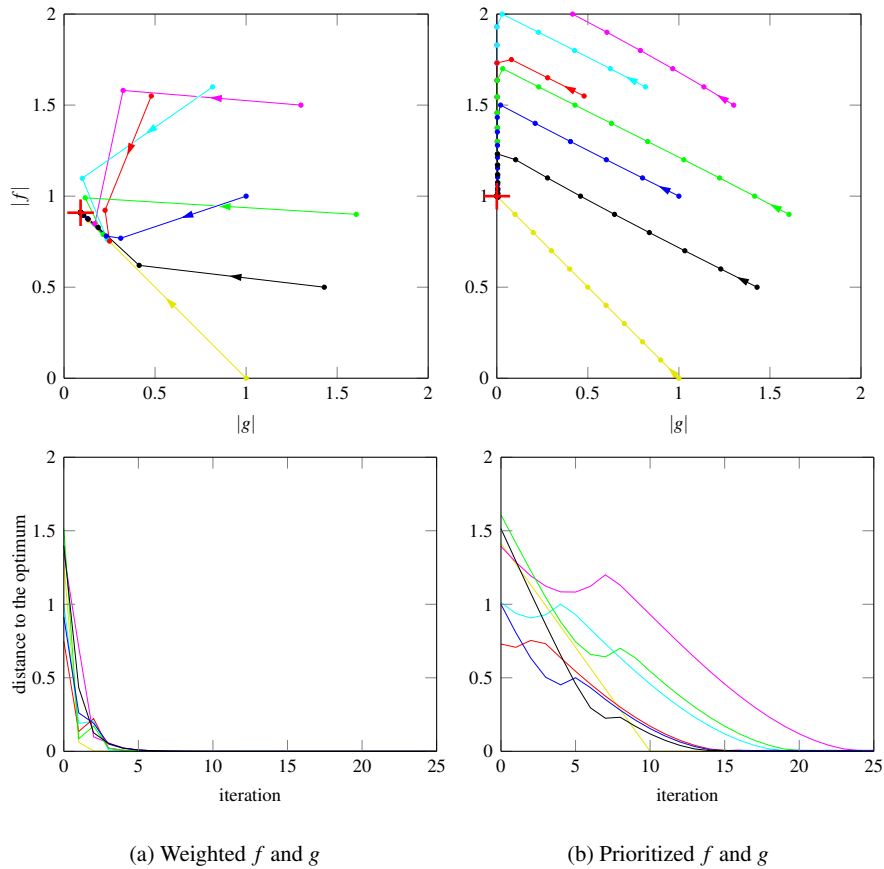


Fig. 10: It is only with a regularization coefficient as high as $\Lambda = 0.55$ (on the left) and a bound as small as $\Delta = 0.1$ (on the right) that convergence begins to appear more reliably.

account for the time required to hand tune these problem-specific parameters in the first place.

7 Conclusion

We have seen that redundancy resolution schemes can result in a limited form of decision making, when attributing the resources of a robot to its different objectives. This is because the amount of resources required to realize a given objective can vary, due particularly to the introduction of inequality constraints, that can be active

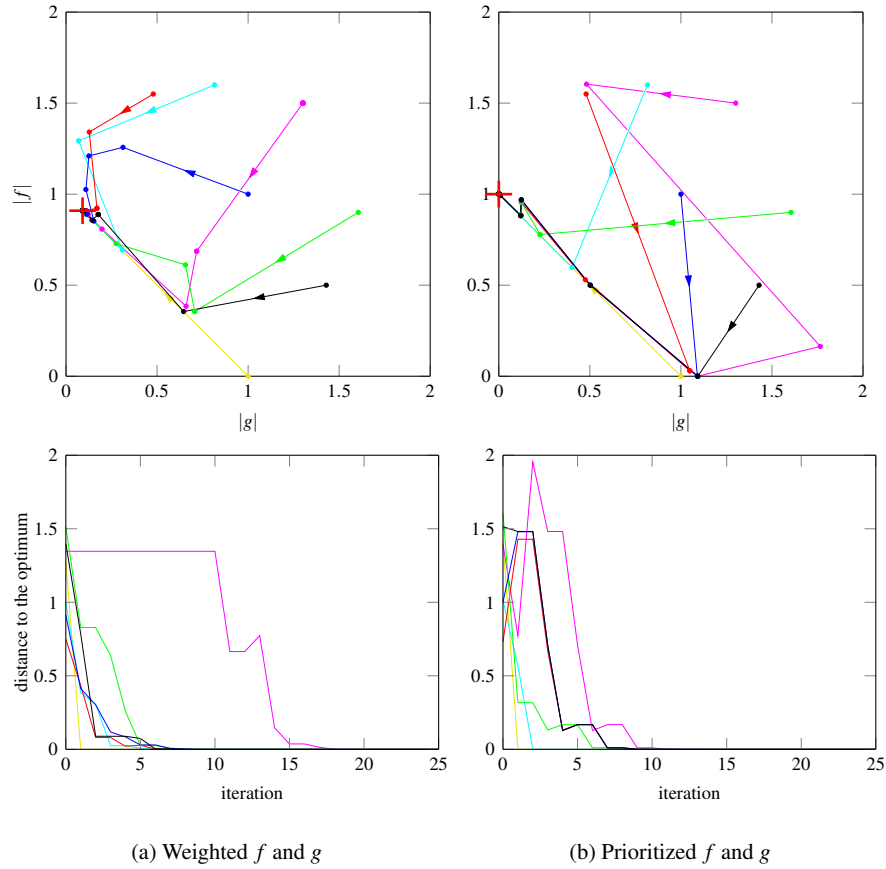


Fig. 11: It appears that the regularization coefficient Λ and the bound Δ must be chosen and adapted carefully in order to obtain convergence. This can be done automatically and efficiently with a simple trust-region method, as discussed in [13].

or not, depending on the situation. Such decision making emerges in case of conflicts between objectives, and we have seen that these conflicts are directly related to situations of linear dependency.

The problem is, that such linear dependency often results in singular solutions. In that case, standard resolution processes appear to go in the wrong direction, causing an unpredictable delay in convergence, because they rely on linearized objectives, which can be inappropriate. It is striking to observe that the resolution processes behave nearly identically for the weighted and for the prioritized approach: both face exactly the same numerical difficulty here.

We propose to use trust-region methods as a remedy. They can be seen as a way to introduce artificial conflicts, automatically and carefully tuned to drive the resolu-

tion processes towards the desired solution. This has been demonstrated numerically with a simple system. More complex situations, with a greater number of objectives, need to be addressed now for a more thorough validation of the proposed method.

In conclusion, no claims can be made yet as to having fully solved the problem uncovered in this chapter. Further study of how humans handle such situations could prove enlightening in this regard, and may provide inspiration for more refined solutions.

References

1. S. Chiaverini, G. Oriolo, and I. D. Walker, "Kinematically Redundant Manipulators," in *Springer Handbook of Robotics* (B. Siciliano and O. Khatib, eds.), ch. 11, pp. 245–265, Springer Berlin Heidelberg, May 2008.
2. O. Kanoun, F. Lamiroux, and P.-B. Wieber, "Kinematic control of redundant manipulators: generalizing the task-priority framework to inequality task," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 785–792, 2011.
3. A. Sherikov, D. Dimitrov, and P.-B. Wieber, "Balancing a humanoid robot with a prioritized contact force distribution," in *IEEE-RAS International Conference on Humanoid Robots*, pp. 223–228, 2015.
4. N. Bohórquez, A. Sherikov, D. Dimitrov, and P.-B. Wieber, "Safe navigation strategies for a biped robot walking in a crowd," in *IEEE-RAS International Conference on Humanoid Robots*, 2016.
5. S. Al Homsí, A. Sherikov, D. Dimitrov, and P.-B. Wieber, "A hierarchical approach to minimum-time control of industrial robots," in *IEEE/RSJ International Conference on Robotics and Automation*, pp. –, 2016.
6. B. Faverjon and P. Tournassoud, "A local based approach for path planning of manipulators with a high number of degrees of freedom," in *IEEE/RSJ International Conference on Robotics and Automation*, pp. 1152–1159, 1987.
7. Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *The International Journal of Robotics Research*, vol. 6, no. 2, pp. 3–15, 1987.
8. B. Siciliano and J.-J. E. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *International Conference on Advanced Robotics*, pp. 1211–1216, 1991.
9. A. Escande, N. Mansard, and P.-B. Wieber, "Hierarchical quadratic programming: Fast online humanoid-robot motion generation," *The International Journal of Robotics Research*, vol. 33, no. 7, pp. 1006–1028, 2014.
10. A. Del Prete, F. Nori, G. Metta, and L. Natale, "Prioritized motion force control of constrained fully-actuated robots: "task space inverse dynamics"," *Robotics and Autonomous Systems*, vol. 63, no. 1, pp. 150–157, 2015.
11. A. A. Maciejewski and C. A. Klein, "Task-priority based redundancy control of robot manipulators," *The International Journal of Robotics Research*, vol. 4, no. 3, pp. 109–116, 1985.
12. T. Sugihara, "Robust solution of prioritized inverse kinematics based on Hestenes-Powell multiplier method," in *IEEE/RSJ International Conference on Intelligent Robots and System*, pp. 510–515, 2014.
13. J. Nocedal and S. Wright, *Numerical optimization (2nd Edition)*. Springer, 2006.
14. F. M. Bianchi and G. Gualandi, "Handling and prioritizing inequality constraints in redundancy resolution optimization-based approach with task priority," Master's thesis, Sapienza University of Rome, 2011.

15. Y. Nakamura and H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 108, no. 3, pp. 163–171, 1986.
16. F. Flacco, A. De Luca, and O. Khatib, "Control of redundant robots under hard joint constraints: Saturation in the null space," *IEEE Transactions on Robotics*, vol. 31, no. 3, pp. 637–654, 2015.
17. D. Dimitrov, A. Sherikov, and P.-B. Wieber, "Efficient resolution of potentially conflicting linear constraints in robotics," (*Submitted*).
18. N. Gould, S. Leyffer, and P. Toint, "A multidimensional filter algorithm for nonlinear equations and nonlinear least squares," *SIAM Journal on Optimization*, vol. 15, no. 1, pp. 17–38, 2004.